iCatch3 API Specification

Introduction

Itesco selection machinery referred further as iCatch3 provides API for means of integration with external systems. API provides all the features available to iCatch3 users through the web UI interface and even more. API basically follows REST concept and uses HTTP/HTTPS as an application level protocol. Resource naming follows common conventions in order to make it easier for API users to begin with integration.

For rapid start using the service you can refer to Quick Start section.

        - Checking correctness of supplied Private Persons Exclusions
        - Getting person query instance settings by query id
        - Getting person query instance settings by its id
        - Getting Private Prospects Address Filter used for query
        - Checking correctness of supplied Private Prospects Address Criteria List
        - Getting extended Private Prospects Query Info
- Company Prospects
    - Creating new Company Prospects settings
    - Updating Company Prospects settings
    - Copying Company Prospects settings
    - Deleting Company Prospects settings
    - Getting information about your Company Prospects settings
    - Getting Stored Company Exclusions
    - Updating Stored Company Exclusions
    - Deleting Stored Company Exclusions
    - Getting Stored Company Address Filter
    - Updating Stored Company Address Filter
    - Deleting Stored Company Address Filter
    - Estimating number of answers for Company Prospects Query
    - Posting Company Prospects Query for execution
    - Getting information about your Company Prospects queries
    - Getting information about your particular query progress
    - Getting information about your Company Prospects Query results
    - Getting all your Company Prospects query results
    - Getting your Company Prospects query results in chunks
    - Updating Company Prospects query results related attributes
    - Getting Company Query Instance settings by query id
    - Getting Company Query Instance settings by its id
    - Getting Company Prospects Exclusions used for query
    - Checking correctness of supplied Company Exclusions
    - Getting Company Prospects Address Filter used for query
    - Checking correctness of supplied Company Address Criteria List
    - Getting extended Company Prospects Query Info
- Group Search
    - Getting information about your Group Search settings
    - Creating new Group Search settings
    - Copying Group Search settings
    - Updating Group Search settings
    - Deleting Group Search settings
    - Estimating number of answers for group query
    - Getting information about your group queries
    - Getting information about your particular group query progress
    - Posting a group query for execution
    - Getting group query instance settings by query id
    - Getting group query instance settings by its id
    - Getting information about your group query results
    - Getting all your group query results
    - Getting your group query results in chunks
    - Updating group query results related attributes
    - Checking correctness of supplied Questions for Group Search
    - Getting extended Group Query Info
    - Getting your Group Query Input
    - Getting available Features List for Group Search
    - Getting available Matching Templates
        - Matching Templates List
        - Matching Rules for Templates
- NIX Queries
    - Getting information about your NIX queries

Quick Start

Within this section you can find manuals on how to quickly start using iCatch3 API. Most typical flows are described and illustrated with Sequence Charts for better understanding.

Logging in, getting your templates list, logging out

This is the starting point of all the further work with iCatch3 API.

You start with posting your credentials to special login URL. In response you are getting an auth token which you further enclose with all your requests. Once you are done you are logging out to invalidate the token been used.

Related Reading:

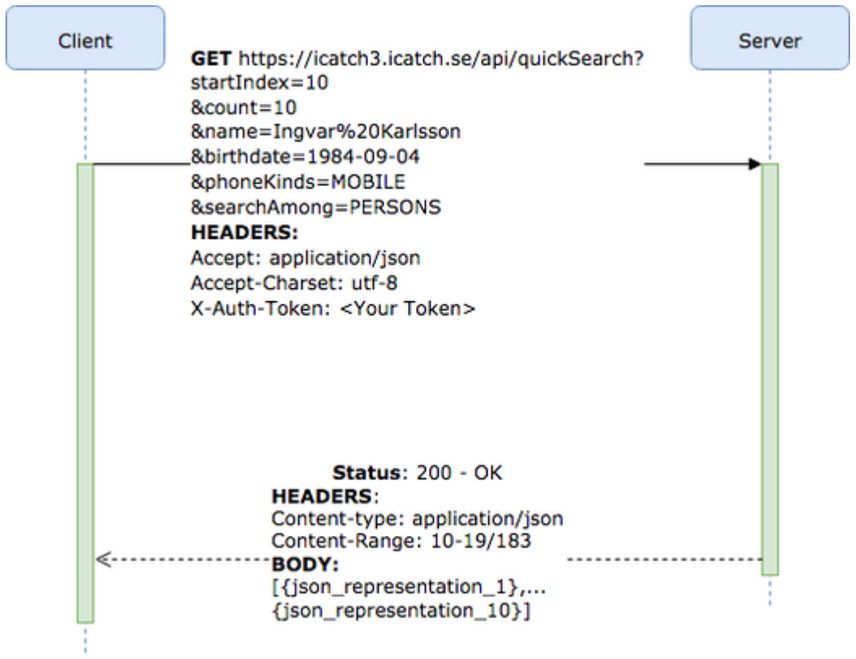
- Protocol
- Authentication
- Authorization

Doing Quick Search

Doing quick search is quite a straightforward operation. The only thing it requires to be done in advance is to be authorized thus have valid auth token available. So, you issue a get request and read response of matching parties.

Related reading:

- Quick Search



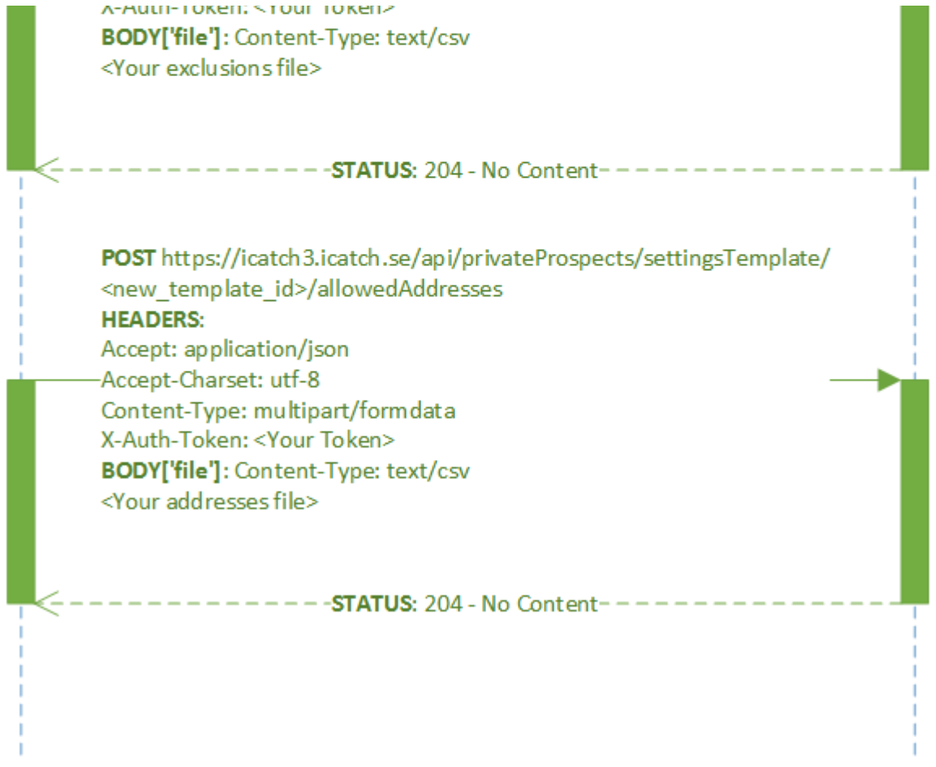
Creating a Private Prospects Setting template

Here we create at settings template for you private prospects queries which will be further use as a base for issuing queries. In iCatch3 all the queries are based on some settings which are built up on top of a settings template. While posting a query you can override any of settings, so that exactly for this query run your custom setting value is used.

You start with posting a definition if a template placing to your request body its properties as described in Private Prospects Settings. In response you are getting full template definition in body as JSON including ID which is further used to refer a template in your other requests. Exclusions/Address filter you could find useful are either uploaded separately (as depicted on chart below) or could be supplied in your initial POST request if you form it in Multi-part way.

Related reading

- Creating new Private Persons Prospects settings
- Getting information about your Private Persons Prospects settings
- Updating Stored Private Prospects Exclusions
- Updating Stored Private Prospects Address Filter
- Getting Stored Private Prospects Exclusions
- Getting Stored Private Prospects Address Filter



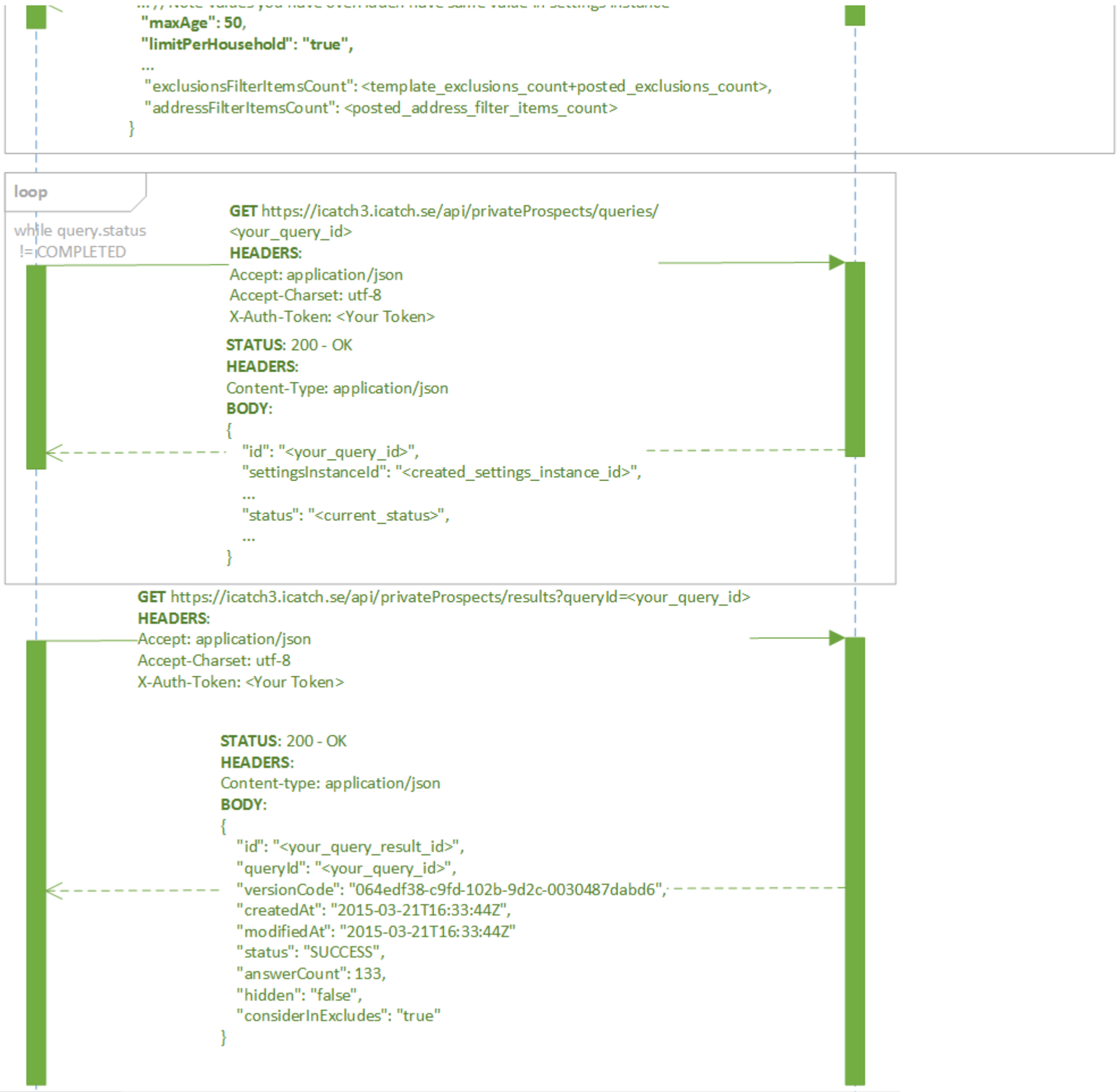
Posting Private Prospects query, retrieving result

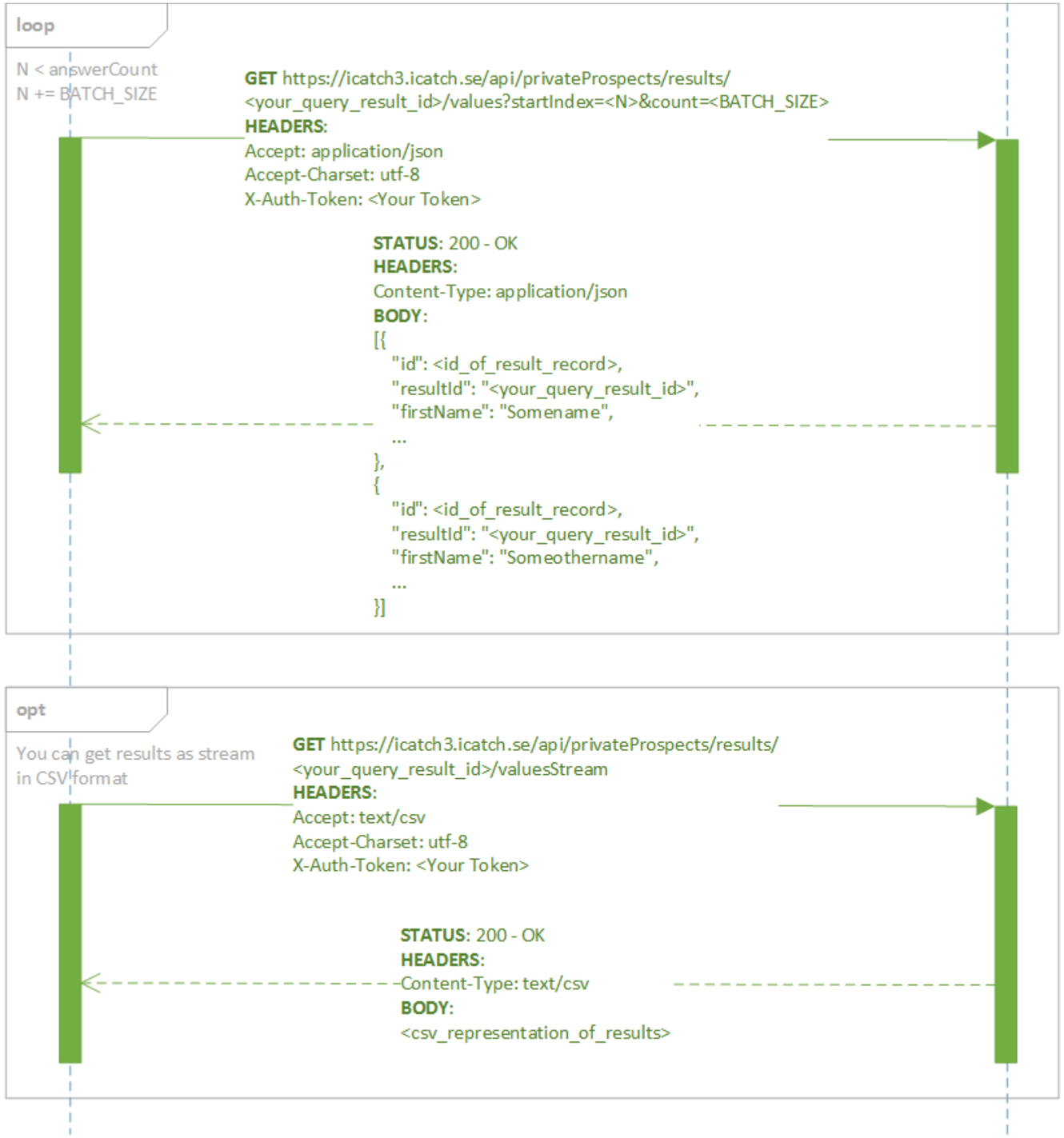
Ones you have already defined a template, you are able to post a query.

This is done by posting Query Definition object to a corresponding URL. In response you'll get a Query object which reflect the state of your query execution at server side. Then you should check your query state awaiting till it's status becomes COMPLETED (you can poll corresponding URL with some period). Once query is completed there you can retrieve Query Result object. This one describes a query result. ID of result can then be used to retrieve actual result value. You have several options here. You can retrieve result in chunks in JSON/CSV format or you can get all the results as a byte stream. For better performance we recommend getting result in a compressed form (see Getting all your private prospects query results).

Related reading

- Estimating number of answers for query
- Posting a query for execution
- Getting information about your particular person prospects query progress
- Getting information about your query results
- Getting all your private prospects query results
- Getting your query results in chunks





Creating a Group Query Setting template

Here we create at settings template for you group queries which will be further use as a base for issuing register care queries. In iCatch3 all the queries are based on some settings which are built up on top of a settings template. While posting a query you can override any of settings, so that exactly for this query run your custom setting value is used. Template does not include your actual input, just settings which are used to process it.

You start with posting a definition if a template placing to your request body its properties as described in Group Search Settings. In response you are getting full template definition in body as JSON including ID which is further used to refer a template in your other requests.

Related reading

- Creating new Group Search settings
- Updating Group Search settings
- Matching Templates List
- Matching Rules for Templates

Posting Group query, retrieving result

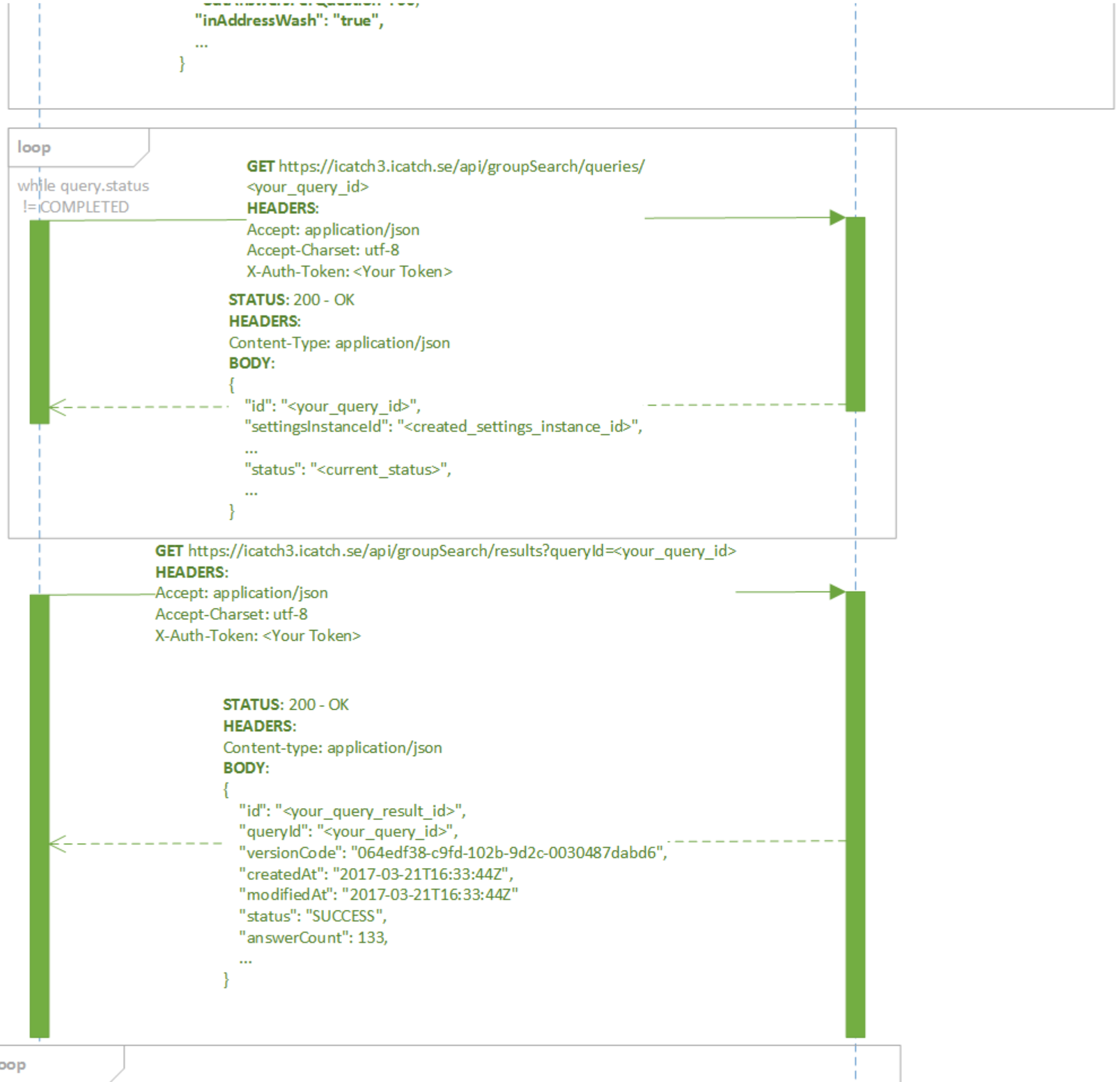
Ones you have already defined a template, you are able to post a query. For this you'll need to prepare input in Group Search Questions format.

You are posting a Group Query Definition object to a corresponding URL. In response you'll get a Query object which reflect the state of your query execution at server side. Then you should check your query state awaiting till it's status becomes COMPLETED (you can poll corresponding URL with some period). Once query is completed there you can retrieve Query Result object. This one describes a query result. ID of result can then be used to retrieve actual result value. You have several options here. You can retrieve result in chunks in JSON/CSV format or you can get all the results as a byte stream. For better performance we recommend getting result in a compressed form (see Getting all your group query results).

Related reading:

- Posting a group query for execution
- Getting information about your group query results
- Getting all your group query results

- Getting your group query results in chunks
- Getting extended Group Query Info
- Getting your Group Query Input



N < answerCount
N += BATCH_SIZE

GET https://icatch3.icatch.se/api/groupSearch/results/<your_query_result_id>/values?startIndex=<N>&count=<BATCH_SIZE>
HEADERS:
Accept: application/json
Accept-Charset: utf-8
X-Auth-Token: <Your Token>

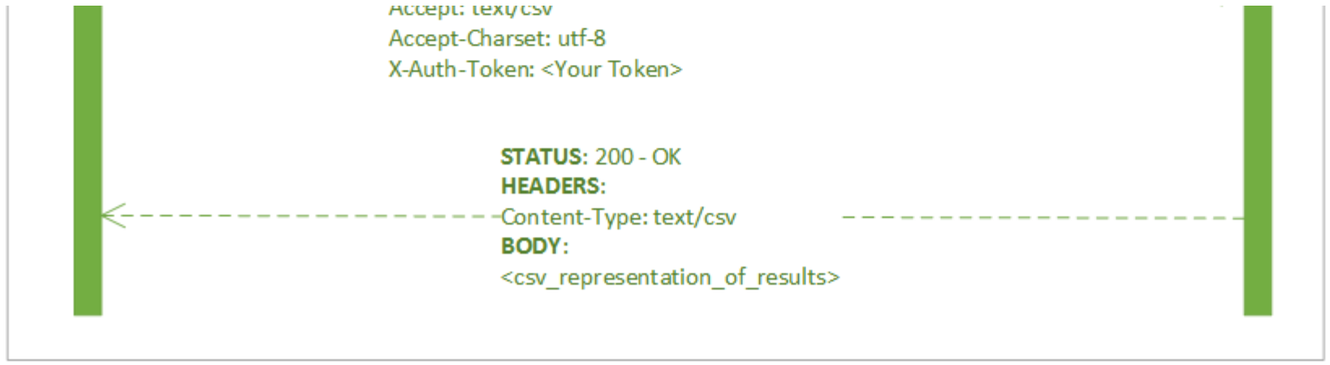
STATUS: 200 - OK
HEADERS:
Content-Type: application/json
BODY:

```
[{
 "questionInfo" : {
  "question" : {
   "id" : 21,
   "firstName" : "SomeName",
   "lastName" : null,
   "orgName" : null,
   ...
   "matchingTemplatesIds" : null
  }
 },
 "groupAnswers" : [
    {
     "matchingTemplateId" : "1234",
     "resultItem" : {
       "firstName" : "SomeName",
       "lastName" : "SomeLastName1",
       ...
       },
     "matchAddressType": null,
     "updated" : []
     }, {
     "matchingTemplateId" : "4321",
     "resultItem" : {
       "firstName" : "SomeName",
       "lastName" : "SomeLastName2",
      ....
       },
     "matchAddressType": null,
     "updated" : []
  ]
}
]
```

opt

You can get results as stream in CSVformat

GET https://icatch3.icatch.se/api/groupSearch/results/<your_query_result_id>/valuesStream
HEADERS:



Protocol

As the main application protocol API is using HTTP/HTTPS. For structured data representation almost all the methods use JSON. Some methods (usually bulk data extraction ones) also support CSV output. Methods which can potentially return relatively big set of data can use multipart content. There are also methods which could return several content types. In order to get response with non default content type (which is application/json in most cases) you should supply a header '**Accept: <required_content_type>**' to define expected content type.

Requests which require authorized access in order to be processed add some requirements for incoming requests format. You can find more details in Authentication section of current specification.

In addition to standard response codes which are natively supported by HTTP protocol response body could contain additional API specific error code which clarifies exact reason of rejected request or gives some other additional information about returned response. This code is usually present within the structured responses under **code** attribute.

Availability

For purposes of accessibility checking and also supplying some general meta data on API version you are communicating to there is provided special web service method ping. This is open endpoint which you can access without authorizing. In response to your PING request you should be getting information on current API version. This can help you detecting API update and quickly react to possible changes. Please, refer our Change Log section for getting information on what API changes were applied ins scope of each release.

Authentication

For means of providing basic security and avoid unauthorized clients gaining info from service, API service method requires **Authentication Token** to be supplied within request.

Authentication Token is a key which is used by the API for identifying and authorizing the access to specific features.

It is expected that each such request contains **X-Auth-Token** header. The authentication token should be set as value of the header without any changes. You should get Authentication token by calling special web service method login.

Please note, token can be invalidated at some point, so be advised to do re-login in case you start getting **401-Unauthorized** responses with **code = 401001** while accessing API methods. In general token could be invalidated in one of following cases:

1. Token validity period has exceeded. Each token is supplied for some definite validity period. The value of the period is account specific and could vary. In most cases validity period is somewhat close to several hours so you should not be forced to do re-login often due to this.
2. User Permissions has been changed. In case if your user account has been supplied with some new or has lost some existing privileges your current token will be invalidated. You should gain new token in order to start working using new permissions set.
3. Your user account has been blocked. For some reasons account could be blocked. In that case you should contact Itesco staff in order to get some explanation of what to do next.
4. Token was invalidated by explicit /logout method call.

When you are basically done with your work session it is recommended to do /logout in order to invalidate your current token. Doing logout is basically good practice that allows you to have control on your session life cycle, minimise security risks and also allows us more efficiently manage our client requests.

Note
The method described above is applicable for following services:

- Quick search
- Private Prospects
- Company Prospects
- Group Search
- User files storage
- Geographical Dictionary
- SNI Codes Dictionary

For accessing such services as

- SMS

you should be using JWT token provided in response to your login endpoint request. JWT token should be supplied as the value of "Authorization" header like this: **Authorization: Bearer *<JWT_TOKEN>***

Localization

Some endpoints support localized answers. Ex. when you are getting list of Industry codes (SNI codes) you can select which SNI dictionary to use for search English or Swedish by defining language with **Accept-Language** header. The value will affect not only the language in which you will get response data content values (like Industry description) but also tells the system which language to use when processing your input (like looking for the industry you have defined with filtering pattern in a specific language dictionary.

At the moment we support following languages:

- English - Accept-Language: EN
- Swedish - Accept-Language: SE

Values of header are case insensitive. If Accept-Language is not defined we are always falling back to using **English**.

Handling conflicts

When dealing with modification of state of an object (usually issuing PUT request with new state to some resource URL) there could happen a situation that resource has been already modified and has different state. Potentially, that could lead to data loss in case we allow request to proceed. API cannot make any assumptions of whether it is valid to update the state of entity with other state, so in that situation there will be issued **409 Status** code in response and update is not going to happen.

For detecting conflict situation each modifiable resource has **versionCode** attribute. The value of the attribute represents ID of version which is assigned to resource on every change of state. The version code gets returned to the client on each change which allows the client to track the state of the object. In addition actual version code gets returned in response with 409 status in order to make it possible for client still doing an update in case he does not matter about resolving the conflict which was detected. The value of versionCode is naturally not required while you are posting new resource. In that case initial version code is going to be returned to you among all other attributes of resource you have posted.

In general it is proposed to use following logic when you are going to update resource state:

1. Get current state of resource.
2. Modify state preserving versionCode value you've got.
3. PUT new state to corresponding resource URL.
4. If you've got response with 409 response code:
    a. If you are interested in resolving conflict
        i. Request actual state of resource
        ii. Resolve difference
        iii. Make a decision of how to update
        iv. PUT new state with most recent versionCode value you've got
    b. If you are not interested in resolving conflict and you want just to update state to one you need
        i. Get the value of actual versionCode from 409 status response body
        ii. Update value of versionCode within your PUT request body
        iii. PUT updated request again.

Paged Reading

Endpoints which return more than one entry support paged extraction of content. This allows us efficiently control the workload to server and reduces amount of resources consumed by clients and server and makes it possible to process really huge amounts of items. All the GET methods which return sets of items (not binary streams) support paging. There are used 2 request parameters **startIndex** and **count** in order to control amount of returned results.

- startIndex - value in range [0..totalItemsCount). Defines start index of answer being returned. 0 if not defined.
- count - value in range [1..totalItemsCount- startIndex]. Defines how many answers should be returned considering defined startIndex. 2 000 if not defined.

You can get Response with status 400 in one of following cases:

1. count <= 0
2. startIndex < 0
3. Start index >= total amount of available items (start index is set after the last item)
4. count > than max allowed page size (currently == 10 000)

In order to organize paged reading you'll also need Content-Range header which is returned with each response by endpoint supporting paged reads. The header in response describes returned content in following format:

```
<startIndex>-<endIndex>/<totalCount>
```

You can refer to endIndex and totalCount values in order to control your paged read operations.

Common response codes

Despite each particular API method has its specifics about response statuses usage, there are some response statuses which are returned by each API call in case of common failures. They all are described in this section.

HTTP Status	Content type	Content Format	Reason
400 - Bad Request	application/json	`{ "code": "400000", "message": "<error message>" }`	Request data cannot be parsed by the server, e.g. due to malformed JSON

400 - Bad Request	application/json	`{`
		`"code": "400001",`
		`"paramName":"<param_name>",`
		`"message": "Parameter is absent in the request."`
		`}`
			Mandatory parameter required to perform operation was not provided. Name of parameter is in body of response.
			Here and down below under "parameter" we mean
			• **Attributes of entity being posted (put)**
			• **GET request query parameter**

400 - Bad Request	aplication/json	`{`
		`"code": "400002",`
		`"paramName":"<param_name>",`
		`"rangeMin": "min_value_allowed",`
		`"rangeMax": "max_value_allowed",`
		`"message": "Parameter value \"<param_value>\" is out of range."`
		`}`
			Issued when value of given parameter is out of expected range.
			• For string parameters this is usually issued when given string length is too long.
			• For numeric values this one is issued when given number exceeds specified range

400 - Bad Request	aplication/json	`{`
		`"code": "400003",`
		`"paramName":"<param_name>",`
		`"allowedValues": "allowedValuesArray",`
		`"message": "Parameter value \"<param_value>\" is not allowed."`
		`}`
			Issued when value of given parameter is constrained by enum of allowed values. Within the body there is returned set of values which are supported**.**

400 - Bad Request	aplication/json	`{`
		`"code": "400004",`
		`"paramNames": [<list_of_params_in_conflict>],`
		`"paramValues": [<list_of_param_values_in_conflict>],`
		`"message": "Parameters values are in conflicting state."`
		`}`
			Issued when values of parameters conflict with each other. E.g. when you have specified minDate value which greater than maxDate value. In body you'll get the names and values of parameters which are in conflicting state.

400 - Bad Request	aplication/json	`{`
		`"code": "400005",`
		`"paramName": "<param_name>",`
		`"paramValue": "<value_being_supplied>",`
		`"message": "Unable to parse parameter value"`
		`}`
			Issued when supplied parameter value has wrong format. This could happen if you have supplied value
			• which have different type than expected
			• which format does not correspond expected one

400 - Bad Request	aplication/json	`{`
		`"code": "400006",`
		`"paramName": "<param_name>",`
		`"message": "Unable to parse parameter value"`
		`}`
			There is present parameter which is not expected in request.

400 - Bad Request	aplication/json	`{`
		`"code": "400007",`
		`"paramName": "<name_of_header>",`
		`"detectedValue": "<supplied_value>",`
		`"message": "<details>"`
		`}`
			The code is returned if system has failed to recognize supplied charset value. paramName will hold name of header which was selected for getting charset, detected value is the value which was parsed from the header.

400 - Bad Request	aplication/json	`{`
		`"code": "400008",`
		`"message": "Parameters <list1_of_params_in_conflict> cannot be set simultaneously`
		`with parameters <list2_of_params_in_conflict>",`
		`"paramNames1":[<list1_of_params_in_conflict>],`
		`"paramNames2":[<list2_of_params_in_conflict>]`
		`}`
			Issued when some parameters cannot be used together with another parameters by a logical conflict.

401 - Unauthorized	application/json	`{ "code": "401000", "message": "<error message>" }`	Unauthorized access attempt

401 - Unauthorized	application/json	`{ "code": "401001", "message": "<error message>" }`	Authentication token in request is not valid. This can also be returned for already discarded tokens (logout was called on them or discarded by inactivity timeout)
401 - Unauthorized	application/json	`{ "code": "401002", "message": "Combination of supplied user name or password is not recognized" }`	Supplied combination of user name and password is not recognized
403 - Forbidden	application/json	`{ "code": "403000", "message": "<error message>" }`	Unable to check validity of access, or access is denied
403 - Forbidden	application/json	`{ "code": "403001", "message": "Result '<id>' is locked" }`	Issued when trying to get results which are locked.
404 - Not Found	application/json	`{ "code": "404000", "message": "<error message>" }`	Resource cannot be located by given URL
405- Method not Allowed	application/json	`{ "code": "405000", "message": "<error message>" }`	Resource could not be accessed using given HTTP method. Ex. attempt to POST to read only resource was done.
409 - Conflict	application/json	`{ "code": "409000", "currentVersionCode" : "<actual_value_of_version_code>", "message": "Unable to modify entity. Supplied version code 'XXX' does not match actual one." }`	This status is usually returned by resource modification requests in case if you are trying to modify version which is not actual. E.g. you are trying to modify the state of entity which was modified by someone else. Please refer Handling conflicts section for getting more info.
412 - Precondition failed	application/json	`{ "code": "412000", "message": "Header \"X-Auth-Token\" is absent in the request." }`	In order to process your request you need to provide API Key as **`X-Auth-Token`** header value. Request does not hold the header.
412 - Precondition failed	application/json	`{ "code": "412001", "message": "Unsupported Content type defined in header: <content_type>" }`	Given Content type is not supported by enpoint you are accessing
413 - Payload too large	application/json	`{ "code": "413000", "message": "<Error description>" }`	Unable to process your request as payload supplied with it is too huge for requested operation to handle it.
500 - Internal Server Error	application/json	`{ "code": "500000", "message": "<error message>" }`	Unexpected error has happened during request processing. Please contact Itesco with error report.
500 - Internal Server Error	text/html	Web container specific payload	Web container internal error. Please contact Itesco with error report.
503 - Service Unavailable	application/json	`{ "code":"503000", "message":"Authentication service is unavailable. Please try again later." }`	The server is currently unable to handle the request due to temporary overloading or maintenance of the server. The implication is that this is a temporary condition which will be alleviated after some delay. See value of 'Retry-After' header for recommended delay before retry attempt

Methods

API consists of methods which you call using HTTP/HTTPS. In sections below the paths to method endpoints are defined as relative path starting after base web API path. So the complete path would be a composition of base path and relative path.

For example, if the API is available by URL **https://icatch3.icatch.se/api** and method URL is stated as **/login**, you should use **https://icatch3.icatch.se/api/login** as an URL for accessing the method endpoint.

Please consider that each request may return an error response. In that case the response status would be 4XX, body holds data in application/json format. Some ttimes additional headers give information in addition to body according to common HTTP errors processing standards.

Down below you can find the summary table of available methods which should give you a high level understanding on methods being exposed. For details on each particular method, please refer the link to the method description. Details section describes parameters, requests and responses specific for each particular method. You should also consider that besides specific error codes there could be returned one of common error codes in case of common failure

	URL	HTTP Method(s)	Request content type(s)	Response content type(s)	Authorized as	Response Specific Status Codes	Description
1	/ping	GET	<no body>	application/json Version Info	not authorized	200	Responds with API version metada on every request
2	/login	POST	application/json Credentials	application/json Authentication Result	not authorized	200	Accepts your login data and supplies authentication token if login was successful
3	/logout	POST	<no body>	<no answer supplied>	PROSP_PRIVATE or PROSP_COMPANY or GROUP_ADMIN	204	Gets the authentication token from the header and invalidates it.
4	/privateProspects/settingsTemplate[?params]	GET	<no body>	application/json list of Private Prospects Setting	PROSP_PRIVATE	200, 204	Returns the list of settings for private prospects queries for a user. If there is no settings for user there is 204 status returned and no body is supplied. Result is ordered by template name in alphabetical order. Following request URL parameters are supported: • startIndex={index_value} • count={count_value} • name={name_pattern} • dateFrom={dateFrom} • dateTo={dateTo} • id={id}
5	/privateProspects/settingsTemplate/{id}	GET	<no body>	application/json Private Prospects Settings	PROSP_PRIVATE	200, 404	Returns private prospects settings by ID or 404 if not found.Please note that BLOB structures will not be returned by plain get request. In order to get access to blob attributes (like exclusions, address filter), please use corresponding sub-url.
6	/privateProspects/settingsTemplate	POST	application/json, Private Prospects Settings	application/json, Private Prospects Settings	PROSP_PRIVATE	200, 201	Creates new private prospects settings and returns the representation of it as well as ID.
7	/privateProspects/settingsTemplate/{id}/copy	POST	application/json, Copy Action Parameters	application/json, Private Prospects Settings	PROSP_PRIVATE	200, 404	Creates a copy of private prospects settings referenced by ID in URL. Copy operation parameters are supplied in body.
8	/privateProspects/settingsTemplate/{id}	PUT	application/json, Private Prospects Settings	application/json, Private Prospects Setting	PROSP_PRIVATE	200, 404, 409	Updates existing private prospects settings with representation supplied within the body. Returns updated settings current representation or 404 if there is no setting with such ID.
9	/privateProspects/settingsTemplate/{id}	DELETE	<no body >	<no body supplied>	PROSP_PRIVATE	200, 404	Deletes private prospects settings referenced by its ID or returns 404 if it has already been deleted.
10	/privateProspects/settingsTemplate/{id}/exclusions	GET	<no body>	Depends on 'Accept' header value Private Persons Exclusions	PROSP_PRIVATE	200, 204	Returns the stream of exclusions associated with private prospects settings template. Stream is returned as CSV (use header "Accept: text/csv"). 204 is returned in case if there is no exclusions defined for this particular settings.
11	/privateProspects/settingsTemplate/{id}/exclusions	DELETE	<no body>	<no body supplied>	PROSP_PRIVATE	204,404	Deletes exclusions associated with private prospects settings template.
12	/privateProspects/settingsTemplate/{id}/allowedAddresses	DELETE	<no body>	<no body supplied>	PROSP_PRIVATE	204,404	Deletes address filter associated with private prospects settings template.
13	/privateProspects/settingsTemplate/{id}/exclusions	POST	multipart/form-data, Private Persons Exclusions	<no body supplied>	PROSP_PRIVATE	204, 404	You can associate exclusions with your private prospects settings using this method. If there were previously stored exclusions they will be replaced with what you have supplied. Request should be multipart with exclusions stored in CSV format. Part containing exclusions should be called '**file**' and have text/csv content type.
14	/privateProspects/settingsTemplate/{id}/exclusions?fileId={preloaded_file_id}	POST	<No_body>	<no body supplied>	PROSP_PRIVATE	204, 404	You can associate exclusions with your private prospects settings using this method. If there were previously stored exclusions they will be replaced with what you have supplied. Request should refer pre-loaded exclusions file ID on file store by parameter **fileId**. Following request URL parameters are supported: • fileId=<id_of_file_on_storage>
15	/privateProspects/settingsTemplate/{id}/allowedAddresses	POST	multipart/form-data, Address Criteria List	<no body supplied>	PROSP_PRIVATE	204, 404	You can associate address filter with your private prospects settings using this method. If there were previously stored address filter it will be replaced with what you have supplied. Request should be multipart with address filter stored in CSV format. Part containing address filter should be called '**file**' and have text/csv content type.
16	/privateProspects/settingsTemplate/{id}/allowedAddresses?field={preloaded_file_id}	POST	<No_body>	<no body supplied>	PROSP_PRIVATE	204, 404	You can associate address filter with your private prospects settings using this method. If there were previously stored address filter it will be replaced with what you have supplied. Request should refer pre-loaded exclusions file ID on file store by parameter **fileId**. Following request URL parameters are supported: • fileId=<id_of_file_on_storage>
17	/privateProspects/settingsTemplate/{id}/allowedAddresses	GET	<no body>	Depends on 'Accept' header value Address Criteria List	PROSP_PRIVATE	200, 204, 404	Returns the stream of allowed addresses stored with the private prospects settings. Stream is returned as CSV (use header "Accept: text/csv"). 204 is returned in case if there is no addressFilter defined for this particular settings.
18	/privateProspects/queries/dryRun	POST	multipart/form-data Query Definition	application/json, Dry Run Result	PROSP_PRIVATE	200, 404	Performs "dry-run" by parameters specified in private prospects settings. Returns the number of results you should get if you issue the same request to /queries URL specifying the same private prospects settings ID and supplying the same overrides of parameters within the body. As for the /queries URL body can hold the parameters which you would like to override for this specific run.

19	/privateProspects/queries/dryRun[?exclusionsFileId={fileId}][&addressFilterFileId={fileId}]	POST	application/json, Query Definition without exclusions/address filters part	application/json, Dry Run Result	PROSP_PRIVATE	200, 404	Performs "dry-run" by parameters specified in private prospects settings. Returns the number of results you should get if you issue the same request to /queries URL specifying the same settings ID and supplying the same overrides of parameters within the body. As for the /queries URL body can hold the parameters which you would like to override for this specific run. Request parameters point to ids of exclusions and address filters. Both are non mandatory. Following request URL parameters are supported: • exclusionsFileId=<id_of_file_on_storage> • addressFilterFileId =<id_of_file_on_storage>
20	/privateProspects/queries	GET	<no body>	application/json, List of Queries	PROSP_PRIVATE	200, 204	Returns the list of queries run against private prospects settings for specific user. If there is no queries for user there is 204 status returned and no body is supplied.
21	/privateProspects/queries/{id}	GET	<no body>	application/json, List of Queries	PROSP_PRIVATE	200, 404	Returns a query run against private prospects settings by query ID or 404 if not found.Please note that BLOB structures will not be returned by plain get request. In order to get access blob attributes, please use corresponding sub-url. Status of query is returned in scope of the response.
22	/privateProspects/queries	POST	multipart/form-data Query Definition	application/json, Query	PROSP_PRIVATE	200	Posts new private prospects query for execution. It is expected that body will hold the reference to the query setting ID and query name. Optionally body can hold attributes of query settings which should override parameter values specified within the query. This also relates to attributes which are defined as additional parts of multipart request.
23	/privateProspects/queries[?exclusionsFileId={fileId}][&addressFilterFileId={fileId}]	POST	application/json, Query Definition without exclusions/address filters part	application/json, Query	PROSP_PRIVATE	200, 404	Posts new private prospects query for execution. It is expected that body will hold the reference to the query setting ID and query name. Optionally body can hold attributes of query settings which should override parameter values specified within the query. Note that this request is not a multipart and counts on pre-loaded files IDs. Following request URL parameters are supported: • exclusionsFileId=<id_of_file_on_storage> • addressFilterFileId =<id_of_file_on_storage>
24	/privateProspects/queries/{id}/settings	GET	<no body>	application/json Private Prospects Settings	PROSP_PRIVATE	200, 204	Returns private prospects settings instance by query id. Once a query is submitted for execution the query entity is returned as the result. It contains id of the query. Then the id can be used to obtain settings instance associated with the query.
25	/privateProspects/settingsInstance/{id}	GET	<no body>	application/json Private Prospects Settings	PROSP_PRIVATE	200, 204	Returns private prospects settings instance by it's id. Once a query is submitted for execution the query entity is returned as the result. It contains id of settings instance associated with the query. Then the id can be used to obtain settings instance.
26	/privateProspects/results	GET	<no body>	application/json List of Private Prospects Query Result	PROSP_PRIVATE	200, 204	Returns the list of results of private prospects queries which have been completed. Note, this would be only metadata about your results including number of answers, link to query, completion date, completion status, etc. 204 is returned if you have no results available Results are ordered from the most recent to most eldest ones. Following request URL parameters are supported: • startIndex={index_value} • count={count_value} • queryId={queryId}
27	/privateProspects/results/{id}	GET	<no body>	application/json, Private Prospects Query Result	PROSP_PRIVATE	200, 404	Returns result for a specific private prospects query or 404 if there is no result for that query.
28	/privateProspects/settingsInstance/{id}/exclusions	GET	<no body>	Depends on 'Accept' header value Private Persons Exclusions	PROSP_PRIVATE	200, 204, 404	Returns the exclusions being used while running specified query. Please note that this could be composition of persistent exclusions associated with setting/query and additional exclusions supplied with query. Only those rows which were recognized in input are present in result of this request.
29	/privateProspects/settingsInstance/{id}/allowedAddresses	GET	<no body>	Depends on 'Accept' header value Address Criteria List	PROSP_PRIVATE	200, 204, 404	Returns the allowed addresses being used while running specified query. Only those rows which were recognized in input are present in result of this request.
30	/privateProspects/results/{id}/values?startIndex={index_value}&count={count_value}	GET	<no body>	Private Prospects Query Result Value	PROSP_PRIVATE	200, 204, 404	Returns specified subset of private prospects query results. Results could be returned as CSV (use header "Accept: text/csv") or as JSON Array (use header "Accept: application/json"). 404 is returned in case if your query is not yet finished. 204 is returned if your result does not contain any single answer. Parameters of request allow to control subset scope.
31	/privateProspects/results/{id}/valuesStream	GET	<no body>	Depends on 'Accept' header value Private Prospects Query Result Value	PROSP_PRIVATE	200, 204, 404	Returns the stream which holds results of private prospects query. Stream is returned as CSV. 404 is returned in case if your query is not yet finished. 204 is returned if your result does not contain any single answer.
32	/privateProspects/results/{id}	PUT	application/json Private Prospects Query Result	application/json, Private Prospects Query Result	PROSP_PRIVATE	200, 404	Updates attributes of private prospects query results which are allowed to update. Please refer Private Prospects Query Result for details on that. Using this method you can mark the results as hidden, exclude from consequent query calls, etc.
33	/privateProspects/results/{id}	DELETE	<no body>	<no body supplied>	PROSP_PRIVATE	201,404	Marks query associated with results as deleted. The one could be restored if needed. After months deleted queries and associated results and settings are permanently deleted.

34	/privateProspects/exclusions[?maxErrorsCount=<>]	POST	multipart/form-data, Private Persons Exclusions	application/json - Private Persons Exclusions Check Result text/csv - Private Persons Exclusions Parse Error report	PROSP_PRIVATE	200	Parses your input and provides private prospects query parse result which holds result of parsing. Within the result you will be able to find number of recognized entries grouped by type as well as list of unrecognized entries with reference to original input by index. Input is taken from the body part of the report. Answer depends on Accept header value. • If you are requesting application/json you'll be getting report as JSON with summaries on parse. Also you can limit output with maxErrorsCount parameter here. • If you are requesting text/csv - you'll be supplied with complete list of lines which we have failed to parse in TAB Separated value format. Max errors count is not considered in this case. Following request URL parameters are supported: • maxErrorsCount={limit_value}
35	/privateProspects/exclusionsParseReport?fileId={preloaded_file_id}[&maxErrorsCount=<>]	GET	<No Body>	application/json, Private Persons Exclusions Check Result text/csv - Private Persons Exclusions Parse Error report	PROSP_PRIVATE	200, 404	Parses your input and provides private prospects query parse result which holds result of parsing. Within the result you will be able to find number of recognized entries grouped by type as well as list of unrecognized entries with reference to original input by index. Input is taken from File preliminary uploaded to File Store and identified by parameter **fileId**. Answer depends on Accept header value. • If you are requesting application/json you'll be getting report as JSON with summaries on parse. Also you can limit here output with maxErrorsCount parameter. • If you are requesting text/csv - you'll be supplied with complete list of lines which we have failed to parse in TAB Separated value format. Max errors count is not considered in this case. Following request URL parameters are supported: • maxErrorsCount={limit_value} • fileId=<id_of_file_on_storage>
36	/privateProspects/allowedAddresses	POST	multipart/form-data, Address Criteria List	application/json - Address Criteria List Check Result text\csv - Address Criteria List parse error report	PROSP_PRIVATE	200	Parses your input and provides private prospects query parse result which holds result of parsing. Within the result you will be able to find number of valid, invalid and unrecognized items. All invalid and unrecognized items have references to original input by index. Answer depends on Accept header value. • If you are requesting application/json you'll be getting report as JSON with summaries on parse. Also you can limit here output with maxErrorsCount parameter. • If you are requesting text/csv - you'll be supplied with complete list of lines which we have failed to parse in TAB Separated value format. Max errors count is not considered in this case. Following request URL parameters are supported: • maxErrorsCount={limit_value}
37	/privateProspects/allowedAddressesParseReport?fileId={file_id}	GET	<No Body>	application/json, Address Criteria List Check Result	PROSP_PRIVATE	400	Parses your input and provides parse result which holds result of parsing. Within the result you will be able to find number of valid, invalid and unrecognized items. All invalid and unrecognized items have references to original input by index. Answer depends on Accept header value. • If you are requesting application/json you'll be getting report as JSON with summaries on parse. Also you can limit output with maxErrorsCount parameter here. • If you are requesting text/csv - you'll be supplied with complete list of lines which we have failed to parse in TAB Separated value format. Max errors count is not considered in this case. Following request URL parameters are supported: • maxErrorsCount={limit_value} • fileId=<id_of_file_on_storage>
38	/privateProspects/queryInfo?[params]	GET	<no body>	application/json, Person Query Info	PROSP_PRIVATE	200, 204	Provides extended information about private prospects queries user has issued. Allows extended filtering abilities comparing to separate resources for queries and results. Allows extracting information about several resources related to query in a single request. This could be useful when building some UI which provides information about query, result and settings at the same time. Result is ordered from the most recent to most eldest. Following request URL parameters are supported: • startIndex={index_value} • count={count_value} • queryName={qName} • queryPostDateFrom={dateFrom} • queryPostDateTo={dateTo} • queryStatus=[{NOT_STARTED|IN_PROGRESS|COMPLETED}, ...] • resultConsidered={true/false} • resultHidden={yes/no}

39	/companyProspects/settingsTemplate[?params]	GET	<no body>	application/json list of Company Prospects Setting	PROSP_COMPANY	200, 204	Returns the list of settings for company prospects queries for a user. If there is no settings for user there is 204 status returned and no body is supplied. Result is ordered by template name in alphabetical order. Following request URL parameters are supported: • startIndex={index_value} • count={count_value} • name={name_pattern} • dateFrom={dateFrom} • dateTo={dateTo} • id={id}
40	/companyProspects/settingsTemplate/{id}	GET	<no body>	application/json Company Prospects Setting	PROSP_COMPANY	200, 404	Returns company prospects settings by ID or 404 if not found.Please note that BLOB structures will not be returned by plain get request. In order to get access to blob attributes, please use corresponding sub-url.
41	/companyProspects/settingsTemplate	POST	application/json, Company Prospects Setting	application/json, Company Prospects Settings	PROSP_COMPANY	201	Creates new company prospects setting and returns the representation of it as well as ID.
42	/companyProspects/settingsTemplate/{id}/copy	POST	application/json, Copy Action Parameters	application/json, Company Prospects Setting	PROSP_COMPANY	200, 404	Creates a copy of company prospects settings referenced by ID in URL. Copy operation parameters that are supplied in body.
43	/companyProspects/settingsTemplate/{id}[?exclusionsFileId={fileId}][&addressFilterFileId={fileId}]	PUT	application/json, Company Prospects Settings	application/json, Company Prospects Settings	PROSP_COMPANY	200, 404, 409	Updates existing setting with representation supplied within the body. Returns updated current representation of company prospects settings or 404 if there is no setting with such ID.
44	/companyProspects/settingsTemplate/{id}	DELETE	<no body >	<no body supplied>	PROSP_COMPANY	200, 404	Deletes company prospects setting referenced by its ID or returns 404 if it has already been deleted.
45	/companyProspects/settingsTemplate/{id}/exclusions	GET	<no body>	Depends on 'Accept' header value Company Prospects Setting	PROSP_COMPANY	200, 204	Returns the stream of exclusions associated with company prospects settings template. Stream is returned as CSV (use header "Accept: text/csv"). 204 is returned in case if there is no exclusions defined for this particular settings.
46	/companyProspects/settingsTemplate/{id}/exclusions	DELETE	<no body>	<no body supplied>	PROSP_COMPANY	204,404	Deletes exclusions associated with company prospects template.
47	/companyProspects/settingsTemplate/{id}/allowedAddresses	DELETE	<no body>	<no body supplied>	PROSP_COMPANY	204,404	Deletes address filter associated with company prospects template.
48	/companyProspects/settingsTemplate/{id}/exclusions	POST	multipart/form-data, Company Exclusions	<no body supplied>	PROSP_COMPANY	204, 404	You can associate exclusions with your company prospects settings using this method. If there were previously stored exclusions they will be replaced with what you have supplied. Request should be multipart with exclusions stored in CSV format. The part containing exclusions should be called '**file**' and have text/csv content type.
49	/companyProspects/settingsTemplate/{id}/exclusions?fileId={preloaded_file_id}	POST	<No_body>	<no body supplied>	PROSP_COMPANY	204, 404	You can associate exclusions with your company prospects settings using this method. If there were previously stored exclusions they will be replaced with what you have supplied. Request should refer to pre-loaded exclusions file ID on file store by parameter **fileId**. Following request URL parameters are supported: • fileId=<id_of_file_on_storage>
50	/companyProspects/settingsTemplate/{id}/allowedAddresses	POST	multipart/form-data, Address Criteria List	<no body supplied>	PROSP_COMPANY	204, 404	You can associate address filter with your company prospects settings using this method. If there were previously stored address filter it will be replaced with what you have supplied. Request should be multipart with address filter stored in CSV format. Part containing address filter should be called '**file**' and have text/csv content type.
51	/companyProspects/settingsTemplate/{id}/allowedAddresses?fileId={preloaded_file_id}	POST	<No_body>	<no body supplied>	PROSP_COMPANY	204, 404	You can associate address filter with your company prospects settings using this method. If there were previously stored address filter it will be replaced with what you have supplied. Request should refer to pre-loaded exclusions file ID on file store by parameter **fileId**. Following request URL parameters are supported: • fileId=<id_of_file_on_storage>
52	/companyProspects/settingsTemplate/{id}/allowedAddresses	GET	<no body>	Depends on 'Accept' header value Address Criteria List	PROSP_COMPANY	200, 204, 404	Returns the stream of allowed addresses stored with the company prospects settings. Stream is returned as CSV (use header "Accept: text/csv"). 204 is returned in case if there is no addressFilter defined for this particular settings.
53	/companyProspects/queries/dryRun	POST	multipart/form-data Query Definition	application/json, Company Query Dry Run Result	PROSP_COMPANY	200, 404	Performs "dry-run" by parameters specified in company prospects settings. Returns the number of results you should get if you issue the same request to /queries URL specifying the same settings ID and supplying the same overrides of parameters within the body. As for the /queries URL body can hold the parameters which you would like to override for this specific run.
54	/companyProspects/queries/dryRun[?exclusionsFileId={fileId}][&addressFilterFileId={fileId}]	POST	application/json, Query Definition without exclusions/address filters part	application/json, Company Query Dry Run Result	PROSP_COMPANY	200, 404	Performs "dry-run" by parameters specified in company prospects settings. Returns the number of results you should get if you issue the same request to /queries URL specifying the same settings ID and supplying the same overrides of parameters within the body. As for the /queries URL body can hold the parameters which you would like to override for this specific run. Request parameters point to ids of exclusions and address filters. Both are non mandatory. Following request URL parameters are supported: • exclusionsFileId=<id_of_file_on_storage> • addressFilterFileId =<id_of_file_on_storage>
55	/companyProspects/queries	GET	<no body>	application/json, List of Queries	PROSP_COMPANY	200, 204	Returns the list of company prospects queries for specific user. If the user doesnot have any queries lists then 204 status is returned and no body is supplied.

56	/companyProspects/queries/{id}	GET	<no body>	application/json, Query	PROSP_COMPANY	200, 404	Returns company prospects query by ID or 404 if not found.Please note that BLOB structures will not be returned by plain get request. In order to get access to blob attributes, please use corresponding sub-url. Status of query is returned in scope of the response.
57	/companyProspects/queries	POST	multipart/form-data Query Definition	application/json, Query	PROSP_COMPANY	200	Posts a new company prospects query for execution. It is expected that body will hold the reference to the query setting ID and query name. Optionally body can hold attributes of query settings which should override parameter values specified within the query. This also relates to attributes which are defined as additional parts of multipart request.
58	/companyProspects/queries[?exclusionsFileId={fileId}][&addressFilterFileId={fileId}]	POST	application/json, Query Definition without exclusions/address filters part	application/json, Query	PROSP_COMPANY	200, 404	Posts a new company prospects query for execution. It is expected that body will hold the reference to the query setting ID and query name. Optionally body can hold attributes of query settings which should override parameter values specified within the query. Note that this request is not multipart and counts on pre-loaded files IDs. Following request URL parameters are supported: • exclusionsFileId=<id_of_file_on_storage> • addressFilterFileId =<id_of_file_on_storage>
59	/companyProspects/queries/{id}/settings	GET	<no body>	application/json Company Query Settings	PROSP_COMPANY	200, 204	Returns settings instance by company prospects query id. Once a query is submitted for execution the query entity is returned as the result. It contains id of of the query. Then the id can be used to obtain settings instance associated with the query.
60	/companyProspects/settingsInstance/{id}	GET	<no body>	application/json Company Query Settings	PROSP_COMPANY	200, 204	Returns company prospects settings instance by it's id. Once a query is submitted for execution the query entity is returned as the result. It contains id of settings instance associated with the query. Then the id can be used to obtain settings instance.
61	/companyProspects/results	GET	<no body>	application/json List of Company Prospects Query Result	PROSP_COMPANY	200, 204	Returns the list of results of company prospects queries which have been completed. Note, this would be only metadata about your results including number of answers, link to query, completion date, completion status, etc. 204 is returned if you have no results available. Result is ordered from the most recent to most eldest one. Following request URL parameters are supported: • startIndex={index_value} • count={count_value} • queryId={queryId}
62	/companyProspects/results/{id}	GET	<no body>	application/json, Company Prospects Query Result	PROSP_COMPANY	200, 404	Returns result for a specific company prospects query or 404 if there is no result for that query.
63	/companyProspects/settingsInstance/{id}/exclusions	GET	<no body>	Depends on 'Accept' header value Company Exclusions	PROSP_COMPANY	200, 204, 404	Returns the exclusions being used when running the query with specified settings. Please note that this could be composition of persistent exclusions associated with setting/query and additional exclusions supplied with query. Only those rows which were recognized in input are present in result of this request.
64	/companyProspects/settingsInstance/{id}/allowedAddresses	GET	<no body>	Depends on 'Accept' header value Address Criteria List	PROSP_COMPANY	200, 204, 404	Returns the allowed addresses being used when running the query with specified settings. Only those rows which were recognized in input are present in result of this request.
65	/companyProspects/results/{id}/values?startIndex={index_value}&count={count_value}	GET	<no body>	Company Prospects Query Result Value	PROSP_COMPANY	200, 204, 404	Returns specified subset of company prospects query results. Results could be returned as CSV (use header "Accept: text/csv") or as JSON Array (use header "Accept: application/json"). 404 is returned in case if your query is not yet finished. 204 is returned if your result does not contain any single answer. Parameters of request allow to control subset scope.
66	/companyProspects/results/{id}/valuesStream	GET	<no body>	Depends on 'Accept' header value Company Prospects Query Result Value	PROSP_COMPANY	200, 204, 404	Returns the stream which holds results of company prospects query. Stream is returned as CSV. 404 is returned in case if your query is not yet finished. 204 is returned if your result does not contain any single answer.
67	/companyProspects/results/{id}	PUT	application/json Private Prospects Query Result	application/json, Company Prospects Query Result	PROSP_COMPANY	200, 404	Updates attributes of company prospects results which are allowed to update. Please refer Company Prospects Query Result for details on that. Using this method you can mark the results as hidden, exclude from consequent query calls, etc.
68	/privateProspects/results/{id}	DELETE	<no body>	<no body supplied>	PROSP_COMPANY	201, 404	Marks query associated with results as deleted. The one could be restored if needed. After months deleted queries and associated results and settings are permanently deleted.
69	/companyProspects/exclusions[?maxErrorCount=<>]	POST	multipart/form-data, Private Persons Exclusions	application/json, Company Exclusions Check Result	PROSP_COMPANY	200	Parses your input and provides company prospects query parse result which holds result of parsing. Within the result you will be able to find number of recognized entries grouped by type as well as list of unrecognized entries with reference to original input by index. Input is taken from the body part of the report. Answer depends on Accept header value. • If you are requesting application/json you'll be getting report as JSON with summaries on parse. Also you can limit here output with maxErrorsCount parameter. • If you are requesting text/csv - you'll be supplied with complete list of lines which we have failed to parse in TAB Separated value format. Max errors count is not considered in this case. Following request URL parameters are supported: • maxErrorsCount={limit_value}

70	/companyProspects/exclusionsParseReport?fileId={preloaded_file_id}[&maxErrorCount=<>]	GET	<No Body>	application/json - Company Exclusions Check Result text/csv - Company Exclusions Parse Error report	PROSP_COMPANY	200, 404	Parses your input and provides company prospects query parse result which holds result of parsing. Within the result you will be able to find number of recognized entries grouped by type as well as list of unrecognized entries with reference to original input by index. Input is taken from File preliminary uploaded to File Store and identified by parameter **fileId**. Answer depends on Accept header value. • If you are requesting application/json you'll be getting report as JSON with summaries on parse. Also you can limit here output with maxErrorsCount parameter. • If you are requesting text/csv - you'll be supplied with complete list of lines which we have failed to parse in TAB Separated value format. Max errors count is not considered in this case. Following request URL parameters are supported: • maxErrorsCount={limit_value} • fileId=<id_of_file_on_storage>
71	/companyProspects/allowedAddresses	POST	multipart/form-data, Address Criteria List	application/json - Address Criteria List Check Result text/csv - Company Exclusions Parse Error report	PROSP_COMPANY	200	Parses your input and provides company prospects query parse result which holds result of parsing. Within the result you will be able to find number of valid, invalid and unrecognized items. All invalid and unrecognized items have references to original input by index. Answer depends on Accept header value. • If you are requesting application/json you'll be getting report as JSON with summaries on parse. Also you can limit here output with maxErrorsCount parameter. • If you are requesting text/csv - you'll be supplied with complete list of lines which we have failed to parse in TAB Separated value format. Max errors count is not considered in this case. Following request URL parameters are supported: • maxErrorsCount={limit_value}
72	/companyProspects/allowedAddressesParseReport?fileId={file_id}	GET	<No Body>	application/json, Address Criteria List Check Result	PROSP_COMPANY	400	Parses your input and provides company prospects query parse result which holds result of parsing. Within the result you will be able to find number of valid, invalid and unrecognized items. All invalid and unrecognized items have references to original input by index. Following request URL parameters are supported: • maxErrorsCount={limit_value} • fileId=<id_of_file_on_storage>
73	/companyProspects/queryInfo?[params]	GET	<no body>	application/json, Company Query Info	PROSP_COMPANY	200, 204	Provides extended information about company prospects queries user has issued. Allows extended filtering abilities comparing to separate resources for queries and results. Allows extracting information about several resources related to query in a single request. This could be useful when building some UI which provides information about query, result and settings at the same time. Result is ordered from the most recent to most eldest one. Following request URL parameters are supported: • startIndex={index_value} • count={count_value} • queryName={qName} • queryPostDateFrom={dateFrom} • queryPostDateTo={dateTo} • queryStatus=[{NOT_STARTED|IN_PROGRESS|COMPLETED}, …] • resultConsidered={true/false} • resultHidden={yes/no}
74	/groupSearch/settingsTemplate[?params]	GET	<no body>	application/json list of Group Search Settings	GROUP_SEARCH	200, 204	Returns the list of settings of group searches for a user. If there is no settings for user there is returned 204 status and no body is supplied. Result is ordered by template name in alphabetical order. Following request URL parameters are supported: • startIndex={index_value} • count={count_value} • name={name_pattern} • dateFrom={dateFrom} • dateTo={dateTo} • id={id}
75	/groupSearch/settingsTemplate/{id}	GET	<no body>	application/json Group Search Settings	GROUP_SEARCH	200, 404	Returns group search settings by ID or 404 if not found.Please note that BLOB structures will not be returned by plain get request. In order to get access to blob attributes, please use corresponding sub-url.
76	/groupSearch/settingsTemplate	POST	application/json, Group Search Settings	application/json, Group Search Settings	GROUP_SEARCH	200, 201	Creates new group search settings and returns its representation as well as ID.
77	/groupSearch/settingsTemplate/{id}/copy	POST	application/json, Copy Action Parameters	application/json, Group Search Settings	GROUP_SEARCH	200, 404	Creates a copy of group search settings referenced by ID in URL. Copy operation parameters are supplied in body.
78	/groupSearch/settingsTemplate/{id}	PUT	application/json, Group Search Settings	application/json, Group Search Settings	GROUP_SEARCH	200, 404, 409	Updates existing group search settings with representation supplied within the body. Returns updated settings current representation or 404 if there is no setting with such ID.
79	/groupSearch/settingsTemplate/{id}	DELETE	<no body>	<no body supplied>	GROUP_SEARCH	200, 404	Deletes group search settings referenced by its ID or returns 404 if it has already been deleted.

80	/groupSearch/queries/dryRun	POST	multipart/form-data Group Query Definition	application/json, Group Query Dry Run Result	GROUP_SEARCH	200, 404	Performs "dry-run" by parameters specified in group search settings and questions you have supplied in the body of the request. Returns the number of results you should get if you issue the same request to /queries URL specifying the same group search settings ID and supplying the same body. Query input is supplied as the separate part of multipart request under name "queryInput". Query Definition is placed as 'setting' part.
81	/groupSearch/queries/dryRun?inputFileId={fileId}	POST	application/json, Group Query Definition without questions part	application/json, Group Query Dry Run Result	GROUP_SEARCH	200, 404	Performs "dry-run" by parameters specified in group search settings and questions you have supplied in the body of request. Returns the number of results you should get if you issue the same request to /queries URL specifying the same group search settings ID and supplying the same body. Request parameters point to id of file which contains questions. It is mandatory part Following request URL parameters are supported: • inputFileId=<id_of_file_on_storage>
82	/groupSearch/queries	GET	<no body>	application/json, List of Queries	GROUP_SEARCH	200, 204	Returns the list of queries run against group search settings for specific user. If there is no queries for user there is returned 204 status and no body is supplied.
83	/groupSearch/queries/{id}	GET	<no body>	application/json, Queries	GROUP_SEARCH	200, 404	Returns a query run against group search settings by query ID or 404 if not found.Please note that BLOB structures will not be returned by plain get request. In order to get access to blob attributes, please use corresponding sub-url. Status of query is returned in scope of the response.
84	/groupSearch/queries	POST	multipart/form-data Group Query Definition	application/json, Query	GROUP_SEARCH	200	Posts new group search query for execution. It is expected that body will hold the reference to the query setting ID and query name. Optionally body can hold attributes of query settings which should override parameter values specified within the query. This also relates to attributes which are defined as additional parts of multipart request. Query input is supplied as the separate part of multipart request under name "queryInput". Query Definition is placed as 'setting' part.
85	/groupSearch/queries?inputFileId={fileId}[&inputFileName{fileName}]	POST	application/json, Group Query Definition without questions part	application/json, Query	GROUP_SEARCH	200, 404	Posts new group search query for execution. It is expected that body will hold the reference to the query setting ID, query name. Optionally body can hold attributes of query settings which should override parameter values specified within the query. Note that this request is not a multipart and counts on pre-loaded files IDs. Following request URL parameters are supported: • inputFileId=<id_of_file_on_storage> • inputFileName - name of file (an optional parameter. It will be used when you'll try to download input. It will be generated if not specified)
86	/groupSearch/queries/{id}/settings	GET	<no body>	application/json Group Search Settings	GROUP_SEARCH	200, 204	Returns group search settings instance by query id. Once a query is submitted for execution the query entity is returned as the result. It contains id of of the query. Then the id can be used to obtain settings instance associated with the query.
87	/groupSearch/settingsInstance/{id}	GET	<no body>	application/json Group Search Settings	GROUP_SEARCH	200, 204, 404	Returns group search settings instance by it's id. Once a query is submitted for execution the query entity is returned as the result. It contains id of settings instance associated with the query. Then the id can be used to obtain settings instance.
88	/groupSearch/settingsInstance/{id}/queryInput	GET	multipart/form-data, Input posted along with query	Data stream	GROUP_SEARCH	200, 204, 404	Returns Questions you have posted for running the query identified by its settings id.
89	/groupSearch/settingsInstance/{id}/queryInput/questionsParseReport	GET	<No Body>	text/csv - Group Query Parse error report	GROUP_SEARCH	200, 204, 404	Returns results of parsing your input which was supplied with a posted query. Input is returned in CSV format (Tab separated). Returned charset depends on Accept-Charset Header.
90	/groupSearch/results[?startIndex={index_value}&count={count_value}] /groupSearch/results[?queryId={queryId}]	GET	<no body>	application/json List of Group Query Result	GROUP_SEARCH	200, 204	Returns the list of results of group queries which have been completed. Note, this would be only metadata about your results including number of answers, link to query, completion date, completion status, etc. 204 is returned if you have no results available. Results are ordered from the most recent to most eldest ones. Following request URL parameters are supported: • startIndex={index_value} • count={count_value} • queryId={queryId}
91	/groupSearch/results/{id}	GET	<no body>	application/json, Group Query Result	GROUP_SEARCH	200, 404	Returns result for a specific group query or 404 if there is no result for that query.
92	/groupSearch/results/{id}/values[?startIndex={index_value}&count={count_value}&transpose={true/false}&outNonParsed={true/false}]	GET	<no body>	Depends on 'Accept' header value Group Query Result Value	GROUP_SEARCH	200, 204, 404	Returns specified subset of group query results. Results could be returned as CSV (use header "Accept: text/csv") or as JSON Array (use header "Accept: application/json"). 404 is returned in case if your query is not yet finished. 204 is returned if your result does not contain any single answer. Parameters of request allow to control subset scope. Under indexes specified within the query we understand indexes of questions answered. So giving request with startIndex =5 and count=10 means "I want to get answers starting from 5th question to 10th" Parameters: • startIndex={index_value} • count={count_value} • transpose - specify if you are getting results in a tabular form (ex. CSV) and would liketo have transpose rows by each question specified • outNonParsed - specify if you would like to keep lines with parse errors in output
93	/groupSearch/results/{id}/valuesStream[&transpose={true/false}&outNonParsed={true/false}]	GET	<no body>	Depends on 'Accept' header value Group Query Result Value	GROUP_SEARCH	200, 204, 404	Returns the stream which holds results of private prospects query. Stream is returned as CSV. 404 is returned in case if your query is not yet finished. 204 is returned if your result does not contain any single answer.
94	/groupSearch/results/{id}	PUT	application/json Group Query Result	application/json,Group Query Result	GROUP_SEARCH	200, 404	Updates attributes of group query results which are allowed to update. Please refer Group Query Result for details on that. Using this method you can mark the results as hidden or block for downloading.

95	/groupSearch/results/{id}	DELETE	<no body>	<no bidy supplied>	GROUP_SEARCH	201, 404	Marks query associated with results as deleted. The one could be restored if needed. After months deleted queries and associated results and settings are permanently deleted.
96	/groupSearch/questions[?maxErrorCount=<>]	POST	multipart/form-data, Group Search Questions	application/json - Group Questions Check Result text/csv - Group Query Parse error report	GROUP_SEARCH	200	Parses your input and provides parse result which describes result of parsing. Within the result you will be able to find which columns were recognized, which values contain parse errors and will not be recognized anyway as well as a subset of matching templates from user's available settings templates which can potentially give some answers to questions you have provided. Answer depends on Accept header value. • If you are requesting application/json you'll be getting report as JSON with summaries on parse. Also you can limit here output with maxErrorsCount parameter. • If you are requesting text/csv - you'll be supplied with complete list of lines which we have failed to parse in TAB Separated value format. maxErrorsCount is not considered in this case. Following request URL parameters are supported: • maxErrorsCount={limit_value}
97	/groupSearch/questionsParseReport?fileId={preloaded_file_id}[&maxErrorCount=<>]	GET	<No Body>	application/json - Group Questions Check Result text/csv - Group Query Parse error report	GROUP_SEARCH	200, 204, 404	Parses your input and provides parse result which descrfbes result of parsing. Within the result you will be able to find which columns were recognized, which values contains parse errors and will no be recognized anyway as well as a subset of matching templates from user's available settings templates which can potentially give some answers to questions you have provided. Input is taken from File preliminary uploaded to File Store and identified by parameter **fileId**. Answer depends on Accept header value. • If you are requesting application/json you'll be getting report as JSON with summaries on parse. Also you can limit here output with maxErrorsCount parameter. • If you are requesting text/csv - you'll be supplied with complete list of lines which we have failed to parse in TAB Separated value format. maxErrorsCount is not considered in this case. Following request URL parameters are supported: • maxErrorsCount={limit_value} • fileId=<id_of_file_on_storage>
98	/groupSearch/queryInfo?[params]	GET	<no body>	application/json, List of Group Query Infos	GROUP_SEARCH	200, 204	Provides extended information about group queries that user has issued. Allows extended filtering abilities comparing to separate resources for queries and results. Allows extracting information about several resources related to query in a single request. This could be useful when building some UI which provides information about query, result and settings at the same time. Result is ordered from the most recent to most eldest. Following request URL parameters are supported: • startIndex={index_value} • count={count_value} • queryName={qName} • queryPostDateFrom={dateFrom} • queryPostDateTo={dateTo} • queryStatus=[{NOT_STARTED|IN_PROGRESS|COMPLETED}, …] • queryId{id_of_query} - could be more than one.
99	/groupSearch/matchingTemplates[?startIndex={index_value}&count={count_value}] /groupSearch/matchingTemplates/{id}	GET	<no body>	application/json, List of Matching Templates	GROUP_SEARCH	200, 204	Provides a list of matching templates or a single template (if referred by ID) available for user issuing the request. The result of call should be used to build correct list of matching templates within group search settings. Result is ordered by matching template name. Following request URL parameters are supported: • startIndex={index_value} • count={count_value}
100	/groupSearch/features[?startIndex={index_value}&count={count_value}&withInFeatures={true/false}&withOutFeatures={true/false}]	GET	<no body>	application/json, List of Features	GROUP_SEARCH	200, 204	Returns the list of client specific questions pre-processing and answers post-processing features available for user issued the request. Results could be used to correctly form settings using user specific features. Result is ordered by user features ids. Following request URL parameters are supported: • startIndex={index_value} • count={count_value} • withInFeatures={true/false} • withOutFeatures{true/false}
101	/nixSearch/queries	GET	<no body>	application/json, List of Queries	NIX_TELE_SERVICE	200, 204	Returns the list of queries run against NIX search settings for specific user. If there is no queries for user there is returned 204 status and no body is supplied.
102	/nixSearch/queries/{id}	GET	<no body>	application/json, Queries	NIX_TELE_SERVICE	200, 404	Returns a query run against NIX search settings by query ID or 404 if not found.Please note that BLOB structures will not be returned by plain get request. In order to get access to BLOB attributes, please use corresponding sub-url. Status of query is returned in scope of the response.
103	/nixSearch/queries	POST	multipart/form-data NIX Query Definition	application/json, Query	NIX_TELE_SERVICE	200	Posts new NIX search query for execution. It is expected that body contains query name. Query input is supplied as the separate part of multipart request under name "queryInput"

104	/nixSearch/queries?inputFileId={fileId}[&inputFileName{fileName}]	POST	application/json, NIX Query Definition with out questions part	application/json, Query	NIX_TELE_SERVICE	200, 404	Posts new NIX search query for execution. It is expected that body contains query name. Note that this request is not a multipart and counts on pre-loaded files IDs. Following request URL parameters are supported: • inputFileId=<id_of_file_on_storage> • inputFileName - name of file (an optional parameter. It will be used when you'll try to download input. It will be generated if not specified)
105	/nixSearch/queries/{id}/settings	GET	<no body>	application/json NIX Query Settings	NIX_TELE_SERVICE	200, 204	Returns NIX search settings instance by query id. Once a query is submitted for execution the query entity is returned as the result. It contains id of of the query. Then the id can be used to obtain settings instance associated with the query.
106	/nixSearch/settingsInstance/{id}	GET	<no body>	application/json NIX Query Settings	NIX_TELE_SERVICE	200, 204, 404	Returns NIX search settings instance by it's id. Once a query is submitted for execution the query entity is returned as the result. It contains id of settings instance associated with the query. Then the id can be used to obtain settings instance.
107	/nixSearch/settingsInstance/{id}/queryInput	GET	multipart/form-data, Input posted along with query	Data stream	NIX_TELE_SERVICE	200, 204, 404	Returns the input you have posted for running the query identified by its settings id.
108	/nixSearch/settingsInstance/{id}/queryInput/inputParseReport	GET	<no body>	text/csv - NIX Query Parse error report	NIX_TELE_SERVICE	200, 204, 404	Returns results of parsing your input which was supplied with a posted query. Input is returned in CSV format (Tab separated). Returned charset depends on Accept-Charset Header.
109	/nixSearch/results[?startIndex={index_value}&count={count_value}] /nixSearch/results[?queryId={queryId}]	GET	<no body>	application/json List of NIX Query Result	NIX_TELE_SERVICE	200, 204	Returns the list of results of NIX queries which have been completed. Note, this would be only metadata about your results including number of answers, link to query, completion date, completion status, etc. 204 is returned if you have no results available. Results are ordered from the most recent to most eldest ones. Following request URL parameters are supported: • startIndex={index_value} • count={count_value} • queryId={queryId}
110	/nixSearch/results/{id}	GET	<no body>	application/json, NIX Query Result	NIX_TELE_SERVICE	200, 404	Returns result for a specific NIX query or 404 if there is no result for that query.
111	/nixSearch/results/{id}/values[?startIndex={index_value}&count={count_value}&combineWithInput={true/false}]	GET	<no body>	Depends on 'Accept' header value NIX Query Result Value	NIX_TELE_SERVICE	200, 204, 404	Returns specified subset of NIX query results. Results could be returned as CSV (use header "Accept: text/csv") or as JSON Array (use header "Accept: application/json"). 404 is returned in case if your query is not yet finished. 204 is returned if your result does not contain any single answer. Parameters of request allow to control subset scope. Parameters: • startIndex={index_value} • count={count_value} • combineWithInput={true/false} - only applicable for CSV
112	/nixSearch/results/{id}/valuesStream[?combineWithInput={true/false}]	GET	<no body>	Depends on 'Accept' header value NIX Query Result Value	NIX_TELE_SERVICE	200, 204, 404	Returns the stream which holds results of NIX query. Stream is returned as CSV. 404 is returned in case if your query is not yet finished. 204 is returned if your result does not contain any single answer. You can specify outNonParsedcombineWithInput parameter for getting answers combined to your input.
113	/nixSearch/results/{id}	PUT	application/json NIX Query Result	application/json, NIX Query Result	NIX_TELE_SERVICE	200, 404	Updates attributes of NIX query results which are allowed to update. Please refer NIX Query Result for details on that. Using this method you can mark the results as hidden or block for downloading.
114	/nixSearch/results/{id}	DELETE	<no body>	<no body supplied>	NIX_TELE_SERVICE	201, 404	Marks query associated with results as deleted. The one could be restored if needed. After months deleted queries and associated results and settings are permanently deleted.
115	/nixSearch/input[?maxErrorCount=<>]	POST	multipart/form-data, NIX Search Input	application/json - NIX Input Check Result text/csv - NIX Query Parse error report	NIX_TELE_SERVICE	200	Parses your input and provides parse result which describes result of parsing. Within the result you will be able to find which values contain parse errors and will not be recognized anyway. Answer depends on Accept header value. • If you are requesting application/json you'll be getting report as JSON with summaries on parse. Also you can limit here output with maxErrorsCount parameter. • If you are requesting text/csv - you'll be supplied with complete list of lines which we have failed to parse in TAB Separated value format. maxErrorsCount is not considered in this case. Following request URL parameters are supported: • maxErrorsCount={limit_value}

116	/nixSearch/inputParseReport?fileId={preloaded_file_id}[&maxErrorCount=<>]	GET	<No Body>	application/json - NIX Input Check Result text/csv - NIX Query Input Parse error report	NIX_TELE_SERVICE	200, 204, 404	Parses your input and provides parse result which describes result of parsing. Within the result you will be able to find which values contain parse errors and will not be recognized anyway. Input is taken from File preliminary uploaded to File Store and identified by parameter **fileId**. Answer depends on Accept header value. • If you are requesting application/json you'll be getting report as JSON with summaries on parse. Also you can limit here output with maxErrorsCount parameter. • If you are requesting text/csv - you'll be supplied with complete list of lines which we have failed to parse in TAB Separated value format. maxErrorsCount is not considered in this case. Following request URL parameters are supported: • maxErrorsCount={limit_value} • fileId=<id_of_file_on_storage>
117	/nixSearch/queryInfo?[params]	GET	<no body>	application/json, List of NIX Query Infos	NIX_TELE_SERVICE	200, 204	Provides extended information about NIX queries that user has issued. Allows extended filtering abilities comparing to separate resources for queries and results. Allows extracting information about several resources related to query in a single request. This could be useful when building some UI which provides information about query, result and settings at the same time. Result is ordered from the most recent to most eldest. Following request URL parameters are supported: • startIndex={index_value} • count={count_value} • queryName={qName} • queryPostDateFrom={dateFrom} • queryPostDateTo={dateTo} • queryStatus=[{NOT_STARTED|IN_PROGRESS|COMPLETED}, …] • queryId{id_of_query} - could be more than one.
118	/files[?temporary=<true/false>]	POST	multipart/form-data, with content as 'file' part	application/json, File Info	PROSP_PRIVATE or PROSP_COMPANY or GROUP_SEARCH or GROUP_ADMIN or NIX_TELE_SERVICE	200, 413	Uploads stream taken from the body of the multipart request to user files storage. In result you get information about your uploaded file which among other data holds ID of your file. This id can then be used in operations which refer to some input parameters as fileIds. You can also GET 413 error in case if your storage quota does not allow you to upload the file.
119	/files/info	GET	<no body>	application/json, List of File Info	PROSP_PRIVATE or PROSP_COMPANY or GROUP_SEARCH or GROUP_ADMIN or NIX_TELE_SERVICE	200, 204	Returns the list of objects describing files stored at user storage area.
120	/files/info/{id}	GET	<no body>	application/json, File Info	PROSP_PRIVATE or PROSP_COMPANY or GROUP_SEARCH or GROUP_ADMIN or NIX_TELE_SERVICE	200, 404	Returns information about a file referenced by its ID
121	/files/{id}	GET	<no body>	application/octetStream	PROSP_PRIVATE or PROSP_COMPANY or GROUP_SEARCH or GROUP_ADMIN or NIX_TELE_SERVICE	200, 204, 404	Downloads a file referenced by ID.
122	/files/{id}	DELETE	<no body>	<no body supplied>	PROSP_PRIVATE or PROSP_COMPANY or GROUP_SEARCH or GROUP_ADMIN or NIX_TELE_SERVICE	204,404	Deletes a file referenced by ID from user storage. Reports 404 if such file does not exist.
123	/dictionary/zips	GET	<no body>	application/json, List of Zip Info	PROSP_PRIVATE or PROSP_COMPANY or GROUP_SEARCH or GROUP_ADMIN or NIX_TELE_SERVICE	200, 204	Returns information about geographical areas matching given parameters. Pattern parameters support wildcards * (any number of any symbols) and ? (any symbol) Result is ordered alphabetically. Following request URL parameters are supported: • startIndex={index_value} • count={count_value} • zipCodePattern={pattern}
124	/dictionary/areaCodes	GET	<no body>	application/json, List of Strings	PROSP_PRIVATE or PROSP_COMPANY or GROUP_SEARCH or GROUP_ADMIN or NIX_TELE_SERVICE	200, 204	Returns the list of area codes matching given filtering criteria. Pattern parameters support wildcards * (any number of any symbols) and ? (any symbol) Result is ordered alphabetically. Following request URL parameters are supported: • startIndex={index_value} • count={count_value} • areaCodePattern={pattern} • topLevel - specify whether you would need to retrieve only fixed Area Codes (default = true) • withGeoReflection - specify whether you would need to retrieve only Area Codes with zip-ranges available (default = false)

125	/dictionary/zipCities	GET	<no body>	application/json, List of Strings	PROSP_PRIVATE or PROSP_COMPANY or GROUP_SEARCH or GROUP_ADMIN or NIX_TELE_SERVICE	200, 204	Returns the list of zip cities matching given filtering criteria. Pattern parameters support wildcards * (any number of any symbols) and ? (any symbol) Result is ordered alphabetically. Following request URL parameters are supported: • startIndex={index_value} • count={count_value} • namePattern={pattern}
126	/dictionary/municipality2zipCity	GET	<no body>	application/json, List of Zip City by Municipality Info	PROSP_PRIVATE or PROSP_COMPANY or GROUP_SEARCH or GROUP_ADMIN or NIX_TELE_SERVICE	200, 204	Returns the list of zip city objects grouped by municipality matching given filtering criteria. Pattern parameters support wildcards * (any number of any symbols) and ? (any symbol) Result is ordered alphabetically. Following request URL parameters are supported: • namePattern={pattern}
127	/dictionary/sniCodes?startIndex={index_value}&count={count_value}&sniCodesPattern={sni_pattern}	GET	<no body>	application/json, List of SNI Code Info	PROSP_PRIVATE or PROSP_COMPANY or GROUP_SEARCH or GROUP_ADMIN or NIX_TELE_SERVICE	200, 204	Returns set of SNI codes matching given pattern and other given criteria. Pattern can hold either part of SNI code itself or a part of its description. Ex. giving pattern like "Mining*" will return all the sni codes which have "Mining" word in their description thus relate to Mining industry. Pattern is case insensitive. Pattern parameters support wildcards * (any number of any symbols) and ? (any symbol) Result is ordered alphabetically. Following request URL parameters are supported: • startIndex={index_value} • count={count_value} • sniCodesPattern={pattern} Please note that returned collection of SNI Codes info holds description in the language specified by **Accept-Language** header. Supported values are: • EN - for getting and filtering in English (default) • SE - for getting and filtering in Swedish
128	/dictionary/counties	GET	<no body>	application/json, List of County Info	PROSP_PRIVATE or PROSP_COMPANY or GROUP_SEARCH or GROUP_ADMIN or NIX_TELE_SERVICE	200, 204	Returns information about counties matching given filtering criteria. Pattern parameters support wildcards *(any number of any symbols) and ? (any symbol) Result is ordered alphabetically. Following request URL parameters are supported: • countyNamePattern={pattern} • municipalityNamePattern={pattern}
129	/queryInfo?[params]	GET	<no body>	application/json, List of Query Info	GROUP_ADMIN	200, 204	Provides extended information about queries user has issued. Note, that this endpoint is available to Admin users only. Allows extended filtering abilities comparing to separate resources for queries and results. Allows extracting information about several resources related to a query in a single request. Result is ordered from the most recent to most eldest. Following request URL parameters are supported: • startIndex={index_value} • count={count_value} • queryName={qName} • queryPostDateFrom={dateFrom} • queryPostDateTo={dateTo} • queryStatus=[{NOT_STARTED|IN_PROGRESS|COMPLETED}, …] • queryType = [{PRIVATE|COMPANY|GROUP}, …] • resultHidden={yes/no} • customerName = {customerName}
130	/admin/privateProspects/results/{id}/locked	PUT	true/false	true/false	GROUP_ADMIN	404, 200	Allows admin users to lock/unlock particular results. Locked results cannot be extracted by users, neither by owner of result nor by admin.
131	/admin/companyProspects/results/{id}/locked						
132	/admin/groupSearch/results/{id}/locked						
133	/quickSearch?<parameters>	GET	<no_body>	application/json, List of Quick Search Results	SEARCH	200,204	Provides a list of entities matching criteria and selection parameters specified in URL. Basically here you define conditions of filtering and also selection parameters like bounds

Types

API operates with some specific structured data types. Down below you should find the description of data types we use. The description of methods above refers the types described here for convenience.

Version Info

The struct holds information on version of API currently up and running

Type Attributes

Name	Type	Constraints	Mandatory	Description
productFamily	String	"icatch3"	yes	Name of product family
productName	String	"icatch3-application"	yes	Name of product
majorVersion	Number		yes	Major version of API. Now it s "1"
minorVersion	String	Numbers split by dots	yes	Minor version of API. Now it is "3.1"
buildNumber	String		yes	Build number following internal build numbering format
versionComment	String		yes	Free text version comment

jSON Representation

```
{
 "productFamily" : "icatch3",
 "productName" : "icatch3-application",
 "buildNumber" : master-777,
 "majorVersion" : "1",
 "minorVersion" : "3.1",
 "versionComment" : "ic3 Search engine application"
}
```

Credentials

The value of type represents combination of user name and password.

Type Attributes

Name	Type	Constraints	Mandatory	Description
login	String		yes	Your user login.
password	String		yes	Your user password
ipAddress	String	IPv4 address	no	IP address of the client. If not specified then request's remote address information will be used.

JSON Representation

```
{
 "login": "myUserLogin",
 "password": "myUserPassword",
 "ipAddress": "173.194.116.161"
}
```

Authentication Result

Values of this type hold result of authentication. Within the result you can find auth token as well as some additional information about user got the token.

Type Attributes

Name	Type	Constraints	Mandatory	Description
authToken	String		yes	Value of auth token which should be used as the value of **X-Auth-Token** header in all the requests which require authorization.

userName	String		yes	Authenticated user login
permissions	String list		yes	Set of permissions associated to user. This could be useful to check whether you have permissions to access one or another iCatch3 feature.
costPlaceName	String		yes	Name of the cost place the user relates to
customerName	String		yes	Name of the customer the user relates to
token	String		no	JWT token. Value you expose to **Authorization** header in relevant requests

JSON Representation

```
{
 "authToken": "String",
 "userName": "String",
 "permissions" : [ "PERM_1", ... , "PERM_N" ],
 "costPlaceName": "String",
 "customerName": "String",
}
```

Query Settings

The type is responsible for holding information about Query Criteria and other settings which affect the query run. The settings could be provided as templates and as instances. Templates is what you as user of the system manage by yourself. You can modify, delete them as you need. Instances is something which system creates for you on query execution stage. Instances describe settings which were exactly used for issuing a query. Instances have special attribute which can allow you to bind them to template.

Private Prospects Settings

Type Attributes

Name	Type	Constraints	Mandatory	Description
id	String		yes	Setting ID. Usually this one is a String representation of UUID.
versionCode	String		yes (for PUT and GET)	ID of version. This ID is used by the system in order to determine conflicts (see Handling conflicts).
name	String	Max Length 200	yes (for templates)	Setting name. This attribute is present for templates only.
templateId	String		yes (for instances)	Setting template ID for settings instance. This attribute is only available for settings instances.
createdAt	Date	Read Only	yes (when getting)	Date of when the setting was initially created. You cannot modify the value of this field.
modifiedAt	Date	Read Only	yes (when getting)	Date of when the setting was last time modified. You cannot modify the value of this field.
ownerId	String	Read Only	yes (when getting)	ID of user who owning the settings.
Criteria				
maxAnswers	Integer	[1..100000]	yes	Defines maximum number of answers you expect to be returned.
phoneKind	Enum String	one of MOBILE, FIXED	no	Kind of phones you would like to get. If not specified it is considered that any kind of phones is acceptable.
phoneNumberInResult	Boolean	true/false	no	Determines whether to place phones to the result set. If false/null phone numbers will not be outputted. Default value is true.
minAge	Integer	>=16, <=maxAge	no	Natural integer defining minimal and maximal age of persons which info should appear in result set. Minimal age value should be less or equal to Maximum age value.
maxAge	Integer	<=99, >=minAge	no	
minBirthDate	Date	<= maxBirthDate	no	Date representing minimal and maximal birth dates of persons who should appear in result set. Minimal date should be less or equal to Maximal date.Also both dates should be more than 16 years ago in the past. In other case your request will be corrected automatically as we don't return people younger than 16 years old.
maxBirthDate	Date	>= minBirthDate	no	
sex	Enum String	MALE, FEMALE	no	Sex of persons you would like to get information about. If not specified it is considered that you can get persons of any sex.

householdOwnership	Enum String	DIRECTLY_OWNED , INDIRECT_OR_RENTAL	no	Kind of ownership of the household of persons who should appear in result. If not defined then data will not be filtered against that criteria. • DIRECTLY_OWNED House is on an address directly owned by a private person • INDIRECT_OR_RENTAL - House is on an address owned by a landlord or by a housing association (Swedish: "bostadsrättsförening")
limitPerHousehold	Boolean	true/false	no	If true then the result set will contain only a single person per each returned household. If false or not defined - there will be considered that we should not limit on that criteria.
nixAddressFilteringMode	Enum String	NIX_FREE, NIX_MARK	no	With this setting you can enable consideration of NIX database content. If not specified you will get no NIX DB related information, so phones from NIX database also can get to the result set. Value could be one of following: • NIX_FREE - Only the records of people who's addresses are not contained within NIX DB will be returned • NIX_MARK - Any records about persons will be returned, but those who is present in the nix DB will be specially marked.
nixPhonesFilteringMode	Enum String	NIX_FREE, NIX_MARK	no	With this setting you can enable consideration of NIX database content. If not specified you will get no NIX DB related information. Value could be one of following: • NIX_FREE - Only the records of people who's phone numbers are not contained within NIX DB will be returned • NIX_MARK - Any records about persons will be returned, but those who is present in the NIX DB will be specially marked.
duplicatesHandlingMode	Enum String	ALLOW_PHONE, ALLOW_SSN	no	If specified you will get duplicates record within the output set according to selected mode. If not specified returned records will all be unique by SSNs and Phone numbers. The value can be one of following: • ALLOW_PHONE - allow including duplicates in case they have different phone numbers • ALLOW_SSN - allow including duplicates in case they have different SSN's
excludeBy	Enum String	SETTING, USER, COST_PLACE, CUSTOMER	no	If specified then results of query will be filtered according to specified mode. If you have specified this parameter then it is mandatory to define **excludePeriod**. There could be specified one of following values: • SETTING - Results returned in scope of recent **excludePeriod** months produced by queries using the setting will not be returned • USER - Results returned in scope of recent **excludePeriod** months produced for the user owning the setting will not be returned • COST_PLACE - Results returned in scope of recent **excludePeriod** months produced for other users of the same cost place as the user owning the setting will not be returned • CUSTOMER - Results returned in scope of recent **excludePeriod** months produced for other users which relates to the same customer as the user owning the setting will not be returned.
excludePeriod	Integer	[1..18]	yes with **exclude** option	Number of months which affect the scope defined for **exclude**. If specified without **excludeBy** then it is just ignored.
geoFilter	Geo Filter	mutually exclusive with **addressFilter**	no	The set of geographic related filtering attributes. All the items of filter are used in disjunction to each other.
exclusions	Private Persons Exclusions	Content-Type: text/csv	no	This parameters should be provided as a multipart request. The part should be named as **file** if this is the part of direct request with no other parts. If it is supplied together with address filter the name of part should correspond the name defined in first column of current table. Exclusions could hold Phone numbers, SSNs and Dwelling Ids. Last ones could be used only by Itesco staff as these are internal IDs which are not available through the public API.
excludeSameHousehold	Boolean	true/false	no	This parameter is related to "exclusions" parameter. If excludeSameHousehold is set to true, then people, who share dwelling ID with owners of SSNs and phones provided in "exclusions", will be excluded. Default value is 'false'.
addressFilter	Address Criteria List	Content-Type: text/csv	no	This parameters should be provided as a multipart request. Part should be named as **file** if this is a part of direct request with no other parts. If it is supplied together with exclusions the name of this part should correspond the name defined in first column of current table.
exclusionsFilterItemsCount	Integer	Read only	yes (when getting)	This attribute appears only when you are getting information about settings. It serves to indicate you that there are some Private Persons Exclusions defined in addition to the returned settings. In order to get excusions themselves you should use separate resource.
addressFilterItemsCount	Integer	Read only	yes (when getting)	This attribute appears only when you are getting information about settings. It serves to indicate you that there are some Address Criteria List defined in addition to the returned settings. In order to get address filter content itself you should use separate resource.

JSON Representation

```
{
 "id": "String",
 "versionCode" : "String",
 "name": "String",
    "createdAt" : <Date_in_format ISO-8601>,
    "modifiedAt" : <Date_in_format ISO-8601>,
 "maxAnswers": <Number_in_range: [1..1000]>,
    "ownerId: "String"
    "phoneKind" : "<String_one_of: FIXED, MOBILE>",
 "minAge": <Number_in_range: [16..99]>,
 "maxAge": <Number_in_range: [16..99]>,
 "minBirthDate": <Date_in_format ISO-8601>,
 "maxBirthDate": <Date_in_format ISO-8601>,
 "sex": "<String_one_of: MALE, FEMALE>",
 "householdOwnership": "<String_one_of: DIRECTLY_OWNED, INDIRECT_OR_RENTAL>",
 "limitPerHousehold": Boolean,
 "nixAddressFilteringMode": "<String_one_of: NIX_FREE, NIX_MARK>",
 "nixPhonesFilteringMode": "<String_one_of: NIX_FREE, NIX_MARK>",
 "duplicatesHandlingMode": "<String_one_of: ALLOW_SSN, ALLOW_PHONE>",
 "excludeBy": "<String_one_of: SETTING, USER,COST_PLACE, CUSTOMER>",
 "excludePeriod": Number,
 "excludeSameHousehold": Boolean
 "geoFilter" : <GeoFilter>,
 "exclusionsFilterItemsCount": <Number>
 "addressFilterItemsCount": <Number>
 }
```

Please note, that when you are getting information about settings you do not receive exclusion and allowedAddresses directly in Json body. You should use separate resources in order to access them.

Example of Settings for POST

```
{
 "name": "myNewName",
 "maxAnswers": 200,
 "phoneKind" : "MOBILE",
 "maxAge": 60,
 "minBirthDate": "1984-01-01",
 "householdOwnership": "DIRECTLY_OWNED",
 "limitPerHousehold": "true",
 "nixPhonesFilteringMode": "NIX_FREE",
 "excludeBy": "SETTING",
 "excludePeriod": 6
 }
```

Example of Settings for PUT

```
{
 "id": "064ed556-c9fd-102b-9d1c-0030487dabd6",
 "name": "myNewName",
 "versionCode": "064ed556-c9fd-102b-9d1c-0030437dabd6",
 "maxAnswers": 200,
 "phoneKind" : "FIXED",
 "maxAge": 30,
 "limitPerHousehold": "true",
 "nixPhonesFilteringMode": "NIX_FREE",
 "excludeBy": "USER",
 "excludePeriod": 3
}
```

Example of Settings for GET

```
{
 "id": "064ed556-c9fd-102b-9d1c-0030487dabd6",
 "name": "myNewName",
 "versionCode": "063ed556-c9fd-102b-9d1c-0030437dabd6",
    "ownerId": "064ed556-c9fd-103b-9d1c-0030437dabd6",
    "createdAt" : "2015-02-215T16:30:44Z",
    "modifiedAt" : "2015-03-26T07:28:27Z",
"maxAnswers": 200,
"phoneKind" : "MOBILE",
"maxAge": 60,
"minBirthDate": "1984-01-01Z",
"householdOwnership": "DIRECTLY_OWNED",
"limitPerHousehold": "true",
"nixPhonesFilteringMode": "NIX_FREE",
"excludeBy": "SETTING",
"excludePeriod": 6,
"exclusionsFilterItemsCount": 10,
"addressFilterItemsCount": 12
}
```

Company Prospects Settings

Type Attributes

Name	Type	Constraints	Mandatory	Description
id	String		yes	Setting ID. Usually this one is a String representation of UUID.
versionCode	String		yes (for PUT and GET)	ID of version. This ID is used by the system in order to determine conflicts (see Handling conflicts).
name	String	Max Length 200	yes (for templates)	Setting name. This attribute is present for templates only.
templateId	String		yes (for instances)	Setting template ID for settings instance. This attribute is only available for settings instances.
createdAt	Date	Read Only	yes (when getting)	Date of when the setting was initially created. You cannot modify the value of this field.

modifiedAt	Date	Read Only	yes (when getting)	Date of when the setting was last time modified. You cannot modify the value of this field.
ownerId	String	Read Only	yes (when getting)	ID of user who owns the settings.
Criteria				
maxAnswers	Integer	[1..100 000]	yes	Defines maximum number of answers you expect to be returned.
phoneKind	Enum String	one of • MOBILE, • FIXED	no	Kind of a phone that expected companies should have. If nothing is chosen, we don't filter on phone kind. Companies with any kind of phone are returned by default. This would include companies which have at least one phone number associated.
onlyWithPhones	Boolean	true/false, Default = false	no	If true then companies which don't have phone number assigned will not be returned.
assignAlternative	Boolean	true/false, Default = true	no	If true we will try to find alternative phone matching defined phoneKind in case if official is missing.
allowAlternativePrivate	Boolean	true/false, Default = false	no	Allow assigning private nix-free numbers as an alternative phone if official phone is missing. This settings is only considered when assignAlternative is set to true.
maxExtraPhoneNumbers	Integer	[0..1 000 000]	no	Defines maximum number of extra phone numbers of a company in result. If not defined there will be no extra phone numbers returned (default).
extraPhoneKind	Enum String	one of • MOBILE • FIXED	no	A kind for extra phones of a company to return. If maxExtraPhoneNumbers is not defined this parameter value is ignored. Default value is null. This means that any kind of additional phone is going to be returned.
matchBranches	List of Enum String	Items could be one of • BY_SNI, • BY_GEO	no	Defines whether SNI-codes and geography matching should be performed against branches. If empty branches will not be filtered at all. There will be returned as much rows about branches as it is allowed by maxBranchPhoneNumbers parameters.
maxBranchPhoneNumbers	Integer	[0..100000]	no	Defiles maximum number of branches with their phones returned in result set. If not defined then no branch phones is going to be returned.
branchPhoneKind	Enum String	one of • MOBILE, • FIXED	no	Defines which kind of branch phones you prefer to see in results. If this is not defined then any branch phone kind will be considered.
sniCodesFilter	SNI Codes Filter	SNI codes filtering conditions	no	Defines conditions on filtering by SNI codes.
minEmployeesCount	Integer	[0..1 000 000], <= maxEmployeesCount if defined	no	Defines minimum and maximum number of employees that expected companies should have. Minimum number of employees should be less that or equal to maximum number. No filtering by default.
maxEmployeesCount		[0..1 000 000], >= maxEmployeesCount if defined	no	
minTurnover	Integer	in 1000's of SEK, <= maxTurnover if defined	no	Defines minimum and maximum turnover value that expected companies should have. If at least one of the values is specified, no companies with unknown turnover returned. Minimum value should be less than or equal to maximum one. No filtering by default.
maxTurnover		in 1000's of SEK, >= minTurnover if defined	no	
propertyForms	List of Enum String	Items could be one of: • LIMITED, • PARTNERSHIP, • SINGLE_PERSON_COMPANY, • ECONOMIC_UNION, • NON_PROFIT, • FOUNDATION, • ASSOCIATION, • PUBLIC_ADMINISTRATION	no	Defines property form for expected companies. Companies with unknown form of property is not returned, if at least one property form is specified. No filtering by default. This correspond to empty list.
minRegistrationDate	Date	<= maxRegistrationDate if defined	no	Defines minimum and maximum registration date expected companies should have. Minimum date should be less than or equal to maximum date.
maxRegistrationDate	Date	>= minRegistrationDate if defined	no	
excludeBy	Enum String	one of: • SETTING • USER • COST_PLACE • CUSTOMER	no	If specified then results of query will be filtered accordingly to the specified mode. If you have specified this parameter then it is mandatory to define **excludePeriod**. There could be specified one of following values: • SETTING - Results returned in scope of recent **excludePeriod** months produced by queries using the setting will not be returned • USER - Results returned in scope of recent **excludePeriod** months produced for the user owning the setting will not be returned • COST_PLACE - Results returned in scope of recent **excludePeriod** months produced for other users of the same cost place as the user owning the setting will not be returned • CUSTOMER - Results returned in scope of recent **excludePeriod** months produced for other users which relates to the same customer as the user owning the setting will not be returned. No exclusion is done by default.
excludePeriod	Integer	[1..18]	yes, if **excludeBy** is specified	Number of months which affect the scope defined for **excludeBy**. If specified without **excludeBy** then it is just ignored.
excludeConsider	List of values	values could be one of: • ORG_NUMBER • PHONE_NUMBER	yes, if **excludeBy** is specified	Define which property of an answer is considered: Organization numbers, phone numbers or both. If more than one value is given they are considered by OR. By default either ORG_NUMBER or PHONE_NUMBER is considered when excluded.
exclusions	Company Exclusions	Content-Type: text/csv	no	This parameters should be provided as a multipart request. The part should be named as **file** if this is a part of direct request with no other parts. If it is supplied together with address filter the name of the part should correspond to the name defined in the first column of current table. Exclusions could hold phone and organization numbers.
geoFilter	Geo Filter	mutually exclusive with **addressFilter**	no	The set of geographic related filtering attributes. All the items of filter are used in disjunction to each other.

addressFilter	Address Criteria List	Content-Type: text/csv	no	This parameters should be provided as a multipart request. The part should be named as **file** if this is a part of direct request with no other parts. If it is supplied together with exclusions the name of the part should correspond to the name defined in the first column of current table.
minBranchCount	Integer	[0..1 000 000], <= maxBranchCount if defined	no	Defines minimum and maximum number of branches of a company that should be presented in result set. For companies which do not have branches we consider number of branches equal to 0.
maxBranchCount	Integer	[0..1 000 000], >= minBranchCount if defined	no	
exclusionsFilterItemsCount	Integer	Read only	yes (when getting)	This attribute appears only when you are getting information about settings. It serves to indicate you that there are some Company Exclusions defined in addition to the returned settings. In order to get exclusions themselves you should use separate resource.
addressFilterItemsCount	Integer	Read only	yes (when getting)	This attribute appears only when you are getting information about settings. It serves to indicate you that there are some Address Criteria List in defined in addition to the returned settings. In order to get address filter content itself you should use separate resource.

JSON Representation

```
{
 "id": "String",
 "versionCode" : "String",
    "name": "String",
    "createdAt" : <Date_in_format ISO-8601>,
    "modifiedAt" : <Date_in_format ISO-8601>,
    "maxAnswers": <Number_in_range: [1..1000]>,
    "ownerId: "String",
    "phoneKind" : "<String_one_of: FIXED, MOBILE, ANY>",
 "assignAlternative": Boolean,
 "allowPrivateAlternative": Boolean,
 "maxExtraPhoneNumbers" :  <Number_in_range: [0..]>,
    "onlyWithPhones": Boolean,
 "extraPhoneKind" : "<String_one_of: FIXED, MOBILE, ANY>",
    "matchBranches": [<String_one_of: "BY_SNI", "BY_GEO",...],
 "sniCodesFilter": <SniCodesFilter>,
 "minRegistrationDate" : <Date_in_format ISO-8601>,
    "maxRegistrationDate" : <Date_in_format ISO-8601>,
    "minEmployeesCount" : Number,
 "maxEmployeesCount" : Number,
    "minTurnover" : Number,
 "maxTurnover" : Number,
 "propertyForms" : [<String_one_of: "LIMITED", "PARTNERSHIP", "SINGLE_PERSON", "ECONOMIC_UNION", "NON_PROFIT", "FOUNDATION", "ASSOCIATION",
"PUBLIC_ADMINISTRATION">,...],
 "excludeConsider" : ["<String one of: ORG_NUMBER, PHONE_NUMBER>",...],
 "excludeBy": "<String_one_of: SETTING, USER,COST_PLACE, CUSTOMER>",
    "excludePeriod": Number,
    "geoFilter" : <GeoFilter>,
    "exclusionsFilterItemsCount": <Number>
    "addressFilterItemsCount": <Number>
}
```

Please note, that when you getting information about settings you will not get exclusion and allowedAddresses directly in Json body. You should be using separate resources in order to access them

Example of Settings for POST

```
{
 "name": "myNewName",
 "maxAnswers": 200,
 "phoneKind" : "MOBILE",
 "minRegistrationDate": "1990-01-01",
 "householdOwnership": "DIRECTLY_OWNED",
 "sniCodesFilter":{
  "includeSniCodes" : ["101", "16232", "212"]
 },
 "excludeBy": "SETTING",
 "excludeConsider" : ["ORG_NUMBER"],
 "excludePeriod": 6
 }
```

Example of Settings for PUT

```
{
 "id": "064ed556-c9fd-102b-9d1c-0030487dabd6",
 "name": "myNewName",
 "versionCode": "064ed556-c9fd-102b-9d1c-0030437dabd6",
 "maxAnswers": 200,
 "phoneKind" : "MOBILE",
 "minRegistrationDate": "1990-01-01",
 "sniCodesFilter":{
  "excludeSniCodes" : ["101", "16232", "212"]
 },
 "propertyForms": ["LIMITED", "ASSOCIATION"],
 "excludeConsider" : ["ORG_NUMBER, PHONE_NUMBER"],
 "excludeBy": "USER",
 "excludePeriod": 3
}
```

Example of Settings for GET

```
{
 "id": "064ed556-c9fd-102b-9d1c-0030487dabd6",
 "name": "myNewName",
 "versionCode": "063ed556-c9fd-102b-9d1c-0030437dabd6",
    "ownerId": "064ed556-c9fd-103b-9d1c-0030437dabd6",
    "createdAt" : "2015-02-215T16:30:44Z",
    "modifiedAt" : "2015-03-26T07:28:27Z",
 "maxAnswers": 200,
 "phoneKind" : "MOBILE",
 "minRegistrationDate": "1990-01-01",
 "sniCodesFilter":{
  "includeSniCodes" : ["101", "16232", "212"]
 },
 "propertyForms": ["LIMITED", "PARTNERSHIP"]
 "excludeBy": "USER",
 "excludeConsider" : ["PHONE_NUMBER"],
 "excludePeriod": 3
 "exclusionsFilterItemsCount": 10,
 "addressFilterItemsCount": 12
 }
```

Group Search Settings

Type Attributes

Name	Type	Constraints	Mandatory	Description
id	String		yes	Setting ID. Usually this one is a String representation of UUID.
versionCode	String		yes (for PUT and GET)	ID of version. This ID is used by the system in order to determine conflicts (see Handling conflicts).
name	String	Max Length 200	yes (for templates)	Setting name. This attribute is present for templates only.
templateId	String		yes (for instances)	Setting template ID for settings instance. This attribute is only available for settings instances.
queryInput	Group Query Input		yes (for instances)	Information about input user has supplied while posting the query. This attribute is only available for settings instance.
createdAt	Date	Read Only	yes (when getting)	Date of when the setting was initially created. You cannot modify the value of this field.
modifiedAt	Date	Read Only	yes (when getting)	Date of when the setting was last time modified. You cannot modify the value of this field.
ownerId	String	Read Only	yes (when getting)	ID of user who the settings belongs to.
Criteria				
inAddressWash	Boolean	true/false	no	Setting this parameter to true enables so called "Address wash" algorithm which does its best to correct possible typical errors which we can encounter while processing your input. It is **ON** by default. Disabling this could have sense if you are confident that given addresses are specified correctly and would like minimize any assumptions made by the system while matching.
inFlipNameOrder	Boolean	true/false	no	Setting parameter to true enables Reversing of order of name and surname if they are given in a single column. By default system treat the first word as a name and the last word as a surname. If your input breaks this rule - you can use this parameter. So if parameter is true then we will consider that a surname comes first. By default parameter is **OFF**
features	List of Feature IDs		no	This is a collection of special Features which you would like to apply to your input/output. Itesco provides service for supplying a client specific features which allow to do some additional actions on input you supply or output you get or both. Our development team could develop any custom logic you require if e.g. your input format doesn't match directly supported by service or if you e.g.. would like to get more information than we supply usually. Please contact Itesco sales stuff to request any special features you need.
matchingTemplates	List of Matching Template IDs		no	Within the field you can define a set of matching templates that we need to apply while performing group search. If you do not define this field we consider automatic detection of set of templates according to your input. If you have a special preference, you can restrict set of matching templates we apply defining the value of the parameter.
outAnswersPerQuestion	Integer	Natural, > 0 <=1000	yes	Defines maximal number of answers you expect system will return within output for each of your questions.

outPhoneFormat	Enum String	one of NATIONAL, NATIONAL_NOSPLIT, INTERNATIONAL	no	Defines format of phones in output you will get. We support following values: • NATIONAL - AAA-XXXXX - Default one. Considered when not defined • NATIONAL_NOSPLIT - 0AAAXXXXX • INTERNATIONAL - 467AAAXXXXX Default value is 'NATIONAL'
searchAmong	List of Enum values	One of: COMPANIES, PERSONS	no	Defines among which kind of data we are going to look for matches. At the moment there are following values supported: • COMPANIES - Lookup among companies information we have • PERSONS - Lookup among private persons information we have If not defined, we are going to look for matches among **ALL KINDS OF DATA WE HAVE**
outPhoneKind	Enum String	One of MOBILE, FIXED	no	Defines whether we should look up only against MOBILE or FIXED phone or any. By default if not specified, we consider that any type of subscription is acceptable within your expected output.
outPhoneKindPrefer	Enum String	One of MOBILE, FIXED	no	Defines which kind of phone you prefer to be found. So if we have more than one phone matching the given criteria we will use this parameter value to determine which one to return if total number of found phones > *outAnswersPerQuestion* value. Makes sense only if *outPhoneKind* parameter is not defined.
outNixAddress	Enum String	One of NIX_MARK, NIX_FREE	no	Controls whether we should compare returned addresses against the NIX registry and act according to a value specified: • NIX_FREE - exclude address if it is found within NIX-Addresses registry • NIX_MARK - return address, but mark it as NIXED one within the result By default, if the value is not specified, we don't do any nix-matching on output rows
outNixPhone	Enum String	One of NIX_MARK, NIX_FREE	no	Controls whether we should compare returned phones against the NIX registry and act according to a value specified: • NIX_FREE - exclude phone if it is found within NIX-Phones registry • NIX_MARK - return phone , but mark it as NIXED one within the result By default, if the value is not specified, we don't do any nix-matching on output rows
matchAddressType	List of enum Enum String	One of 'OFFICIAL', 'OPERATOR'	no	Controls whether we should perform matching of addresses based either on an official address or on operator data. Actual for Company selection only. • OFFICIAL - use official addresses • OPERATOR - uses operator data By default, the value is not specified. This means that we will lookup on both.
outEmptyWhenNotFound	Boolean	true/false	no	It is used in case when you'd like to have at least one output row returned for each of the questions you have specified (they will be charged). See Group Query Result Value paragraph to know result format for not answered questions are returned. For this you need to set this parameter to true. Otherwise output will not hold questions rows to which we did not manage to find any of answers. By default it is **ON.**
outUpdateModes	List of enum Enum String	one of "address", "phone", "name"	no	When this list is not empty and an input row contains either Phone number or Organization Number/SSN we return the most actual information on this phone/SSN/orgNum we have in database accordingly to specified update mode. The result "updated" column will have one or more values depending on update modes requested and used for a particular question: address, phone, name. It will be empty if this wasn't updated and outUpdateModes criteria is set. If there are more than one value they are split by semicolon ";". Updated rows are charged additionally. The "updated" column is empty if this criteria is not set. By default option is **OFF.**

JSON Representation

```
{
 "id": "String",
 "versionCode" : "String",
    "name": "String",
    "createdAt" : <Date_in_format ISO-8601>,
    "modifiedAt" : <Date_in_format ISO-8601>,
    "maxAnswers": <Number_in_range: [1..1000]>,
    "ownerId: "String",
    "inAddressWash" : "true/false",
 "inFlipNameOrder": "true/false",
 "features": ["<String_one_of_valid_feature_ids">,...],
 "matchingTemplates": ["<String_one_of_valid_matching_template_ids">,...],
 "outAnswersPerQuestion" :  <Number_in_range: [0..]>,
 "searchAmong" : ["<String_one_of: COMPANIES, PERSONS>",...],
 "matchAddressType" : ["String one of OFFICIAL, OPEARTOR", ...],
    "outPhoneKind" : "<String_one_of: FIXED, MOBILE>",
 "outPhoneKindPrefer" : "<String_one_of: FIXED, MOBILE>",
 "outNixAddress" : "<String_one_of: NIX_FREE, NIX_MARK>",
 "outNixPhone" : "<String_one_of: NIX_FREE, NIX_MARK>",
 "outUpdateModes": ["<String_one_of: COMPANIES, PERSONS>",...]
 }
```

Example of Settings for POST

```
{
 "name": "myNewName",
 "outAnswersPerQuestion": 200,
 "outPhoneKindPrefer" : "MOBILE",
 "searchAmong": ["PERSONS"],
 "matchingTemplates":["1100", "1110", "1120", "1130", "1200", "1210"]
 }
```

Example of Settings for PUT

```
{
 "id": "064ed556-c9fd-102b-9d1c-0030487dabd6",
 "versionCode": "064ed556-c9fd-102b-9d1c-0030437dabd6",
 "name": "myNewName",
 "outAnswersPerQuestion": 200,
 "outPhoneKindPrefer" : "MOBILE",
 "searchAmong": ["PERSONS"],
 "matchingTemplates":["1100", "1110", "1120", "1130", "1200", "1210"]
 }
```

NIX Query Settings

Type Attributes

Name	Type	Constraints	Mandatory	Description
id	String		yes	Setting ID. Usually this one is a String representation of UUID.
versionCode	String		yes (for PUT and GET)	ID of version. This ID is used by the system in order to determine conflicts (see handling conflicts).
name	String	Max Length 200	yes	Setting name. This attribute is present for templates only.
queryInput	NIX Query Input		yes	Information about input user has supplied while posting the query.
createdAt	Date	Read Only	yes (when getting)	Date of when the setting was initially created. You cannot modify the value of this field.
modifiedAt	Date	Read Only	yes (when getting)	Date of when the setting was last time modified. You cannot modify the value of this field.
ownerId	String	Read Only	yes (when getting)	ID of user who the settings belongs to.

Nix Query Input

This structure describes query input which was used for issuing NIX Phones query.

Type Attributes

Name	Type	Constraints	Mandatory	Description
id	String		yes	Setting ID. Usually this one is a String representation of UUID.
versionCode	String		yes (for PUT and GET)	ID of version. This ID is used by the system in order to determine conflicts (see Handling conflicts).
instanceId	String	ID	yes	ID of settings instance to which input relates to.
name	String	Max Length 200	yes (for templates)	Setting name. This attribute is present for templates only.
createdAt	Date	Read Only	yes (when getting)	Date of when the setting was initially created. You cannot modify the value of this field.
modifiedAt	Date	Read Only	yes (when getting)	Date of when the setting was last time modified. You cannot modify the value of this field.
ownerId	String	Read Only	yes (when getting)	ID of user who the settings belongs to.
totalInputItems	Integer	positive	yes	Total count of non empty input items detected within the input
totalParsed	Integer	<= totalInputItems	yes	Total count of valid input items. These are going to be processed by a query engine.
totalErrors	Integer	<= totalInputItems	yes	Total count of items we did not managed to parse, likely containing input errors
totalDuplicates	Integer	<= totalInputItems	yes	Total count of duplicates detected within the user input

Group Query Input

This structure describes input which user has posted when executing a group query. Input is a part of Group Settings Instance, but as it could be quite huge we are not supplying it when settings are retrieved (separate endpoint serves for getting an input). Instead there is provided an extract of information of input being supplied when you are retrieving settings instance. For information on exact format of input please refer Group Questions section.

Type Attributes

Name	Type	Constraints	Mandatory	Description
id	String		yes	Setting ID. Usually this one is a String representation of UUID.
versionCode	String		yes (for PUT and GET)	ID of version. This ID is used by the system in order to determine conflicts (see Handling conflicts).
instanceId	String	ID	yes	ID of settings instance to which input relates to.
name	String		no	Name of a file supplied when posting information on input. Could be null if there was not supplied any.
createdAt	Date	Read Only	yes (when getting)	Date of when the setting was initially created. You cannot modify the value of this field.
modifiedAt	Date	Read Only	yes (when getting)	Date of when the setting was last time modified. You cannot modify the value of this field.
ownerId	String	Read Only	yes (when getting)	ID of user who the settings belongs to.
totalInputItems	Inetger		yes	Total number of questions detected within supplied input
totalParsed	Integer	<= totalInputItems	yes	Number of questions which system managed to recognize. These are going to be processed by a query engine.
totalErrors	Integer	<= totalInputItems	yes	Number of items with errors detected within your input.

JSON Representation

```
{
 "instanceId": "String",
 "name" : "String",
 "totalInputItems" :  <Number_in_range: [0..]>,
    "totalParsed" :  <Number_in_range: [0..]>,
    "totalErrors" :  <Number_in_range: [0..]>
}
```

SNI Codes Filter

This type holds definition of filtering by SNI code property of company. Basically you can define a set of include/exclude values. SNI Codes Filter is a part of Company Prospects Settings. Please note, that exclude setting takes precedence over include one. In case if the same SNI code is defined within includes and excludes it will be considered as excluded one.

Type Attributes

Name	Type	Constraints	Mandatory	Description
includeSniCodes	String list	2-5 digits SNI-code values (DD, DDD, DDDD, DDDDD)	no	Defines list of SNI codes for companies expected in result.
excludeSniCodes	String list	2-5 digits SNI-code values (DD, DDD, DDDD, DDDDD)	no	Defines list of SNI codes for companies not expected in result.
considerSecondarySNI	Boolean	true/false. Default is false	no	Defines whether we should consider secondary SNI codes when filtering on SNI codes property.

JSON Representation

```
{
 "includeSniCodes" : ["sni_1", "sni_2", ...],
 "excludeSniCodes" : ["sni_1", "sni_2", ...],
 "considerSecondary": "true/false"
}
```

For samples of usage in requests please refer Company Prospects Setting.

Geo Filter

This type holds definition of filtering by different kinds of geo properties. Geo Filter is part of Private Prospects Settings and Company Prospects Settings.

Type Attributes

Name	Type	Constraints	Mandatory	Description
zipCodes	String list	zip code values (DDDDD, DDDDD-DDDDD),	yes(when getting)	If specified the results will be limited to defined set of zip codes. Each item of the list represents zip code or range of zip codes. Zip code is a numeric string consisting of 5 numeric characters, e.g. 82691. ZIP range separator is hyphen, e.g. 10120-10130.
zipCities	String list	One of supported zipCities captions	yes(when getting)	If specified the results will be limited to defined set of zip Cities. Each item in the list represents a zip City name. In order to get the list of counties you can refer specific method of the API.
areaCodes	String list	Items: area code values (DDD)	yes(when getting)	If specified the results will be limited to defined set of area codes. Each item of the list should represent the area code as string containing 3 digits.
counties	String list	One of supported counties captions	yes(when getting)	If specified the results will be limited to defined set of counties. Each item in the list represents a county name. In order to get the list of counties you can refer specific method of the API.
municipalities	String list	One of supported municipalities captions	yes(when getting)	If specified the results will be limited to defined set of municipalities. Each item in the list represents a municipality name. In order to get the list of municipalities you can refer specific method of the API

JSON Representation

```
{
 "zipCodes": ["Zip_code_1", "Zip_code_2", ...],
 "zipCities": ["Zip_City_name_1", "Zip_City_name_2", ...],
 "areaCodes": ["Area_code_1", "Area_code_2", ...],
 "counties": ["County_name_1", "County_name_2", ...],
 "municipalities": ["Municipality_name_1", "Municipality_name_2", ...]
}
```

For samples of usage in requests please refer Private Prospects Settings.

Query

Query is the definition of task for Private persons/Companies/Group Search selection machinery. Basically it consists of metadata, reference to settings should be used when doing selections. It could be used to track progress of query being executed. Also Id of the query is used as key for getting results when query has finished. This object is read only as it reflects the state of execution of your query tasks on server

Type Attributes

Name	Type	Constraints	Mandatory	Description
id	String		yes	Query id. Usually this one is a String representation of UUID.
name	String	Max Length 200	yes	Query name.
versionCode	String		yes	ID of version. This ID is used by the system in order to determine conflicts (see Handling conflicts).
createdAt	Date		yes	Date of when the query was initially created. You cannot modify the value of this field.
modifiedAt	Date		yes	Date of when the setting was last time modified. You cannot modify the value of this field.
settingsInstanceId	String		yes	Settings instance id. Usually this one is a String representation of UUID.
status	Enum String	One of NOT_STARTED, IN_PROGRESS, COMPLETED	yes	Status of a query. Can be used for tracking the progress of query execution.
type	Enum String	One of PRIVATE, COMPANY, GROUP, NIX_PHONE	yes	Type of query being submitted.
startedAt	Date		yes	Date and time when execution of a query has started. It is undefined if status of your query is NOT_STARTED.
completedAt	Date		yes	Date and time when execution of a query has finished. It is undefined if status of your query status is not COMPLETED.
progress	Integer	[0..100]	yes	Holds value intended to help with tracking the progress. 0 considered to be not started queries 100 considered to be completed queries.

JSON Representation

```
{
 "id": "String",
 "name": "String",
 "versionCode": "String",
 "createdAt" : <Date_in_format ISO-8601>,
    "modifiedAt" : <Date_in_format ISO-8601>,
 "settingsInstanceId" : "String",
    "status": "<String_one_of: NOT_STARTED, IN_PROGRESS, COMPLETED>",
    "type": "<String_one_of: PRIVATE, COMPANY, GROUP>",
 "startedAt": "<Date_in_format ISO-8601>",
    "completedAt": "<Date_in_format ISO-8601>",
    "progress": Integer
}
```

Example of Query for GET

```
{
 "id": "064ed556-c9fd-102b-9d1c-0030487dabd6",
 "name": "myNewName",
 "type": "PRIVATE",
 "settingsId": "064edf38-c9fd-102b-9d1c-0030487dabd6",
 "versionCode": "064edf38-c9fd-102b-9d2c-0030487dabd6",
 "status": { "status": "IN_PROGRESS", "startedAt": "2015-03-26T07:26:17Z", progress: 85 },
  "createdAt" : "2015-02-215T16:30:44Z",
    "modifiedAt" : "2015-03-26T07:28:27Z"
}
```

Query Definition

This object is used to define a query task for execution. It is used for performing queries and making dry run. Basically it holds a reference to settings which should be used and criteria overrides. Under overrides we mean a set of fields in a form of Private Prospects Setting /Company Prospects Settings / Group Search Settings / NIX Query Settings object which relate to selection **Criteria**. The values specified within the Query object have precedence over similar values defined in bound settings instance. This object is used only for submitting a query for execution. You will not be able to get it after submitting. Instead you will be able to get Private Prospects Settings / Company Prospects Settings / Group Search Settings / NIX Query Settings object which was build from referenced settings and overrides you have supplied as well as query metadata itself as Query object.

Note that prospects settings and criteria overrides types should be consistent. Both should be either private prospects settings or company prospects settings or group search settings. None of them can override another one. The same relates to exclusions.

Type Attributes

Name	Type	Constraints	Mandatory	Description
settingsTemplateId	String	UUID	yes (for queries supporting templates)	ID of settings you own which should be used as basic settings for your search. Usually this one is a String representation of UUID.
name	String	200 chars max	yes (for issuing query)	Name of a query which you would like to use for its identification. This could be any string matching constraint. For dry run name of a query is not required.
criteriaOverrides	Private Prospects Settings / Company Prospects Settings / Group Search Settings / NIX Query Settings	Sub-set of properties defining query parameters	no	Set of settings which should be used instead of ones defined in base settings. Basically this is sub-set of criteria from Private Prospects Settings / Company Prospects Settings / Group Search Settings / NIX Query Settings. All fields have the same meaning and constraints.
exclusions	Private Persons Exclusions / Company Exclusions	Content-Type: text/csv (supplied as the part of multipart request)	no	With this attribute you could define exclusions set which would be only current query specific. Note that the exclusions you have supplied will be merged to base settings exclusions, so the query will use both lists. Specific for Private Prospects Settings / Company Prospects Settings.
addressFilter	Address Criteria List	Content-Type: text/csv (supplied as the part of multipart request)	no	With this attribute you could define address filter set which would be only current query specific. Specific for Private Prospects Settings / Company Prospects Settings.

queryInput	Text input stream	Content-Type: text/* (supplied as the part of multipart request)	yes (for Group searches and Nix queries)	The part under this name should hold a query input which is going to be processed by queries which accept user input (like group searches or nix queries).

Example of Settings for POST

```
{
 "name": "myNewName",

 "settingsId": "064edf38-c9fd-102b-9d1c-0030487dabd6"
 "criteriaOverrides": {
  "maxAge": 30,
  "excludeBy": "USER",
  "excludePeriod": 3
 }
}
```

NIX Query Definition

This object is used to define nix search task for execution. It is used for performing searches. Basically it only holds the reference to an input used to perform searches. The values specified within the Query object have precedence over similar values defined in bound settings instance. This object is used only for submitting the query for execution. You will not be able to get it after submitting. Instead you will be able to get NIX Query Settings object which was build from referenced settings and overrides you have supplied as well as query metadata itself as Query object.

Type Attributes

Name	Type	Constraints	Mandatory	Description
name	String	200 chars max	yes	Name of a query which you would like to use for its identification. This could be any string matching constraint.
criteriaOverrides	NIX Query Settings	Sub-set of properties defining query parameters	no	Set of settings which should be used instead of ones defined in base settings. Basically this is sub-set of criteria from NIX Query Settings. All fields have the same meaning and constraints.
queryInput	Text input stream	Content-Type: text/* (supplied as the part of multipart request)	yes	The part under this name should hold a query input which is going to be processed by queries which accept user input (like group searches or nix queries).

Example of settings part for post

```
{
 "name": "myNixQueryNumberOne"
}
```

Group Query Definition

This object is used to define group search task for execution. It is used for performing searches and making dry runs. Basically it holds the reference to settings which should be used and criteria overrides. Under overrides we mean the set of fields form the Group Search Settings object which relate to selection **Criteria**. The values specified within the Query object have precedence over similar values defined in bound settings instance. This object is used only for submitting the query for execution. You will not be able to get it after submitting. Instead you will be able to get Group Search Settings object which was build from referenced settings and overrides you have supplied as well as query metadata itself as Query object.

Type Attributes

Name	Type	Constraints	Mandatory	Description
settingsTemplateId	String	UUID	yes	ID of settings you owe which should be used as basic settings for your search. Usually this one is the String representation of UUID.
name	String	200 chars max	yes (for issuing query)	Name of query which you would like to use for identifying the query. This could be any string matching constraint. For dry run name of query is not required.
criteriaOverrides	Group Search Settings	Sub-set of properties defining query parameters	no	Set of settings which should be used instead of ones defined in base settings. Basically this is sub-set of criteria from Group Search Settings. All fields have the same meaning and constraints
queryInput	Group Search Questions		yes (if issuing without preloaded file)	With this attribute you define an input of the group search process. If you are not using POST with **questionsFileId** parameter to start the query, then this parameter should be defined in a separate part of multipart request. Part name should have the name 'queryInput'. Setting in this case should be defined in part having 'setting' name.

Example of Settings part for POST

```
{
 "name": "myNewName",

 "settingsId": "064edf38-c9fd-102b-9d1c-0030487dabd6"
 "criteriaOverrides": {
  "outEmptyWhenNotFound": "true"
 }
}
```

Query Result

Private Prospects Query Result

The structure actually describes private persons query result. It does not hold result itself, instead it holds ID which then could be used to retrieve result by accessing corresponding resource. Also this structure is used to update some properties of your result (like hidden/charged). When putting it to service you don't have to define all the fields - only those which are going to be changed.

Type Attributes

Name	Type	Constraints	Mandatory	Description
id	String		yes (when getting)	Result id. Usually this one is a String representation of UUID.
queryId	String		yes (when getting)	Id of query the result is bound to.
versionCode	String		yes (for PUT and GET)	ID of version. This ID is used by the system in order to determine conflicts (see Handling conflicts).
createdAt	Date	Read Only	yes (when getting)	Date of when the query result was initially created. You cannot modify the value of this field.
modifiedAt	Date	Read Only		Date of when the query result was last time modified. You cannot modify the value of this field.
status	Enum String	ReadOnly, one of SUCCESS, FAILURE	yes (when getting)	Date of when the setting was last time modified. You cannot modify the value of this field.
answerCount	Integer	positive	yes (if status == SUCCESS)	Contains number of answers returned in result. If query execution has failed, this value is not defined.
mobileAnswerCount	Integer	positive	yes (if status == SUCCESS)	Contains number of answers in result which has mobile phones associated.
errorReason	String		yes (if status == FAILURE)	Holds description of an error which prevented the query from being successfully executed.
hidden	Boolean	true/false	yes (when getting)	Determines whether this result is hidden or not. Note, it still be returned in the list of results, so any handling is client specific.
charged	Boolean	true/false	yes (when getting)	Determines whether charging for getting the results has happened or not.
locked	Boolean	true/false	yes (when getting)	Determines whether your result is locked or not. Locked results cannot be extracted. Please note that this property can only be changed by users having admin privileges. By default all results are not locked. A user with admin access can change result setting to locked = true.
considerInExcludes	Boolean	true/false	yes (when getting)	By default results take part in further selection as excluded ones in case if user has decided to use **excludeBy** criteria. Although you can mark some results as not taking part in excludes if you PUT the value of this attribute == "false".

JSON Representation

```
{
 "id": "String",
 "queryId": "String",
 "versionCode": "String",
 "createdAt": "<Date_in_format ISO-8601>",
    "modifiedAt" : "<Date_in_format ISO-8601>",
    "status": "<String_one_of: SUCCESS, FAILURE>",
 "answerCount": Number,
 "mobileAnswerCount": Number,
 "errorReason": "String",
 "hidden": "Boolean",
 "considerInExcludes": "Boolean",
 "locked": "Boolean",
 "charged": "Boolean"
}
```

Example for GET

```
{
 "id": "064ed556-c9fd-102b-9d1c-0030487dabd6",
 "queryId": "064ed556-c9fd-102b-9d1c-0030487dabd3",
 "versionCode": "064edf38-c9fd-102b-9d2c-0030487dabd6",
 "createdAt": "2015-02-215T16:30:44Z",
    "modifiedAt": "2015-02-215T16:30:46Z",
    "status": "SUCCESS",
 "answerCount": 133,
 "mobileAnswerCount": 111,
 "hidden": "false",
 "considerInExcludes": "true",
 "locked": "false",
 "charged": "true"
}
```

Example for PUT

```
{
 "id": "064ed556-c9fd-102b-9d1c-0030487dabd6",
 "versionCode": "064edf38-c9fd-102b-9d2c-0030487dabd6",
 "isHidden": "true",
 "considerInExcludes": "false",
 "locked": "false"
 }
```

Company Prospects Query Result

The structure actually describes company query result. It does not hold result itself, instead it holds ID which then could be used to retrieve result by accessing corresponding resource. Also this structure is used to update some properties of your result (like hidden/charged). When putting it to service you don't have to define all the fields - only those which are going to be changed.

Type Attributes

Name	Type	Constraints	Mandatory	Description
id	String		yes (when getting)	Result id. Usually this one is a String representation of UUID.
queryId	String		yes (when getting)	Id of query the result is bound to.
versionCode	String		yes (for PUT and GET)	ID of version. This ID is used by the system in order to determine conflicts (see Handling conflicts).
createdAt	Date	Read Only	yes (when getting)	Date of when the query result was initially created. You cannot modify the value of this field.
modifiedAt	Date	Read Only		Date of when the query result was last time modified. You cannot modify the value of this field.
status	Enum String	ReadOnly, one of SUCCESS, FAILURE	yes (when getting)	Date of when the setting was last time modified. You cannot modify the value of this field.
answerCount	Integer	positive	yes (if status == SUCCESS)	Contains number of answers returned in result. If query execution has failed, this value is not defined.
errorReason	String		yes (if status == FAILURE)	Holds description of an error which prevented the query from being successfully executed.
hidden	Boolean	true/false	yes (when getting)	Determines whether this result is hidden or not. Note, it still be returned in list of results, so any handling is client specific.
considerInExcludes	Boolean	true/false	yes (when getting)	By default results take part in further selection as excluded ones in case if user has decided to use **excludeBy** criteria. Although you can mark some results as not taking part in excludes if you PUT the value of this attribute == "false".
charged	Boolean	true/false	yes (when getting)	Determines whether your result is locked or not. Locked results cannot be extracted. Please note that this property can only be changed by users having admin privileges.
locked	Boolean	true/false	yes (when getting)	Determines whether your result is locked or not. Locked results cannot be extracted. Please note that this property can only be changed by users having admin privileges. By default all results are not locked. A user with admin access can change result setting to locked= true.
extraAnswerCount	Integer	positive	yes (if status == SUCCESS)	Number of extra phone numbers you've got in your query results. 0 if you did not requested extra phone numbers
branchAnswerCount	Integer	positive	yes (if status == SUCCESS)	Number of branches you've got in your query results. 0 if you did not requested extra phone numbers

JSON Representation

```
{
 "id": "String",
 "queryId": "String",
 "versionCode": "String",
 "createdAt": "<Date_in_format ISO-8601>",
    "modifiedAt" : "<Date_in_format ISO-8601>",
    "status": "<String_one_of: SUCCESS, FAILURE>",
 "answerCount": Number,
 "errorReason": "String",
 "hidden": "Boolean",
 "considerInExcludes": "Boolean",
 "locked": "Boolean",
 "charged": "Boolean",
 "extraAnswerCount": Number
 }
```

Example for GET


```
{
 "id": "064ed556-c9fd-102b-9d1c-0030487dabd6",
 "queryId": "064ed556-c9fd-102b-9d1c-0030487dabd3",
 "versionCode": "064edf38-c9fd-102b-9d2c-0030487dabd6",
 "createdAt": "2015-02-215T16:30:44Z",
    "modifiedAt": "2015-02-215T16:30:46Z",
    "status": "SUCCESS",
 "answerCount": 133,
 "hidden": "false",
 "considerInExcludes": "true",
 "charged": "true",
 "extraPhoneAnswerCount": 13300
}
```


Example for PUT


```
{
 "id": "064ed556-c9fd-102b-9d1c-0030487dabd6",
 "versionCode": "064edf38-c9fd-102b-9d2c-0030487dabd6",
 "hidden": "true",
 "considerInExcludes": "false"
 "locked" : "true"
}
```


Group Query Result

The structure actually describes group search query result. It does not hold result itself, instead it holds ID which then could be used to retrieve result by accessing corresponding resource. Also this structure is used to update some properties of your result (like hidden/charged). When putting it to service you don't have to define all the fields - only those which are going to be changed.

Type Attributes

Name	Type	Constraints	Mandatory	Description
id	String		yes (when getting)	Result id. Usually this one is a String representation of UUID.
queryId	String		yes (when getting)	Id of query the result is bound to.
versionCode	String		yes (for PUT and GET)	ID of version. This ID is used by the system in order to determine conflicts (see Handling conflicts).
createdAt	Date	Read Only	yes (when getting)	Date of when the query result was initially created. You cannot modify the value of this field.
modifiedAt	Date	Read Only		Date of when the query result was last time modified. You cannot modify the value of this field.
status	Enum String	ReadOnly, one of SUCCESS, FAILURE	yes (when getting)	Date of when the setting was last time modified. You cannot modify the value of this field.
answerCount	Integer	positive	yes (if status == SUCCESS)	Total number of answers we have generated for result questions set. If query execution has failed, this value is not defined.
errorReason	String		yes (if status == FAILURE)	Holds the description of error which prevented the query from being successfully executed.
hidden	Boolean	true/false	yes (when getting)	Determines whether this result is hidden or not. Note, it still be returned in list of results, so any handling is client specific.
charged	Boolean	true/false	yes (when getting)	Determines whether your result is locked or not. Locked results cannot be extracted. Please note that this property can only be changed by users having admin privileges.

locked	Boolean	true/false	yes (when getting)	Determines whether your result is locked or not. Locked results cannot be extracted. Please note that this property can only be changed by users having admin privileges. By default all results are not locked. A user with admin access can change result setting to locked= true.
questionsCount	Integer	positive	yes (if status == SUCCESS)	Number of questions which we have managed to recognize as valid ones and were looking for answers to them.
answeredCount	Integer	positive	yes (if status == SUCCESS)	Number of questions for which we have supplied at least one answer.
updatedCount	Integer	positive	yes (if status == SUCCESS)	Number of updated rows in case if query was launched with 'update' feature ON.

Example for GET

```
{
 "id": "064ed556-c9fd-102b-9d1c-0030487dabd6",
 "queryId": "064ed556-c9fd-102b-9d1c-0030487dabd3",
 "versionCode": "064edf38-c9fd-102b-9d2c-0030487dabd6",
 "createdAt": "2015-02-215T16:30:44Z",
    "modifiedAt": "2015-02-215T16:30:46Z",
    "status": "SUCCESS",
 "answerCount": 14444,
 "hidden": "false",
 "charged": "true",
 "questionsCount" : 18678,
 "answeredCount": 13333
}
```

Example for PUT

```
{
 "id": "064ed556-c9fd-102b-9d1c-0030487dabd6",
 "versionCode": "064edf38-c9fd-102b-9d2c-0030487dabd6",
 "hidden": "true",
 "locked" : "true"
}
```

NIX Query Result

The structure actually describes group search query result. It does not hold result itself, instead it holds ID which then could be used to retrieve result by accessing corresponding resource. Also this structure is used to update some properties of your result (like hidden/charged). When putting it to service you don't have to define all the fields - only those which are going to be changed.

Name	Type	Constraints	Mandatory	Description
id	String		yes (when getting)	Result id. Usually this one is a String representation of UUID.
queryId	String		yes (when getting)	Id of query the result is bound to.
versionCode	String		yes (for PUT and GET)	ID of version. This ID is used by the system in order to determine conflicts (see Handling conflicts).
createdAt	Date	Read Only	yes (when getting)	Date of when the query result was initially created. You cannot modify the value of this field.
modifiedAt	Date	Read Only		Date of when the query result was last time modified. You cannot modify the value of this field.
status	Enum String	ReadOnly, one of SUCCESS, FAILURE	yes (when getting)	Date of when the setting was last time modified. You cannot modify the value of this field.
answerCount	Integer	positive	yes (if status == SUCCESS)	Total number of answers we have generated for result questions set. If query execution has failed, this value is not defined.

errorReason	String		yes (if status == FAILURE)	Holds the description of error which prevented the query from being successfully executed.
hidden	Boolean	true/false	yes (when getting)	Determines whether this result is hidden or not. Note, it still be returned in list of results, so any handling is client specific.
charged	Boolean	true/false	yes (when getting)	Determines whether your result is locked or not. Locked results cannot be extracted. Please note that this property can only be changed by users having admin privileges.
locked	Boolean	true/false	yes (when getting)	Determines whether your result is locked or not. Locked results cannot be extracted. Please note that this property can only be changed by users having admin privileges. By default all results are not locked. A user with admin access can change result setting to locked= true.
nixItemsCount	Integer	positive	yes (when getting)	Defines how many nixed phones were detected among phones supplied for query

Example for GET


```
{
 "id": "064ed556-c9fd-102b-9d1c-0030487dabd6",
 "queryId": "064ed556-c9fd-102b-9d1c-0030487dabd3",
 "versionCode": "064edf38-c9fd-102b-9d2c-0030487dabd6",
 "createdAt": "2015-02-215T16:30:44Z",
    "modifiedAt": "2015-02-215T16:30:46Z",
    "status": "SUCCESS",
 "answerCount": 133,
 "hidden": "false",
 "charged": "true",
 "nixItemsCount": 122
}
```


Example for PUT


```
{
 "id": "064ed556-c9fd-102b-9d1c-0030487dabd6",
 "versionCode": "064edf38-c9fd-102b-9d2c-0030487dabd6",
 "hidden": "true",
 "locked" : "true"
}
```


Query Result Value

This data type represents actual results of the query being successfully performed. The structure holds collection of answers where each answer holds the information as defined below.

Private Prospects Query Result Value

Type Attributes

Name	Type	Result format		Mandatory	Description
		CSV	JSON		
id	Integer			yes	Id of answer in the set. This is not unique across several results and is unique only in scope of the returned answers collection.
firstName	String			yes	First name of a person matched selection criteria.
lastName	String			yes	Last name of a person matched selection criteria.
mainFirstName	String			yes	Main first name of a person matched selection criteria.
birthDate	Date	YYYYMMDD	ISO-8601 date format	yes	Birth date of a person matched selection criteria.

street	String			yes	Name of street the person is registered to.
houseNum	String			yes	Number of house the person is registered to.
houseNumSuffix	String			no	House Number Suffix the person is registered to.
flat	String			no	Flat number or caption
zipCode	Integer	5 digits	5 digits	yes	Zip Code of address the person is registered to.
zipCity	String			yes	Name of postal City the person is registered to.
phone	String	National format with area separated from phone number <area code>-<phone number> '-' (hyphen) - in case of number is not returned according to query settings	Phone number in international format without separators. The field is absent if phone number should not be return according to query settings.	no	Phone number of person matched selection criteria. This could be mobile or fixed depending on selection criteria. Presence of phone number also depends on selection criteria.
phoneNixed	Boolean	'Nix telefon' or '-' (hyphen)	true/false the field is absent or null if mode is not NIX_MARK	no	If there was defined for settings nixPhonesFilteringMode=NIX_MARK, the value shows whether the phone returned is found in NIX or not.
addressNixed	Boolean	'Nix adress' or '-' (hyphen)	true/false the field is absent or null if mode is not NIX_MARK	no	If there was defined for settings nixAddressFilteringMode=NIX_MARK, the value shows whether the address returned is found in NIX or not.
municipality	String			yes	Name of municipality the person is registered to.
sex	String	man/kvinna	MALE/FEMALE	yes	Defines sex of the person matched selection criteria.

Please note, that current implementation is using TABs as separators, so they are displayed below as '–>'. Also if there is no value for some column the value is replaced by '-'.

CSV Representation

```
id-->firstName-->lastName-->mainFirstName-->birthDate-->street-->houseNum-->houseNumSuffix-->flat-->zipCode-->zipCity-->phone-->phoneNixed-->addressNixed-->municipa
lity-->sex
```

CSV represenation example

```
id firstName lastName mainFirstName birthDate street houseNum houseNumSuffix flat zipCode zipCity phone phoneNixed addressNixed municipality sex
1 Hans Söderberg Gunnar  19520412 Banérgatan 12   13555 Älvsjö - - - STOCKHOLM man
2 Mikael Olof Alarud  19930215 Klippgatan 16 0801  146566 Hässelby - - - STOCKHOLM man
3 Johan Fredrik Regland   19440323 Guldregnsbacken 1  1604 11422 Stockholm - - - STOCKHOLM kvinna
4 Karl Viking Bengtsson 19990319 Aprikosgatan 333 C 1351 17787 Solna - - - SOLNA man
```

JSON representation example

```
{
 "id": 1,
    "firstName": "Hans",
    "lastName": "Söderberg",
    "mainFirstName": "Gunnar",
    "birthDate": "1952-04-12T00:00:00.000Z",
    "street": "Banérgatan",
    "houseNum": "12",
    "houseNumSuffix": null,
    "zipCode": 13555,
    "zipCity": "Älvsjö",
    "municipality": "STOCKHOLM",
    "phone": "String",
    "sex": MALE,
    "phoneNixed": true,
    "addressNixed": false
}
```

Company Prospects Query Result Value / Company Prospects Flat Result Value

Information about companies is displayed differently in csv and json formats. On the server side different objects are used for storing company information to csv and json. They are a flat object and an object with nested object fields respectively. All missing values are returned as empty string in CSV format.

In the table below difference is marked in CSV and JSON column

Type Attributes

Name (json / csv fields)	Type	Result format		Mandatory	Description
		CSV	JSON		
id	Integer			yes	Id (position) of answer in the set. This is not unique across several results. In json it is unique only in scope of the returned answers collection. In csv it stays the same for all lines representing headquarters with additional information and all branches of the same company, though it is unique for companies withing single result collection.
name	String			yes	Name of the company.
orgNum	String		10 digits	yes	Organization number of the company.
cfar	String		8 digits	no	SCB's Business Register's eight-digit identity for workplaces.
propertyForm	String			yes	Legal form of a company matching selection criteria.
street	String	empty if not presented		no	Name of street of the company postal address.
houseNum	String	empty if not presented		no	Number of house of the company postal address.
houseNumSuffix	String	empty if not presented		no	House number suffix of the company postal address.
flat	String	empty if not presented		no	Flat number or its caption
zipCode	Integer	5 digits or empty if not presented	5 digits	no	Zip code of the company postal address.
zipCity	String	empty if not presented		no	Name of city of the company postal address.
visitAddressStreet	String	empty if not presented		no	Name of street the company is located in.
visitAddressHouseNum	String	empty if not presented		no	Number of house the company is located in.
visitAddressHouseNumSuffix	String	empty if not presented		no	House number suffix of a house the company is located in.
visitFlat	String	empty if not presented		no	Flat number or its caption

visitAddressZipCode	Integer	5 digits or empty if not presented	5 digits	no	Zip code of address the company is located at.
visitAddressZipCity	String	empty if not presented		no	Name of postal City the company is located at.
phone	String	National format with area separated from phone number <area code>-<phone number> or empty if not presented	Phone number in international format without separators.	no	Phone number of a company matching selection criteria. This could be mobile or fixed depending on selection criteria. Also this column may hold "alternative" number if corresponding option chosen when defining query parameters
phoneType	Enum	One of: • OFFICIAL • BRANCH • PRIVATE • COMPANY • OTHER	One of: • OFFICIAL • BRANCH • PRIVATE • COMPANY • OTHER	no	For a headquarters, if returning alternative phone is chosen then this column describes what kind of phone is returned: • OFFICIAL - official company phone we got from authorities • BRANCH - phone of a branch of the company • PRIVATE - if allowed - private nix-free number related to the company • COMPANY - other phone we know for the company • OTHER - other kind of phone If returning alternative numbers is not allowed and phone is present - OFFICIAL type is going to be assigned For extra phone items, the phone type can be one of the following: • PRIVATE • COMPANY • OTHER For branches, a BRANCH type is present if a phone number is present.
sniCode	String	2-5 digits or empty if not presented	2-5 digits	no	SNI code of a company matching selection criteria.
sniDescription	String	empty if not presented		no	Textual description of an SNI code for a company matching selection criteria.
turnover	Long			no	Turnover value of the company (in 1000's SEK). It is specified as '0' when data is not present.
shareCapital	Long			no	Value of shareCapital of a company (in 1000's SEK). It is specified as '0' when data is not present.
finReportDate	Date	YYYY-MM-DD or hyphen if not presented	ISO-8601 date format	no	Date of the last financial report of a company.
employeesCount	Integer			no	Number of employees of the company. It is specified as '0' when data is not presented.
regDate	Date	YYYY-MM-DD or hyphen if not presented	ISO-8601 date format	no	Registration date of the company.
extra phone items	List	***NOT PRESENTED***		no	Json only: see Company Branch Result Value (Json representation)
branches	List	***NOT PRESENTED***		no	Json only: see Company Additional Result Value (Json representation)
fullName	String		***NOT PRESENTED***	no	CSV only: Additional item.Branch name.
bearerName	String		***NOT PRESENTED***	no	CSV only: Additional item. Owner name of a company matching selection criteria.
additionalId	Integer		***NOT PRESENTED***	yes when csv result is an extra phone item. Otherwise no.	CSV only: Additional id. Index of record among additional phones for a single company. It is unique only within lines with additional info of a single company.
branchId	Integer		***NOT PRESENTED***	yes if csv result item is a branch. Otherwise no.	CSV only: Branch id. It is unique only within branches of a single company.

Company Branch Result Value (Json representation)

Name	Type	Result format	Mandatory	Description
id	Integer		yes	Id (position) of answer in the set. This is not unique across several results. In Json it is unique only in scope of the returned answers collection. In csv it stays the same for all lines representing headquarters with additional information and all branches of the same company, though it is unique for companies within single result collection.
cfar	String	8 digits	no	SCB's Business Register's eight-digit identity for workplaces.
name	String		yes	Name of the company.

street	String		no	Name of street of the company postal address.
houseNum	String		no	Number of house of the company postal address.
houseNumSuffix	String		no	House number suffix of the company postal address.
flat	String		no	Flat number or its caption
zipCode	Integer	5 digits	no	Zip code of the company postal address.
zipCity	String		no	Name of city of the company postal address.
visitAddressStreet	String		no	Name of street the company is located in.
visitAddressHouseNum	String		no	Number of house the company is located in.
visitAddressHouseNumSuffix	String		no	House number suffix of a house the company is located in.
visitFlat	String		no	Flat number or its caption the company located in.
visitAddressZipCode	Integer	5 digits	no	Zip code of address the company is located at.
visitAddressZipCity	String		no	Name of postal City the company is located at.
phone	String	Phone number in international format without separators.	no	Phone number of a company matching selection criteria. This could be mobile or fixed depending on selection criteria.
sniCode	String	2-5 digits	no	SNI code of company matching selection criteria.
sniDescription	String		no	Textual description of an SNI code for a company matching selection criteria.

Company Additional Result Value (Json representation)

Name	Type	Result format	Mandatory	Description
id	Integer		yes	Index of record among additional phones for a single company. It is unique only within collection of additional items.
extraFullName	String		no	Name of company or phone holder
extraBearerName	String		no	Owner name of a company matching selection criteria.
extraPhone	String		no	Additional phone in the company.

JSON representation example

```
 [{
"id":1,
"name":"Company Name AB",
"cfar": "00000000",
"street":"STREET",
"houseNum":"6",
"houseNumSuffix":null,
"flat": null,
"zipCode":99999,
"zipCity":"CITY",
"visitStreet":"STREET",
"visitHouseNum":"2",
"visitHouseNumSuffix":null,
"visitFlat": null,
"visitZipCode":999999,
"visitZipCity":"CITY",
"phone":"46000000000",
"sniCode":"00000",
"sniDescription": "SNI 00000 Description",
"orgNum":"0000000000",
"regDate":"1985-11-18T00:00:00.000Z",
```

```
"propertyForm":"foundation",
"turnover":1239979128253978605,
"employeesCount":228681466,
"shareCapital":8131557919626350773,
"finReportDate":"1951-12-11T00:00:00.000Z",
"branches":[{
 "id":1,
    "cfar": "00000000",
    "name":"Branch Name AB",
 "street":"STREET",
 "houseNum":"2",
 "houseNumSuffix":null,
    "flat": null,
    "zipCode":99999,
 "zipCity":"CITY",
 "visitStreet":"STREET",
 "visitHouseNum":"3",
    "visitHouseNumSuffix":null,
    "visitFlat": null,
    "visitZipCode":99999,
 "visitZipCity":"CITY",
 "phone":"460000000",
 "sniCode":"00000",
 "sniDescription":"SNI 00000 Description"
}],
"additionalItems":[{
 "id":1,
"extraFullName":"Full Name",
"extraBearerName":"Bearer Name",
"extraPhone":"4600000000",
 "companyId":"5c2a090f-6d5c-4b94-ada7-a356172f1dc3"},
{"id":2,
"extraFullName":"Full Name",
"extraBearerName":"Bearer name AB",
```

```
 "extraPhone":"460000000",
 "companyId":"5c2a090f-6d5c-4b94-ada7-a356172f1dc3"}]
}]
```

Please note, that current implementation is using TABs as separators.

CSV representation example (delimeter = tab, line delimeter = \r\n)

```
id    name    orgNum        cfar    propertyForm    street    houseNum    houseNumSuffix     zipCode    zipCity    visitAddressStreet    visitAddressHouseNum
visitAddressHouseNumSuffix visitFlat visitAddressZipCode    visitAddressZipCity    phone    phoneType    sniCode    sniDescription    turnover    shareCapital
finReportDate    employeesCount    regDate    fullName    bearerName    additionalId    branchId
1    Company_1    0000000000        00000000    other    STREET_1    1        1 99999    CITY_1    VISIT_STREET_1    1        88888    VISIT_CITY_1    093-000000
OFFICIAL    00000    SNI_00000_Description    3163435436498572951    6098604510667083778    1920-08-07    1000874036    1973-06-16
1    Company_1    0000000000        00000000    other    STREET_1    1        1 99999    CITY_1    VISIT_STREET_1    1        88888    VISIT_CITY_1    039-000000
OTHER    00000    SNI_00000_Description    3163435436498572951    6098604510667083778    1920-08-07    1000874036    1973-06-16    FULL_NAME_1    BEARER_NAME_1    1

1    Company_1    0000000000        00000000    other    STREET_1    1        1 99999    CITY_1    VISIT_STREET_1    1        88888    VISIT_CITY_1    094-000000
PRIVATE    00000    SNI_00000_Description    3163435436498572951    6098604510667083778    1920-08-07    1000874036    1973-06-16    FULL_NAME_2    BEARER_NAME_2
2
1    Company_1    0000000000        00000000    other    STREET_1    1        1 99999    CITY_1    VISIT_STREET_1    1        88888    VISIT_CITY_1    096-000000
COMPANY    00000 SNI_00000_Description    3163435436498572951    6098604510667083778    1920-08-07    1000874036    1973-06-16    FULL_NAME_4    BEARER_NAME_4    3

2    Company_2    1111111111        11111111    other    STREET_2    1        12 77777    CITY_2    VISIT_STREET_2    1        66666    VISIT_CITY_2    070-0000000
BRANCH    11111    SNI_11111_Description    2684081246197958078    7043950280576607816    1995-01-23    1183623074    1950-05-06
2    Company_2    1111111111        11111111    other    STREET_2    1        12 77777    CITY_2    VISIT_STREET_2    1        66666    VISIT_CITY_2    070-0000000
COMPANY    11111    SNI_11111_Description    2684081246197958078    7043950280576607816    1995-01-23    1183623074    1950-05-06    FULL_NAME_5    BEARER_NAME_5
070-00000    1
2    Branch_1_of_Company_2    2222222222    22222222    other    STREET_3    58        55555    CITY_3    VISIT_STREET_3    2        22222    VISIT_CITY_3
011-000000 BRANCH    22222    SNI_22222_Description    2684081246197958078    7043950280576607816        1183623074    1950-05-06                1
2    Branch_2_of_Company_2    3333333333        other    STREET_4    2        44444    CITY_4                        085-0000000    BRANCH    33333
SNI_33333_Description    2684081246197958078    7043950280576607816    1995-01-23    1183623074                        2
```

Group Query Result Value

Result values information differs for CSV and JSON format.

Depending on settings, the result CSV can contain empty result rows. They represent not answered questions. Empty rows look differently depending on different settings:

1. No update modes specified. Such a row will contain only input question id.
2. There is any of the update modes specified in the settings. The row will contain a question id and question fields and also 'not updated' statement in the correspondent fields.

CSV Format Group Query Result Value

Please note that we are using TAB separated value now when supplying an output.

STANDARD (NOT TRANSPOSED REPRESENTATION)

In a standard CSV representation each line has 1 result value information. Result values are ordered by input question ids.

Name	Type	Constraints	Mandatory	Description
questionId	String	Unique among questions	yes	ID of question to which this answer relates
firstName	String		no	First name of matching person (person result value) or a bearer of company phone (company result value)

lastName	String		no	Last name of matching person (person result value) or a bearer of company phone (company result value)
orgName	String		no	Name of matching company
regNum	String	12 digits SSN/10 digits org Number	no	12 digits SSN of matching person or 10 digits registration number of company
street	String		no	Street name or "box", "pl", "mailbox" and others.
houseNum	String		no	House number, street number, box number, mailbox etc.
suffix	String		no	House number suffix, anything that remains after House number in address, but not a flat number.
flat	String		no	Flat number or its caption.
zipCode	String		no	Holds zip code of search target area.
zipCity	String		no	City of search target area.
addressNixed	Boolean	yes/no	no	If you have requested nix-marking for an address than this field will hold marker notifying the address is within the NIX registry
phone	String		no	Phone number.
phoneNixed	Boolean	yes/no	no	If you have requested nix-marking for a phone than this field will hold marker notifying that a phone is within the NIX registry
matchingTemplate	String		no	Name of Matching template which has shown that this answers your question
sequenceNo	Integer	1+	no	Increasing number which shows an order of answer in scope of a single question. If you have ex. 5 answers to a single question, then you'll experience this field raising its value from 1 to 5.
updated	Boolean	address phone name/not updated	no	If you have requested update input feature than this column will show you that we have found new information on your question and returning it within the row. The field will have as least one of the words depending on the specified update modes in the settings. The words are space-separated: 'address phone name'. If any of update modes is specified in the settings and there is a requirement to return empty result rows, such a line will have 'not updated' value stated beside other fields from the initial question.

Please note that company can have more that one address (they can be visit and mail addresses). Result value will contain the one matching with the input question if any of address fields are specified in the question. If no address fields are specified in the question, any of company addresses can be present in the result.

Standard CSV representation example

```
questionId firstName lastName orgName birthDate regNum street houseNum suffix flat zipCode zipCity addressNixed phone phoneNixed matchingTemplate sequenceNo updated
0 Carl Magnus Liljeqvist  1965-07-21 196507212491 Annikas gata 8   42167 Västra Frölunda no 0732252200 no 1140 1
0 Carl Magnus Liljeqvist  1965-07-21 196507212491 Annikas gata 8   42167 Västra Frölunda no 0736010615 no 1140 2
0 Carl Magnus Liljeqvist  1965-07-21 196507212491 Lindholmspiren 9   41756 Göteborg no 0736111605 no 1140 3
0 Carl Magnus Liljeqvist  1965-07-21 196507212491 Annikas gata 8   42167 Västra Frölunda no 032143246 no 1140 4
0   Magnus Liljeqvist  196507212491 Annikas Gata 8   42167 Västra Frölunda no 0331346 no 1140 5
10 Carl Magnus Liljeqvist  1965-07-21 198407213192 Annikas gata 71 A 12 62167 Stockgolm no 0732525439 no 1140 1
10 Carl Magnus Liljeqvist  1965-07-21 198407213192 Annikas gata 71 A 12 62167 Stockgolm no 0311433454 no 1140 2
10 Carl Magnus Liljeqvist  1965-07-21 198407213192 Annikas gata 71 A 12 62167 Stockgolm no 0732767605 no 1140 3
10 Carl Magnus Liljeqvist  1965-07-21 198407213192 Lindholmspiren 11   48158 Göteborg no 0732767605 no 1140 4
10   Magnus Liljeqvist  198407213192 Annikas Gata 99   42167 Västra Frölunda no 0311433454 no 1140 5
```

TRANSPOSED REPRESENTATION

In a transposed CSV view all answers to a certain question are situated in the same row. The first answer will look the same as in a standard CSV view. All other answers are returned right after the last column of the first answer. They will have the following fields:

Name	Type	Mandatory	Description
phone	String	yes	Phone number
phoneNixed	Boolean	no	If you have requested nix-marking for an address than this field will hold marker notifying the address is within the NIX registry

Transposed CSV representation example

```
questionId firstName lastName orgName birthDate regNum street houseNum suffix flat zipCode zipCity addressNixed phone phoneNixed matchingTemplate sequenceNo updated
phone1 phoneNixed1 phone2 phoneNixed2 phone3 phoneNixed3 phone4 phoneNixed4
0 Carl Magnus Liljeqvist  1991-07-11 199107112491 Annikas gata 65  42168 Västra Frölunda no 0732171700 no 1140   0736496605 no 0793450605 no 031453149 no 037544846
no
10 Carl Magnus Liljeqvist  1991-07-11 199107112491 Annikas gata 65  42168 Västra Frölunda no 0732171700 no 1140   035843146 no 0736690605 no 0736000300 no 031000146
no
8 Carl Magnus Liljeqvist  1991-07-11 199107112491 Annikas gata 65  42168 Västra Frölunda no 033143657 no 1140   0736801605 no 0736080675 no 0732521111 no 031185246
no
9 Carl Magnus Liljeqvist  1991-07-11 199107112491 Lindholmspiren 13  1001 41756 Göteborg no 0736987605 no 1140   0732520050 no 031147747 no 0736250675 no 031193187
no
```

JSON Format Group Query Result Value

Name	Type	Mandatory	Description
`groupQuestionInfo`	Group Question Info	yes	Is a wrapper of Group Question object. See Group Question.
groupAnswers	List	no	can be empty if no results found for a particular question and Group Search Setting outEmptyWhenNotFound = true. See Group Answer

GROUP QUESTION

At least one of non-mandatory fields should be present in the question.

Name	Type	Constraints	Mandatory	Description
id	String	Unique among questions	yes	ID of question to which this answer relates
firstName	String		no	First name of a person
lastName	String		no	Last name of a person
orgName	String		no	Company name
birthDate	Date		no	Person's birthdate
regNum	Long		no	Person's ssn or company's organization number
street	String		no	Address street
houseNum	String		no	Address house number
houseNumSuffix	String		no	Address house number suffix
zipCode	Integer		no	Address zip code
location	String		no	Address city
phone	String		no	Phone of a person or a company in 46[digits] format

GROUP ANSWER

Name	Type	Constraints	Mandatory	Description
matchingTemplateId	String		yes	An id of a template used to match a result item against the input question
resultItem	Group Query Answer Item	any of PersonQueryResultItem, CompanyQueryResultItem	yes	See Person Result Item and Company Result Item
matchAddressType	Enum String	any of MAIL, VISIT	yes	Represents the type of company result item address used to match it against the question. MatchAddressType is null for person result item as it has only one address.
updated	Enum Set	any of UPDATE_ADDRESS, UPDATE_PHONE, UPDATE_NAME	no	If Group Search Setting outUpdateModes has any values, some of them can be used to match a result item. The ones used for matching are present in this set.

GROUP QUERY ANSWER ITEM

Person Result Item and Company Result Item represent group query answer item.

JSON representation example

```
{
  "questionInfo" : {
    "question" : {
      "id" : 21,
      "firstName" : "SomeName",
      "lastName" : null,
      "orgName" : null,
      "birthDate" : null,
      "regNum" : null,
      "street" : null,
      "houseNum" : null,
      "houseNumSuffix" : null,
      "flat":null,
      "zipCode" : null,
      "location" : "SomeCity",
      "phone" : null,
      "matchingTemplatesIds" : null
    }
  },
  "groupAnswers" : [
  {
      "matchingTemplateId" : "1111111111",
     "resultItem" : {
        "firstName" : "SomeName",
        "lastName" : "SomeLastName1",
        "mainFirstName" : null,
        "ssn" : 190000000000,
        "birthDate" : "2017-02-01T15:25:32.519Z",
        "street" : "SomeStreet1",
        "houseNum" : "1",
        "houseNumSuffix" : null,
            "flat" : null,
            "zipCode" : 11111,
        "zipCity" : "SomeCity",
        "phone" : "46701111111",
        "phoneAreaCode" : "070",
        "phoneNixed" : null,
        "addressNixed" : null,
        "municipality" : null,
        "sex" : "MALE"
      },
  "matchAddressType": null,
  "updated" : []
    }, {
```

```
   "matchingTemplateId" : "1111111112",
   "resultItem" : {
      "firstName" : "SomeName",
      "lastName" : "SomeLastName2",
      "mainFirstName" : null,
      "ssn" : 190000000001,
      "birthDate" : "2017-02-01T15:25:32.520Z",
      "street" : "SomeStreet2",
      "houseNum" : "1",
      "houseNumSuffix" : null,
          "flat" : "1001",
          "zipCode" : 11112,
     "zipCity" : "SomeCity",
      "phone" : "46701111112",
      "phoneAreaCode" : "070",
      "phoneNixed" : null,
      "addressNixed" : null,
      "municipality" : null,
      "sex" : "FEMALE"
    },
"matchAddressType": null,
```

```
  "updated" : []
  ]
}
```

NIX Query Result Value

Information about nixed phones is displayed in 2 different formats: JSON and CSV. Basically that are 2 representations of the same information on whether the phone is nixed or not

Type Attributes

Name	Type	Constraints	Mandatory	Description
phoneNumber	String		yes	Phone number which was checked against NIX registry. For CSV it is in National Format (0AAA-XXXXXXX). For JSON this on is as for other queries presented in international format without +
nixed	Boolean		yes	yes/no - in CSV, true/false in JSON

JSON representation example

```
[{
 "phoneNumber": "070-1234567",
    "nixed" : false
},
{
 "phoneNumber": "070-7654321",
    "nixed" : true
}]
```

CSV Representation

```
070-1234567 yes
070-7654321 no
```

Quick Search Results

This section describes data structures used to represent the results of Quick Search. At the moment quick search is capable returning information on Persons and Companies in Sweden. Both are returned in scope of single results array. Items could be distinguished not only by set of fields defined, but also by a '@type' filed value which could be:

- Person - when item represents information on person
- Company - when item represents information on organization

Person Quick Search Result Item

The result item is quite similar to one we use for Private Prospects Query Result Value.

Type Attributes

Name	Type	Constraints	Mandatory	Description
id	Integer	Positive. Starts with 0	yes	Id of answer in the set. This is just a sequence number of item in entire result set.
@type	String	Always 'Person'	yes	Defines a type of returned result item.
firstName	String		yes	First name of a person matched selection criteria.
lastName	String		yes	Last name of a person matched selection criteria.

mainFirstName	String		yes	Main first name of a person matched selection criteria.
birthDate	Date	ISO-8601 date format	yes	Birth date of a person matched selection criteria.
street	String		yes	Name of street the person is registered to.
houseNum	String		yes	Number of house the person is registered to.
houseNumSuffix	String		no	House Number Suffix the person is registered to.
flat	String		no	Flat number or caption
zipCode	Integer	5 digits	yes	Zip Code of address the person is registered to.
zipCity	String		yes	Name of postal City the person is registered to.
addressNixed	Boolean		yes	NIX flag for the address.
phone	String	Phone number in international format without separators and +.	yes	Phone number of person matched selection criteria. This could be mobile or fixed depending on selection criteria.
phoneAreaCode	String	Phone area code. Starts with 0.	yes	Phone area code. Starts with 0, e.g. 08.
phoneNixed	Boolean		yes	NIX flag for the phone number.
municipality	String		yes	Name of municipality the person is registered to.
sex	String	MALE/FEMALE	yes	Defines sex of the person matched selection criteria.
dwellingId	String		no	This is UUID-like identifier which could be used to group results by inhabitancy of the same dwelling. 2 cohabitants of same household will have the same dwelling id in case if we have this. Empty value value of the field could mean that we don't have information to which dwelling is the person related
cohabitants	Array	Collection if Person Quick Search Result Item	no	If it was required to return cohabitants, this one holds information on persons living in the same dwellings. They all should have the same dwelling id as its enclosing entry.

Company Quick Search Result Item

The result item describes the organization matching given criteria. Is a slightly enlightened version of Company Prospects Query Result Value for headquarters with some slight additions from extra phone items

Type Attributes

Name	Type	Constraints	Mandatory	Description
id	Integer	Positive. Starts with 1	yes	Id of answer in the set. This is just a sequence number of item in entire result set.
@type	String	Always 'Company'	yes	Type of item. Always the same for a specific item type.
name	String		yes	Name of the company.
orgNum	String	10-12 digits	yes	Organization number of the company.
cfar	String	8 digits	no	SCB's Business Register's eight-digit identity for workplaces.
propertyForm	String		yes	Legal form of a company matching selection criteria.
street	String		no	Name of street of the company postal address.
houseNum	String		no	Number of house of the company postal address.
houseNumSuffix	String		no	House number suffix of the company postal address.
flat	String		no	Flat number or its caption
zipCode	Integer	5 digits	no	Zip code of the company postal address.
zipCity	String		no	Name of city of the company postal address.
addressNixed	Boolean		no	NIX flag for the postal address.
visitStreet	String		no	Name of street the company is located in.
visitHouseNum	String		no	Number of house the company is located in.
visitHouseNumSuffix	String		no	House number suffix of a house the company is located in.
visitFlat	String		no	Flat number or its caption
visitZipCode	Integer	5 digits	no	Zip code of address the company is located at.
visitZipCity	String		no	Name of postal City the company is located at.
visitAddressNixed	Boolean		no	NIX flag for the visit address.
phone	String	Phone number in international format without separators and +.	yes	Phone number of a company matching selection criteria. This could be mobile or fixed depending on selection criteria.
phoneAreaCode	String	Phone area code. Starts with 0.	yes	Phone area code. Starts with 0, e.g. 08.

phoneKind	String	One of: • MOBILE • FIXED	yes	Phone kind of the phone number.
phoneType	Enum	One of: • OFFICIAL • BRANCH • PRIVATE • COMPANY • OTHER	no	For a headquarters, if returning alternative phone is chosen then this column describes what kind of phone is returned: • OFFICIAL - official company phone we got from authorities • BRANCH - phone of a branch of the company • PRIVATE - if allowed - private nix-free number related to the company • COMPANY - other phone we know for the company • OTHER - other kind of phone
phoneNixed	Boolean		no	NIX flag for the phone number.
sniCode	String	2-5 digits	no	SNI code of a company matching selection criteria.
sniDescription	String		no	Textual description of an SNI code for a company matching selection criteria.
turnover	Long		no	Turnover value of the company (in 1000's SEK). It is specified as '0' when data is not present.
shareCapital	Long		no	Value of shareCapital of a company (in 1000's SEK). It is specified as '0' when data is not present.
finReportDate	Date	ISO-8601 date format	no	Date of the last financial report of a company.
employeesCount	Integer		no	Number of employees of the company. It is specified as '0' when data is not presented.
regDate	Date	ISO-8601 date format	no	Registration date of the company.
fullName	String	Defined only for following phon types: PRIVATE/COMPANY/OTHER	no	Additional information on phone holder.
firstName				
lastName				

Dry Run

Here comes the description of structures which you get in scope of so called "dry run". Dry run is performed to get estimated number of answers to your query parameters. The structure is not intended to hold the answers themselves - only limited set of metadata describing result you get when you perform actual query run.

Person Query Dry Run Result

Type Attributes

Name	Type	Constraints	Mandatory	Description
answerCount	Integer	Positive	yes	Estimated number of answers you'll get after running the query
totalAnswersCount	Integer	Positive	yes	Total number of answers available matching your criteria

Example of Response Body for POST

```
{
 "answerCount": 50,
 "totalAnswersCount": 200
}
```

Company Query Dry Run Result

Type Attributes

Name	Type	Constraints	Mandatory	Description
answerCount	Integer	Positive	yes	Estimated number of answers you'll get after running the query
totalAnswersCount	Integer	Positive	yes	Total number of answers available matching your criteria
hqAnswerCount	Integer	Positive	yes	Estimated number of headquarters rows you should get if issue query request with same parameters

branchAnswerCount	Integer	Positive	yes	Estimated number of branch rows you should get if issue query request with same parameters
extraAnswersCount	Integer	Positive	yes	Estimated number of extra phones rows you should get if issue query request with same parameters

Example of Response Body for POST

```
{
 "answerCount": 156,
 "totalAnswersCount": 2000,
 "hqAnswerCount": 156,
 "branchAnswerCount": 200,
 "extraAnswersCount": 150
}
```

Group Query Dry Run Result

Type Attributes

Name	Type	Constraints	Mandatory	Description
questionsCount	Integer	Positive	yes	Number of question we have managed to parse from your input
answeredQuestionsCount	Integer	Positive	yes	Estimated Number of questions which have at least 1 answer
numberOfAnswers	Integer	Positive	yes	Estimated number of answers in total

NIX Search Questions

This structure describes questions you set to the iCatch system for processing. At the moment we support question submission in as single column CSV. Line endings could be LF, CRLF, CR.

> **Note**
> Our parsed does its best to detect phone number in input you have supplied, although you should try to keeping the rule "The more clear your input is, the more predictable results you'll get".

CSV Representation

```
Here come valid phones which will be accepted
46927266061
+46325382518
0240056000
0046250308668
06/90938958
025-0093233
'46340787599'
"46340787599"
(0492)365258
092-195-70-04
058 508 1751
4649:8947699
46954765086
+46798351820

Invalid numbers - this will be marked as parse error:
46954765086123
0046250312308668
0246000

Here we have something strange - this will be marked as parse error:
46431970758dsdsd
ssdd0431970758dsdsd
4665something3361211
This is@@@notAPhone
46977530712"
"46623289153
4a6b9c5d4e7f6g5h0i8j6k
```

NIX Input Check Result

This structure encapsulates results of checking validity of supplied NIX Query Input.

Type Attributes

Name	Type	Constraints	Mandatory	Description
totalItemsCount	Integer		yes	Total number of non empty input items detected within your input
validItemsCount	Integer		yes	Total number of valid lines which should be accepted by the query engine
duplicateItemsCount	Integer		yes	Number of items which are detected as duplicate ones within supplied input
totalErrorCount	Integer		yes	Total number of error items detected in your input
parseErrors	Collection of ParseError		yes	Holds mapping of items which have issues with format unrecognized entries to their index in input.

NIX Query Input Parse error report

This structure provides all the errors detected during the parse of supplied user input.

Name	Type	Constraints	Mandatory	Description
Line number	Integer	Starts with 1	yes	Number of line in original user import which we have failed to parse
Line	Integer		yes	String from original input we have failed to parse

Group Search Questions

This structure describes questions you set to the iCatch system for processing. At the moment we support question submission in CSV format. Values you supply within the questions object should be consistent with matching templates you are going to use. For example, if you have specified a matching template which performs matching by SSN and supply input which doesn't hold any SSN you will just get no answers, because questions with insufficient criteria will be skipped.

Type Attributes

Name	Type	Constraints	Mandatory	Description
id	String	Unique among questions	yes	This field is used to identify each particular question within your given questions set. In result you'll get answers which will hold values taken from this field to identify which exact question each answer is generated for.
firstName	String		no	Holds first name if you are looking for private persons. Not used in companies selections
lastName	String		no	Holds last name if you are looking for private persons. Not used in companies selections
orgName	String		no	Holds name of organization if you are looking for such. Not used in persons selections.
birthDate	Date	YYYYMMDD	no	Holds birth date if you need to perform matching. Despite we consider that date will be defined in format 'YYYYMMDD' we still counting that input can contain separator chars, so we remove all non digits from this column value before using it.
regNum	String	10/12 digits SSN/org Number	no	Holds 10 or 12 digits SSN when looking for persons or organization number when looking for companies. Despite we consider that reg number will be defined as digits only string we still counting that input can contain separator chars, so we remove all non digits from this column value before using it.
street	String		no	Street name or "box", "pl", "mailbox" and others.
houseNum	String		no	House number, Street number, box number, mailbox etc.
houseNumSuffix	String		no	Suffix of house number
flat	String		no	Flat name or caption
zipCode	String		no	Holds zip code of search target area.
location	String		no	Name of PostalCity of search target area.
phone	String		no	Phone number.

Please note that we are using TAB separated value now by default, although Input could be either comma, or semi-colon or pipe separted

Example in text/csv

```
1    Stephan           78234367823    Broadway                 128934
2    George          24782632786    Proskury                86323232
3    John           78234367823    Unten Den Linden                 128935
4    Jane    Doe    J&J    24782632786    Baker    26       65442    Stockholm    222333111
```

Group Questions Check Result

This structure encapsulates results of checking validity of supplied Group Query Input.

Type Attributes

Name	Type	Constraints	Mandatory	Description
matchingTemplates	Collection of String	Items are names of Matching Templates	yes	Defines set of matching templates which could be used to process parsed input
totalItemsCount	Integer		yes	Total number of questions detected within supplied input
validItemsCount	Integer		yes	Number of items which we will be able to use as valid question in group search machinery
totalErrorCount	Integer		yes	Number of items with errors detected within your input
parseErrors	Collection of ParseError		yes	Holds mapping of items which have issues with format unrecognized entries to their index in input.

JSON Representation

```
{
 "matchingTemplates": ["<String>"],
    "totalItemsCount": Number,
 "validItemsCount": Number,
    "totalErrorCount": Number,
 "parseErrors": [ {index: Number, value: "String"}, {index: Number, value: "String"}, ... ]
 }
```

Example of parse report

```
{
 "matchingTemplates": ["1130", "1140"],
    "totalItemsCount": 22,
 "validItemsCount": 10,
    "totalErrorCount": 12,
 "parseErrors": [ {index: 2, value: "abra"}, {index: 13, value: "This is the END!"}]
}
```

Group Query Parse error report

This structure provides all the errors detected during the parse of supplied user input.

Name	Type	Constraint	Mandatory	Description
Line number	Integer	Starts with 1	yes	Number of line in original user import which we have failed to parse
Line	String		yes	String from original input we have failed to parse

Example of parse error report. CSV representation

```
5 Abra
25 Cadabra} sasa
```

Matching Template

Matching template is an entity which describes how your input is going to be matched against an information we have in our database. In general it defines which attributes are going to be compared and in which way. In other words matching template describes a set of rules which should be performed on input row and output row to consider one matches another one. So if a row matches by all rules defined within a matching template it gets considered a matching row by the system in case if matching template was applied.

Type Attributes

Name	Type	Constraints	Mandatory	Description
id	String	Unique	yes	ID of matching template. Please note that standard matching templates have predefined IDs listed in this section.
name	String	Not empty	yes	Defines a name of matching templates. For standard templates it is the same as ID.
rules	List of Matching Rules	Not empty	yes	Holds collection of matching rules which should be applied to input in scope of current template

Example of Matching Template

```
[ {
 "id":"1100",
 "name": "1100",
    "rules" : [ {
    "field" : "firstName",
  "matchingKinds" : [ "EXACT", "ALIAS", "PHONETIC" ]
 }, {
  "field" : "lastName",
  "matchingKinds" : [ "EXACT", "ALIAS", "PHONETIC" ]
 }, {
    "field" : "regNum",
    "matchingKinds" : [ "EXACT" ]
 }, {
    "field" : "street",
    "matchingKinds" : [ "EXACT", "PHONETIC" ]
 }, {
    "field" : "streetNumber",
    "matchingKinds" : [ "EXACT" ]
 }, {
    "field" : "county",
  "matchingKinds" : [ "EXACT", "PHONETIC" ]
 } ],
  }]
```

Matching Rule

This is not quite self-standing entity and is only used in context of Matching Template. The rule defines a field which should be matched and a matching way.

Type Attributes

Name	Type	Constraints	Mandatory	Description
field	Enum String	One of: firstName,lastName,orgName,birthDate,regNum,street,houseNum,houseNumSuffix,zipCode,location,phone	yes	Defines a name of field in your input which should the matching rule to be applied to.
matchingKinds	List of Enum String	One of: "EXACT", "ALIAS", "PHONETIC"	yes	Defines a matching kind of which should be applied to a name defined witih the rule. Refer this section to get more details on matching kinds.

Please note, that not all the values of matching kind could be mapped to a particular field coming out of common sense in the first place. Ex. We cannot apply Alias match to Registration numbers and there is no other treating of given reg number than its exact or birth date match. Table bellow describe a matching of field name to possible values which could be defines within a matchingKind attribute. It also gives some explanations on fields names we support.

Field Name	Possible Matching Kinds	Applicable for	Comment
firstName	EXACT, ALIAS, PHONETIC	Persons	Match by First name of person. Can potentially hold first and last name of person. Matching engine will automatically detect a kind of situation and use values as first and last name
lastName	EXACT, ALIAS, PHONETIC	Persons	Match by Last name of person. Can potentially hold first and last name of person. Matching engine will automatically detect a kind of situation and use values as first and last name
orgName	EXACT, PHONETIC	Companies	Match by Company name.
regNum	EXACT	Persons, Companies	For persons - match by SSN value. For Companies - match by their Organization number.
birthDate	EXACT	Persons, Single Person Companies	Match by Birth date of the person
street	EXACT, ALIAS, PHONETIC	Persons, Companies	Match by Name of street .
houseNum	EXACT	Persons, Companies	Match by number of house on the street.
houseNumSuffix	EXACT	Persons, Companies	Match By Suffix of house number.

flat	EXACT	Persons, Companies	Match by Flat number
zipCode	EXACT	Persons, Companies	Match By Postal City to which the address relates.
location	EXACT, ALIAS, PHONETIC	Persons, Companies	Match By location
phone	EXACT	Persons, Companies	Match By Phone number.

Feature

Feature is a special processing kind available for your user. If you have special agreements with Itesco AB, you can have some specific features which you can use in group search which are out of standard scope of processing settings available out of the box. Features could be related to either input or output processing now. Input transforming features can, ex. adapt your input to format acceptable by the system and automatically start processing of it, or do some additional filtering/clean up to increase the volume of answers or reduce and made them more precise. Output features can ex. extend your answers with information which we do not supply as a part of standard delivery (ex. add phone operator name to each phone). So what is available to you depends only on which agreements you have with Itesco. The object returned by your features dictionary endpoint describe a feature available to you. At the moment there are supported only plain features which you cannot parametrize on template definition stage, but this could be changed in the future releases. The most important information you extract from the feature definition is it's NAME which you should put to features set if you would like to enable feature in your group query.

Type Attributes

Name	Type	Constraints	Mandatory	Description
id	String	unique	yes	ID of feature under which you refer it within your Settings. Please note that standard features have predefined IDs which you can find listed here
type	Enum String	one of • PRE_PROCESSOR • POST_PROCESSOR	yes	Defines a type of the feature. Type could be one of • PRE_PROCESSOR - prepares given input for further processing • POST_PROCESSOR - processes calculated result before providing them to user
name	String		yes	Name of the feature.
description	String		yes	Description of feature in human readable form of what the feature is and what it does

Exclusions

This type represents set of exclusions which should be used while querying the database for matching persons. Basically this is collection of strings.

Private Persons Exclusions

Private Persons Exclusions string collection can hold:

- 12-digit Swedish social security numbers, on the format NNNNNNNN-NNNN where N is a numeric character
- 10-digit Swedish social security numbers, on the format NNNNNN-NNNN where N is a numeric character
- Dwelling ID:s (40-characters long)
- Phone numbers should start with 0046 OR 0 OR +46 followed by 8-9 digits. May hold hyphens or spaces which will be removed before checking for format validity.

Example in text/csv

```
+46706219293
+46 70 5689 124
+46 70-5689-125
0706219296
0046706219293
f0524095ad56464bb4ffe03033b44bce5a471663
19670311-5623
670311-5624
```

Company Exclusions

Company Exclusions String collection can hold:

- company organization numbers in 10/12 digits format

Address Criteria List

The type represent set of addresses which the query result should be limited to. Basically this is collection of items of following type:

Name	Type	Constraints	Mandatory	Description
street	String		yes	Name of street
houseNum	Integer		no	Number of house
houseNumSuffix	String		no	Suffix addition to house number
flat	String		no	Number or caption of flat
zipCode	String	DDDDD	no	Zip code of address
zipCity	String		no	Name of Postal City

Please note, that current implementation is using TABs as separators.

Example in text/csv

```
Brovallsvägen 12  18494    åkersberga
Brovallsvägen 12 a 18494    åkersberga
Brovallsvägen 12 a 123 18494    åkersberga
Brovallsvägen   18494    åkersberga
Brovallsvägen    åkersberga
```

Exclusions Check Result

Private Persons Exclusions Check Result

This structure encapsulates results of checking validity of supplied Private Persons Exclusions.

Type Attributes

Name	Type	Constraints	Mandatory	Description
phonesCount	Integer		yes	Count of entries recognized as phone numbers in input.
ssnCount	integer		yes	Count of entries recognized as SSN's in input.
dwellingIdCount	Integer		yes	Count of entries recognized as dwelling IDs in input.
parseErrors	Collection of ParseError		yes	Holds mapping of unrecognized entries to their index in input.
totalErrorCount	Integer		yes	Total number of errors detected during parsing exclusions

JSON Representation

```
{
 "phonesCount": Number,
 "ssnCount": Number,
 "dwellingIdCount": Number,
 "parseErrors": [ {index: Number, value: "String"}, {index: Number, value: "String"}, ... ],
    "totalErrorCount": Number
}
```

Following example shows parse result with 10 phones, 1 dwelling ID and no SSNs recognized. Also there were error entries in lines 5 and 400

```
{
 "phonesCount": 10,
 "ssnCount": 0,
 "dwellingIdCount": 1,
 "parseErrors": [ {index: 5, value: "Abra"}, {index: 400, value: "Cadabra"}],
 "totalErrorCount" : 2
}
```

Private Persons Exclusions Parse Error report

This structure provides all the errors detected during the parse of supplied exclusions

Type Attributes

Name	Type	Constraints	Mandatory	Description
Line number	Integer	Starts with 1	yes	Number of line in original user import which we have failed to parse
Line	String		yes	String from original input we have failed to parse

CSV Representation

```
5 Abra
400 Cadabra
```

Company Exclusions Check Result

This structure encapsulates results of checking validity of supplied Company Exclusions.

Name	Type	Constraints	Mandatory	Description
orgNoCount	int		yes	Company organization number

JSON Representation

```
{
 "orgNoCount": Number,
 "parseErrors": [ {index: Number, value: "String"}, {index: Number, value: "String"}, ... ],
    "totalErrorCount": Number
}
```

Following example shows parse result with 10 phones, 1 dwelling ID and no SSNs recognized. Also there were error entries in lines 5 and 400

```
{
 "orgNoCount": 10,
 "parseErrors": [ {index: 5, value: "Abra"}, {index: 400, value: "Cadabra"}],
 "totalErrorCount" : 2
}
```

Company Exclusions Parse Error report

This structure provides all the errors detected during the parse of supplied exclusions.

Type Attributes

Name	Type	Constraints	Mandatory	Description
Line number	Integer	Starts with 1	yes	Number of line in original user import which we have failed to parse
Line	String		yes	String from original input we have failed to parse

CSV Representation

```
5 Abra
400 Cadabra
```

Address Criteria List Check Result

This structure encapsulates results of checking validity of supplied Address Criteria List.

Type Attributes

Name	Type	Constraints	Mandatory	Description
validItemsCount	Integer		yes	Count of entries in correct format and have valid content (adresses really exist).
invalidItems	Collection of ParseError		yes	Entries defined in correct format, but with invalid content (addresses have not been found)
parseErrors	Collection of ParseError		yes	Holds mapping of unrecognized entries to their index in input.
totalErrorCount	Integer		yes	Total number of parse errors detected
invalidItemsCount	Integer		yes	Total number of invalid items detected (Correct format but referring non existing geo items)

JSON Representation

```
{
 "validItemsCount": Number,
 "invalidItems": [ {index: Number, value: "String"}, {index: Number, value: "String"}, ... ],
 "parseErrors": [ {index: Number, value: "String"}, {index: Number, value: "String"}, ... ],
    "totalErrorCount" : Number,
    "invalidItemsCount" : Number
}
```

Following example shows parse result with 10 valid items. Also there were error entries and invalid ones

```
{
 "validItemsCount": Number,
 "invalidItems": [ {index: 10, value: "invalidItem"}],
 "parseErrors": [ {index: 5, value: "Abra-cadabra"}, {index: 6, value: "this is something unparsable"} ],
    "totalErrorCount" : 2,
    "invalidItemsCount" : 1
}
```

Addres criteria list parse error report

This structure encapsulates all errors got while parsing supplied Address Criteria List.

Name	Type	Constraints	Mandatory	Description
Line number	Integer	Starts with 1	yes	Number of line in original user import which we have failed to parse
Line	String		yes	String from original input we have failed to parse

CSV Representation

```
5 Abra-cadabra
6 this is something unparsable
```

Person Query Info

This object provides additional information about query allowing efficiently representing information about several resources managed by the system. At the moment it is no more than union of Query Definition, Private Prospects Query Result and Private Prospects Query Settings. Please note, that if some query is still performing or it is failed, naturally it is not worth to expect some result be specified within the QueryInfo object describing it.

JSON Representation

```
[{
 "@type" : "PersonQuery",
    "query": { <QueryDefintion1>},
 "result": {<QueryResult1>},
    "userInfo": {<UserInfo1>},
 "settings": {<QuerySettings1>}
 },
{
 "@type" : "PersonQuery",
    "query": { <QueryDefintion2>},
 "result": {<QueryResult2>},
 "userInfo": {<UserInfo2>},
 "settings": {<QuerySettings2>}
},
{
 "@type" : "PersonQuery",
    "query": { <IncompleteQueryDefinition>},
 "settings": {<IncompleteQuerySettings>}
},
{
 "@type" : "PersonQuery",
    "query": { <FailedQueryDefinition>},
    "settings": {<FailedQuerySettings>}
 },
...
]
```

Company Query Info

This object provides additional information about query allowing efficiently representing information about several resources managed by the system. At the moment it is no more than union of Query Definition, Company Prospects Query Result and Company Prospects Query Settings. Please note, that if some query is still performing or it is failed, naturally it is not worth to expect some result be specified within the QueryInfo object describing it.

JSON Representation

```
[{
 "@type" : "CompanyQuery",
    "query": { <QueryDefintion1>},
 "result": {<QueryResult1>},
 "settings": {<QuerySettings1>}
 },
{
 "@type" : "CompanyQuery",
    "query": { <QueryDefintion2>},
 "result": {<QueryResult2>},
 "settings": {<QuerySettings2>}
},
{
 "@type" : "CompanyQuery",
    "query": { <IncompleteQueryDefinition>},
 "settings": {<IncompleteQuerySettings>}
},
{
 "@type" : "CompanyQuery",
    "query": { <FailedQueryDefinition>},
    "settings": {<FailedQuerySettings>}
 },
...
]
```

Group Query Info

This object provides additional information about query allowing efficiently representing information about several resources managed by the system. At the moment it is no more than union of Group Query Definition, Group Query Result and Group Search Settings. Please note, that if some query is still performing or it is failed, naturally it is not worth to expect some result be specified within the QueryInfo object describing it.

JSON Reperesentation

```
[{
    "@type" : "GroupQuery",
    "query": { <QueryDefintion1>},
    "result": {<QueryResult1>},
    "settings": {<QuerySettings1>}
 },
{
    "@type" : "GroupQuery",
    "query": { <QueryDefintion2>},
    "result": {<QueryResult2>},
    "settings": {<QuerySettings2>}
},
{
    "@type" : "GroupQuery",
    "query": { <IncompleteQueryDefinition>},
    "settings": {<IncompleteQuerySettings>}
},
{
    "@type" : "GroupQuery",
    "query": { <FailedQueryDefinition>},
    "settings": {<FailedQuerySettings>}
 },
...
]
```

NIX Query Info

This object provides additional information about query allowing efficiently representing information about several resources managed by the system. At the moment it is no more than union of NIX Query Definition, NIX Query Result and NIX Query Settings. Please note, that if some query is still performing or it is failed, naturally it is not worth to expect some result be specified within the QueryInfo object describing it.

JSON Reperesentation

```
[{
    "@type" : "NixPhoneQuery",
    "query": { <QueryDefintion1>},
    "result": {<QueryResult1>},
    "settings": {<QuerySettings1>}
 },
{
    "@type" : "NixPhoneQuery",
    "query": { <QueryDefintion2>},
    "result": {<QueryResult2>},
    "settings": {<QuerySettings2>}
},
{
    "@type" : "NixPhoneQuery",
    "query": { <IncompleteQueryDefinition>},
    "settings": {<IncompleteQuerySettings>}
},
{
    "@type" : "NixPhoneQuery",
    "query": { <FailedQueryDefinition>},
    "settings": {<FailedQuerySettings>}
 },
...
]
```

Query Info

This object provides additional information about query allowing efficiently representing information about several resources managed by the system. At the moment it is union of Query Definition, Query Result, Query settings and information about User who has issued query. Please note, that if some query is still performing or it is failed, naturally it is not worth to expect some result be specified within the QueryInfo object describing it.

Query info objects additionally hold information about type of query being returned using service field '@type'. This field could have one of following values:

- PersonQuery - for private persons queries
- CompanyQuery - for company queries
- GroupQuery - for group searches
- NixPhoneQuery - for nix phone searches

JSON Representation

```
[{
 "@type" : "<QueryType1>",
 "query": { <QueryDefintion1>},
 "result": {<QueryResult1>},
 "settings": {<QuerySettings1>},
 "userInfo": {<UserInfo1>},
},
{
 "@type" : "<QueryType2>",
 "query": { <QueryDefintion2>},
 "result": {<QueryResult2>},
 "settings": {<QuerySettings2>},
 "userInfo": {<UserInfo2>}
 },
{
 "@type" : "<QueryType3>",
 "query": { <IncompleteQueryDefinition>},
 "settings": {<IncompleteQuerySettings>},
 "userInfo": {<IncomplateQueryUserInfo>}
 },
{
 "@type" : "<QueryType4>",
 "query": { <FailedQueryDefinition>},
 "settings": {<FailedQuerySettings>},
 "userInfo": {<FailedQueryUserInfo>}
 },
...
]
```

Copy Action Parameters

This structure is used to transfer parameters of copy operation for templates

Type Attributes

Name	Type	Constraints	Mandatory	Description
newName	String	Same as on Prospects Setting.name	no	New name which should be given to copy going to be created. If not defined new name will be generated as <originalName>-copy-YYYYMMDDhhmmss

File Info

The structure describes data uploaded to user's blob data storage which allows accessing uploaded blob data blocks in form of files.

Type Attributes

Name	Type	Constraints	Mandatory	Description
id	String		yes	File ID. Do not change if you want to get access to file
versionCode	String		yes (for PUT and GET)	ID of version. This ID is used by the system in order to determine conflicts (see Handling conflicts).

createdAt	Date	Read Only	yes (when getting)	Date of when the file was initially created. You cannot modify the value of this field.
modifiedAt	Date	Read Only	yes (when getting)	Date of when the setting was last time modified. You cannot modify the value of this field.
ownerId	String	ReadOnly	yes (when getting)	Id ow owner who the file belongs to
temporary	Boolean	true/false	yes (when getting)	Defines whether file is temporary or not
size	Number		yes (when getting)	Size in bytes of your file on storage

JSON Representation

```
{
 "id": "String",
 "versionCode" : "String",
    "createdAt" : <Date_in_format ISO-8601>,
    "modifiedAt" : <Date_in_format ISO-8601>,
    "ownerId: "String"
    "temporary" : "Boolean",
 "size" : <Number>
 }
```

Example for GET

```
{
 "id": "064ed556-c9fd-102b-9d1c-0030487dabd6",
 "versionCode": "064edf38-c9fd-102b-9d2c-0030487dabd6",
 "createdAt": "2015-02-215T16:30:44Z",
    "modifiedAt": "2015-02-215T16:30:46Z",
    "ownerId": "064cdf38-c9fd-102b-9d2c-0030487dabd6",
    "status": "SUCCESS",
 "answerCount": 133,
 "hidden": "false",
 "considerInExcludes": "true"
}
```

Zip Info

The structure which holds information about zip code and other attributes of a geographical area

Type Attributes

Name	Type	Constraints	Mandatory	Description
zip	String	Only digits	yes	Holds value of the zip code assosiated to represent geographical area
countyName	String	Only letters	yes	Holds the name of county which the geographical area belongs to
municipality	String	Only letters	yes	Holds the name municipality which the geographical area belongs to

JSON Representation

```
{
    zip: "String_with_only_digits",
    countyName: "String_with_only_letters",
    municipalityName : "String_with_only_letters"
}
```

Example for GET

```
{
    zip: "10005",
    countyName: "NORRTÄLJE",
    municipalityName : "STOCKHOLM"
}
```

County Info

The structure which holds information about county and its municipalities

Type Attributes

Name	Type	Constraints	Mandatory	Description
name	String	Only letters	yes	Name of county which the structure describes
municipalities	String list	Items consist of letters only	yes	Set of counties which relate to the given municipality

JSON Representation

```
{
    "name": "String_with_only_letters",
    "municipalities": ["MUNICIPALITY_1_NAME", .., "MUNICIPALITY_N_NAME"]
}
```

Example for GET

```
{
    "name": "BLEKINGE",
    "municipalities": [
        "KARLSHAMN",
        "KARLSKRONA",
        "OLOFSTRÖM",
        "RONNEBY",
        "SÖLVESBORG"
    ]
}
```

Zip City by Municipality Info

The structure which holds information about municipality and its zipCities

Type Attributes

Name	Type	Constraints	Mandatory	Description
municipality	String	Only letters	yes	Name of county which the structure describes
cities	String list	Items consist of letters only	yes	Set of counties which relate to the given municipality

JSON Representation

```
{
    "municipality": "String_with_only_letters",
    "cities": ["CITY_1_NAME", .., "CITY_N_NAME"]
}
```

Example for GET

```
{
    "municipality": "ANEBY",
    "cities": [
        "ANEBY",
        "FRINNARYD",
        "LEKERYD",
        "SUNHULTSBRUNN"
    ]
}
```

SNI Code Info

The structure which holds info about SNI code got from dictionary

JSON Representation

```
{
    "code": "String_value_of_SNI_code",
    "description": "String_description_of_business_area"
}
```

Example for GET

```
{
    "code": "01120",
    "description": "Growing of rice"
}
```

Operations

Within this section you should find the description of operations you can perform using API. Description of each operation holds definition of input/output, supported data types and possible error codes which could be returned in the particular case.

Checking API availability

For means of checking accessibility there is provided special endpoint. There is no requirement for being authorized to access it. The main useful feature of the endpoint is supplying you with the API version info you are communicating to. API version info is packaged as Version Info object.

> **Important**
> Ususally you can expect some API changes in case if your application detects at least 'minorVersion' field update. For getting information on what exactly has changed in release please refer Change Log section of present document. Also you can always access the most actual version of API documentation by following URL https://icatch3.icatch.se/assets/files/iCatch3_API_Specification.pdf.

Syntax

GET /ping

Request Format

Content-Type: application/json

Accept: application/json

Accept-Charset: utf-8

Response format

Content-Type: application/json

X-Client-IP: IP address which server considers to be the client one

Body: Version Info

Response codes

- Status 200 - OK. Result is in response body.

Authorization

Logging into the system

In order to get access to features which requires authorization you should authorize yourself. You can find more details on Authorization in corresponding section of the specification.

Syntax

POST /login

Request Format

Content-Type: application/json

Accept: application/json

Accept-Charset: utf-8

Body: Credentials

Response format

Content-Type: application/json

Body: Authentication Result

Response codes

- Status 200 - OK. Result is in response body.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section)
- Status 401 - Authorization failure
  - 401002 - given combination of user name and password is not recognized

Logging out

Gets the authentication token from the header and invalidates it. Further usage of token will result Responses with status 401 and sub-code 40102.

Syntax

POST /logout

Request Format

X-Auth-Token: <your_token>

Accept: application/json

Accept-Charset: utf-8

Content-Type: application/json

No body is expected.

Response format

Content type: application/json (in case if this is error response)

Response codes

- Status 204 - OK. Successfully logged you out.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section)
- Status 401 - Authorization failure (sub-codes are described in Common Error Codes section)

Quick Search

As an authorized user you can issue direct search requests to our database and instantly get results in scope of single request/response cycle. This implies just issuing plain GET request and configure selection properties using Request HTTP parameters. In response you are getting results encoded in a body in form of JSON array. It consists of Quick Search Result items.

So, basically you define here your filtering logic by composing them into necessary set of Request parameters. Start index from which you would like to get results and number of results you want is also defined within criteria you supply. This is usually enough for organizing paged extraction of results.

We are using standard Content-Range header syntax to report you about count of records available. So, ex. If you issue your request with count = 10 and no startIndex, you'll get a first page of defined size (10 in example) with header Content-Range: 0-10/33 if we have totally 33 matching results. More details on paged getting of results you can find in Paged reading section.

Syntax

GET /quickSearch?<parameters>

Request Format

X-Auth-Token: <your_token>

Accept: application/json

Accept-Charset: utf-8

Quick Search Request Parameters are supported

Response format

Content type: application/json

Body: Array of Quick Search Result items

Response codes

- Status 200 - OK. Result is in response body.
- Status 201 - Empty. There are no results matching you requests.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section)
- Status 401 - Authorization failure (sub-codes are described in Common Error Codes section)

Quick Search Request Parameters

Complete set of parameters is listed in table below. Please note, that no parameters of ones mentioned below are mandatory, you can just omit them in JSON you send. For some values, when not defined the actual value is taken by default. For other omitting means that you just don't apply the logic parameter is responsible for.

Please note, that Parameters you define basically follow the URL encode rules.

Boolean values could be defined as 'true'/'false' literals and also as 1/0.

Arrays could be just given as comma separated value list after parameter (as in example) but there are alternative ways. All options are:

- GET /quickSearch?searchAmong=PERSONS,COMPANIES
- GET /quickSearch?searchAmong=PERSONS&searchAmong=COMPANIES
- GET /quickSearch?searchAmong[0]=PERSONS&searchAmong[1]=COMPANIES

The complete list of available parameters is listed below:

Name	Type	Constraints	Default	Description
Paging				
startIndex	Integer	>=0, < total available count	0	Value in range [0..totalItemsCount). Defines start index of answer being returned. 0 if not defined.
count	Integer	>0	2000	Value in range [1..totalItemsCount- startIndex]. Defines how many answers should be returned considering defined startIndex. 2 000 if not defined
Filtering criteria				
name	String		N/A	One or several names of the person. Considered to hold complete person name including the last name if **lastName** is not defined. If holds comma, then everything before comma considered to be a last name
lastName	String		N/A	Last name of a person
orgName	String	Only for companies	N/A	Name of organization.
regNum	String		N/A	Can hold either SSN of person or Organization number of company
birthdate	Date	Formats: YYYY-MM-DD, YYYYMMDD Only for persons	N/A	Birth date of the person.
address	String		N/A	Free form address. Considered only if **street**, **houseNum**, **houseNumSuffix** are not given. If you would like to have better matches, consider using particular fields for defining parts of address.
street	String		N/A	Street name
houseNum	String		N/A	House number
houseNumSuffix	String		N/A	Suffix of house number. Can hold flat number. Flat number could be either split by one of standard abbreviations like LGH, L, LAG, LGHT, LH, etc... or just be a valid flat number coming after house suffix. In this case System will automatically split such declaration into 2 fields (house number suffix and flat) while making matching.
flat	String		N/A	Flat number

city	String		N/A	Address city
zipCode	Int	5 digits > 0	N/A	Zip code of address
dwellingId	Int	Only for persons	N/A	Special aggregation ID which is assigned by Itesco for dwellings the persons inhabit. You can get it from result item and use for getting cohabitants of the person
phone	String	Formats: +46XXXXXXXXX, 46XXXXXXXXX, 0046XXXXXXXXX, 0XXXXXXXXX, 0XX-XXXXXXXX, 0X/XXXXXXXX	N/A	Phone number of person/company. This could be given in one of formats supported. Preferably
sniCodes	Array	of Strings	[]	Array of strings where each item represent SNI code.
Search parameters				
exact	Bool	true/false	false	If set to true, then result is returned matching exactly to values you have defined as filtering parameters
searchAmong	Array	of PERSONS, COMPANIES	[]	If defined only records matching party types mentioned in array are returned. Else any are returned.
phoneKinds	Array	of FIXED, MOBILE	[]	If defined only phones matching given ones are returned in scope of result items. Parties without phones are not returned.
cohabitants	Bool	true/false only for persons	false	If set to true, then besides persons matching the criteria supplied there will be returned ones inhabiting the same dwelling. Note, that this parameter has sense only for private persons.

Here are several samples of requests you can issue:

Sample for Company (by name only fixed phones, exact match):

```
GET /quickSearch?
    startIndex=10
    &count=10
    &name=Svenska%20Company
    &sniCodes=01,02
    &phoneKinds=FIXED
    &searchAmong=COMPANIES
    &exact=1
```

Sample for Person (by name only mobiles, allow inexact matching, with cohabitants):

```
GET /quickSearch?
    startIndex=10
    &count=10
    &name=Ingvar%20Karlsson
    &birthdate=1984-09-04
    &phoneKinds=MOBILE
    &searchAmong=PERSONS
```

Sample for any (by address, allow inexact, any kind of answer and phone)

```
GET /quickSearch?
    street=Somevagen%20138%20A%20LGH%201011
    &city=Stockholm
```

Private Prospects

Creating new Private Persons Prospects settings

As an authorized user you can create settings templates for further usage with private person prospects. In result of request you will be supplied with created template settings parameters.

> **Encodings**
> Please note that we do our best to detect encoding of content you have supplied in body (for exclusions/addresses parts of request), but if we fail - there will be used ISO-8859-1. So if you see some issues with parsing or getting corrupt output when requesting uploaded results, please let us know and consider using ISO-8859-1 encoding for your input till we are working on the solution of the problem.

Syntax

POST /privateProspects/settingsTemplate

POST /privateProspects/settingsTemplate[?exclusionsFileId={fileId}][&addressFilterFileId={fileId}]

POST /privateProspects/settingsTemplate (multipart, parts: setting, exclusions, addresFilter)

Request Format

Parameters:

- exclusionsFileId - ID of pre-loaded file (see Uploading input files) holding exclusions. Non Mandatory. If not defined template related exclusions are used.
- addressFilterFileId - ID of pre-loaded file (see Uploading input files) holding address filter. Non Mandatory.

X-Auth-Token: <your_token>

Content-Type: application/json

Accept: application/json

Accept-Charset: utf-8

Body: Private Prospects Setting

You should not supply ID as the ID will be generated for you once setting is created. Supplied ID will be ignored If you are going to supply exclusions or address filter you need to issue multipart request. Also **createdAt** and **modifiedAt** fields values will be ignored even if supplied

Please note, when posting as multipart it is expected one mandatory part called **setting** which holds your settings values in json and filters (called **exclusions** and **addresFilter**) none of which are mandatory. Also you could post with file IDs of pre-loaded files.

Response format

Content-Type: application/json

Body: Private Prospects Setting

Only settings in json format will be returned. For accessing exclusions or address filter associated with your setting please use separate methods:

- Getting Stored Private Prospects Exclusions
- Getting Stored Private Prospects Address Filter

Response codes

- Status 200 - OK. Result is in response body.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section)
- Status 401 - Authorization failure (sub-codes are described in Common Error Codes section)

Updating Private Persons Prospects settings

As an authorized user you can update settings templates which were previously stored. In return you should get actual state of settings you have been modified.

Syntax

PUT /privateProspects/settingsTemplate/{id}

PUT /privateProspects/settingsTemplate/{id}[?exclusionsFileId={fileId}][&addressFilterFileId={fileId}]

Request Format

Parameters:

- exclusionsFileId - ID of pre-loaded file (see Uploading input files) holding exclusions. Non Mandatory. If not defined template related exclusions are used. Posting empty exclusions will not remove associated exclusions. Use DELETE endpoints in order to get rid of them.
- addressFilterFileId - ID of pre-loaded file (see Uploading input files) holding address filter. Non Mandatory. Posting empty address filter will not remove associated address filter items. Use DELETE endpoints in order to get rid of them.

X-Auth-Token: <your_token>

Content-Type: application/json

Accept: application/json

Accept-Charset: utf-8

Body: Private Prospects Setting

If you supply ID different from one used in URL, you'll get error. This is done to prevent from occasional modification of your settings. At the same time if you did not supplied ID within the body - error will not be issued.

Please note that this endpoint does not support multipart requests due to HTTP semantics

Response format

Content type: application/json

Body: Private Prospects Setting

Only settings template in json format will be returned. For accessing exclusions or address filter associated with your setting please use separate methods:

- Getting Stored Private Prospects Exclusions
- Getting Stored Private Prospects Address Filter

Response codes

- Status 200 - OK. Result is in response body.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section)
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if settings with given ID is not present.
- Status 409 - In case of Conflict (see Handling conflicts)

Copying Private Persons Prospects settings

As an authorized user you can create a copy of your settings templates for further usage with private person prospects. In result of request you will be supplied with created copy of template template settings parameters.

Syntax

POST /privateProspects/settingsTemplate/{id}/copy

Request Format

X-Auth-Token: <your_token>

Content-Type: application/json

Accept: application/json

Accept-Charset: utf-8

Body: Copy Action Parameters

You should supply ID of original entity in URL. In case if this entity does not exists or does not belongs to user issuing request there will be returned 404 status

Response format

Content-Type: application/json

Body: Private Prospects Setting

Only settings in json format will be returned. For accessing exclusions or address filter associated with your setting please use separate methods:

- Getting Stored Private Prospects Exclusions
- Getting Stored Private Prospects Address Filter

Response codes

- Status 200 - OK. Result is in response body.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section)
- Status 401 - Authorization failure (sub-codes are described in Common Error Codes section)
- Status 404 - Entity with deined id is either not exists or does not belongs to user issued request

Deleting Private Persons Prospects settings

As an authorized user you can delete previously stored settings template. Result of operation is determined by returned status code. Please note, that while dropping the settings template you are also dropping such settings attributes like Private Persons Exclusions and Address Criteria List. After issuing request you'll be getting 404 status responses for following operations:

- DELETE /privateProspects/settingsTemplate/{id}
- DELETE /privateProspects/settingsTemplate/{id}/exclusions
- DELETE /privateProspects/settingsTemplate/{id}/allowedAddresses

Syntax

DELETE /privateProspects/settingsTemplate/{id}

Request Format

X-Auth-Token: <your_token>

Accept: application/json

Accept-Charset: utf-8

Body: <no_body_expected>

Response format

Content-Type: application/json

In case of error content type could be application/json with error in body

Body: <no_body_expected>

Response codes

- Status 204 - OK. Setting was located and successfully deleted.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section).
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if settings with given ID does not exist or was already deleted.

Getting information about your Private Persons Prospects settings

As an authorized user you can get information about Private Persons Prospects settings associated with your user. You can get all your settings, or filter by ID or name. The single resource is responsible for the querying. It supports the set of parameters which could be specified in order to make your search more precise.

Syntax

GET /privateProspects/settingsTemplate[?startIndex={startIndex}&count={count}&name={name_pattern}] - get all settings for user issuing request matching paramters

GET /privateProspects/settingsTemplate/{id} - get settings with given ID

Request Format

Parameters:

- startIndex, count (see Paged Reading)
- name - pattern of template name. Supports wildcards (* and ?)

X-Auth-Token: <your_token>

Accept: application/json

Accept-Charset: utf-8

Body: <no_body_expected>

Response format

Content-Type: application/json

Content-Range: <startIndex>-<endIndex>/<totalCount>

Body: Collection of Private Prospects Setting

Note that in return we will not issue multipart response. Only settings in json format will be returned. For accessing exclusions or address filter assocoiated with your setting please use separate methods:

- Getting Stored Exclusions
- Getting Stored Address Filter

Response codes

- Status 200 - OK. There is at least one result matching your request parameters in the body.
- Status 204 - Ok. There are no settings matching your request parameters.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section)
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if you have been issuing get by ID and settings with given ID does not exist.

Getting Stored Private Prospects Exclusions

As an authorized user you could submit exclusions which should be used when performing your query associated to the settings. Despite you can define exclusions list directly within your POST/PUT request body JSON part, system does not make any difference in exclusions supplied in such a way and supplied as the separate part of multipart request. You are always getting exclusions in form of application/octet-stream. Although format of response depends on value of **Accept** header you have supplied in request. Also you can request data in necessary encoding using 'Accept-charset' header or 'charset' parameter of 'Accept' header.

Syntax

GET /privateProspects/settingsTemplate/{id}/exclusions

Request Format

X-Auth-Token: <your_token>

Accept-Charset: <encoding> (ex. 'iso-8859-1', 'utf8'. UTF-8 is assumed if not specified)

Accept :text/csv OR application/json OR application/zip OR application/gzip ('charset' parameter value is used if there is no Accept-Charset header)

With this header you specify which output format you prefer to get in response. Please note that if you have requested zip/gzip content than values will be returned in CSV format zipped/gzipped

Response format

Content-Type: depends on Accept header value

Body: Stream in format defined by Accept header value. For format refer Private Persons Exclusions.

Response codes

- Status 200 - OK. There is at least one exclusion associated with your settings.
- Status 204 - Ok. There are no exclusions associated with your settings.
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if settings with given ID does not exist.

Updating Stored Private Prospects Exclusions

As an authorized user you can update exclusions which can be used with specific settings in future queries. If settings are already associated with some exclusions, it will be replaced with a new set of exclusions.

Syntax

POST /privateProspects/settingsTemplate/{id}/exclusions

POST /privateProspects/settingsTemplate/{id}/exclusions?fileId={preloaded_file_id}

Request Format

Parameters:

- one of the next respectively:
  - exclusionsFilterFile - file with allowed addresses (see Private Persons Exclusions)
  - storedFileId - ID of pre-loaded file (see Uploading input files) holding address filter.

X-Auth-Token: <your_token>

Content-Type: application/json

Accept: application/json

Accept-Charset: utf-8

Response format

Content-Type: application/json

Body: <no_body_expected>

Response codes

- Status 200 - OK. Exclusions are updated.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section)
- Status 401 - Authorization failure (sub-codes are described in Common Error Codes section)
- Status 404 - Entity with provided id is either does not exist or does not belong to user issued request

Deleting Stored Private Prospects Exclusions

As an authorized user you can delete exclusions associated with a template.

Syntax

DELETE /privateProspects/settingsTemplate/{id}/exclusions

Request Format

X-Auth-Token: <your_token>

Content-Type: application/json

Accept: application/json

Accept-Charset: utf-8

Response format

Content-Type: application/json

Body: <no_body_expected>

Response codes

- Status 204 - OK. Settings were located and exclusions were successfully deleted.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section).
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if settings with given ID does not exist or it doesn't have exclusions. Also error will be returned if exclusions were already deleted.

Getting Stored Private Prospects Address Filter

As an authorized user you could submit list of allowed addresses which should be used when performing your query associated to the settings. Despite you can define allowed addresses list directly within your POST/PUT request body JSON part, system does not make any difference in addresses supplied in such a way and supplied as the separate part of multipart request. You are always getting addresses in form of application/octet-stream. Although format of response depends on value of **Accept** header you have supplied in request. Also you can request data in necessary encoding using 'Accept-charset' header or 'charset' parameter of 'Accept' header.

Syntax

GET /privateProspects/settingsTemplate/{id}/allowedAddresses

Request Format

X-Auth-Token: <your_token>

Accept-Charset: <encoding> (ex. 'iso-8859-1', 'utf8'. UTF-8 is assumed if not specified)

Accept :text/csv OR application/json OR application/zip OR application/gzip ('charset' parameter value is used if there is no Accept-Charset header)

With this header you specify which output format you prefer to get in response. Please note that if you have requested zip/gzip content than values will be returned in CSV format zipped/gzipped

Response format

Content-Type: depends on 'Accept' header value

Body: Stream in format defined by Accept header value. For format refer Address Criteria List.

Response codes

- Status 200 - OK. There is at least one address Filter entry associated with your settings.
- Status 204 - Ok. There are no address Filter entries associated with your settings.
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if settings with given ID does not exist.

Updating Stored Private Prospects Address Filter

As an authorized user you can update a list of allowed addresses associated specific some settings.

Syntax

POST /privateProspects/settingsTemplate/{id}/allowedAddresses

POST /privateProspects/settingsTemplate/{id}/allowedAddresses?fileId={preloaded_file_id}

Request Format

Parameters:

- one of the next respectively:
  - addressFilterFile - file with allowed addresses (see Address Criteria List)
  - storedFileId - ID of pre-loaded file (see Uploading input files) holding address filter. Non Mandatory.

X-Auth-Token: <your_token>

Content-Type: application/json

Accept: application/json

Accept-Charset: utf-8

Response format

Content-Type: application/json

Body: <no_body_expected>

Response codes

- Status 200 - OK. Address filter is successfully updated.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section)
- Status 401 - Authorization failure (sub-codes are described in Common Error Codes section)
- Status 404 - Entity with provided id either does not exist or does not belong to user issued request

Deleting Stored Private Prospects Address Filter

As an authorized user you can delete previously stored list of allowed addresses. Result of operation is determined by returned status code. After issuing request you'll be getting 404 status responses for following operations:

- DELETE /privateProspects/settingsTemplate/{id}/allowedAddresses

Syntax

DELETE /privateProspects/settingsTemplate/{id}/allowedAddresses

Request Format

X-Auth-Token: <your_token>

Accept: application/json

Accept-Charset: utf-8

Body: <no_body_expected>

Response format

Content-Type: application/json

In case of error content type could be application/json with error in body

Body: <no_body_expected>

Response codes

- Status 204 - OK. List of addresses was located and successfully deleted.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section).
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if list of addresses with given ID does not exist or was already deleted.

Estimating number of answers for query

As an authorized user you can estimate how many answers you are going to get if you run a query with specified settings and additional overrides. For that you should POST your query parameters to the special resource. In response you should get the Dry Run Result which holds results of estimation. The estimation is preformed in scope of single request-response cycle (unlike when posting to /privateProspects/queries resource), so you should consider that performing could take some time. Please, be advised about that when setting up your client connection timeout values.

> **Encodings**
> Please note that we do our best to detect encoding of content you have supplied in body (for exclusions/addresses parts of request), but if we fail - there will be used ISO-8859-1. So if you see some issues with parsing or getting corrupt output when requesting uploaded results, please let us know and consider using ISO-8859-1 encoding for your input till we are working on the solution of the problem.

Syntax

POST /privateProspects/queries/dryRun

POST /privateProspects/queries/dryRun[?exclusionsFileId={fileId}][&addressFilterFileId={fileId}]

Request Format

Parameters:

- exclusionsFileId - ID of pre-loaded file (see Uploading input files) holding exclusions. Non Mandatory. If not defined template related exclusions are used. If defined then exclusions get merged with template exclusions for the run
- addressFilterFileId - ID of pre-loaded file (see Uploading input files) holding address filter. Non Mandatory. If not defined template related address filter is used. If defined then given address filter is used. Please note that supplied address filter has priority over GEO filter. So if your template has Geo filter defined than it will not be considered.

X-Auth-Token: <your_token>

Content-Type: multipart/form-data if you are uploading exclusions/address filters in body; application/json if you are referring pre-loaded files

Accept: application/json

Accept-Charset: utf-8

Body: Prospects Query

Response format

Content-Type: application/json

Body: Dry Run Result

Response codes

- Status 200 - OK. Results of estimation are in the body.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section)
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if settings with given ID is not present. In case if file with defined ID does not exists for exclusions/address filters

Posting a query for execution

As an authorized user you can post a query for execution. While posting the query you must supply the reference to settings which you would like to use. Additionally you can supply settings parameters which would override corresponding parameters within the settings.

Encodings
Please note that we do our best to detect encoding of content you have supplied in body (for exclusions/addresses parts of request), but if we fail - there will be used ISO-8859-1. So if you see some issues with parsing or getting corrupt output when requesting uploaded results, please let us know and consider using ISO-8859-1 encoding for your input till we are working on the solution of the problem.

Syntax

POST /privateProspects/queries

POST /privateProspects/queries[?exclusionsFileId={fileId}][&addressFilterFileId={fileId}]

Request Format

Parameters:

- exclusionsFileId - ID of pre-loaded file (see Uploading input files) holding exclusions. Non Mandatory. If not defined template related exclusions are used. If defined then exclusions get merged with template exclusions for the run
- addressFilterFileId - ID of pre-loaded file (see Uploading input files) holding address filter. Non Mandatory. If not defined template related address filter is used. If defined then given address filter is used. Please note that supplied address filter has priority over GEO filter. So if your template has Geo filter defined than it will not be considered.

X-Auth-Token: <your_token>

Content-Type: multipart/form-data if you are uploading exclusions/address filters in body; application/json if you are referring pre-loaded files

Accept: application/json

Accept-Charset: utf-8

Body: Prospects Query

Please note, that if you have supplied Address Criteria List this would replace ones defined in query settings associated. If you have supplied Private Persons Exclusions they will be merged and for exclusions and result is going to be used when calculating query result.

Response format

Content-Type: application/json

Body: Prospects Query

In response you will not get associated exclusions or allowed addresses. Supplied exclusions and addresses are associated with settings instance created automatically from your template and overrides (and mentioned in query object under settingsId field). They could be retrieved using separate resources:

- Getting query settings instance related exclusions
- Getting query settings instance related addresses limitations

Response codes

- Status 200 - OK. Posted query metadata is in the body. Metadata also contains assigned ID.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section).
- Status 404 - In case if file with defined ID does not exists for exclusions/address filters
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if settings with given ID is not present. In case if file with defined ID does not exists for exclusions/address filters

As an authorized user you can get information about Private Persons Prospects queries you have posted. You can get all your settings, or filter by ID or name. The single resource is responsible for the querying.

Syntax

GET /privateProspects/queries[?startIndex={startIndex}&count={count}] - get all queries for user has issued.

GET /privateProspects/queries/{id} - get query with given ID

Request Format

Parameters:

- startIndex, count (see Paged Reading)

X-Auth-Token: <your_token>

Accept: application/json

Accept-Charset: utf-8

Body: <no_body_expected>

Response format

Content-Type: application/json

Body: Collection of Prospects Query

Content-Range: <startIndex>-<endIndex>/<totalCount>

In response you will not get associated exclusions or allowed addresses. Supplied exclusions and addresses are associated with results and could be retrieved using separate resources:

- Getting Stored Private Prospects Exclusions
- Getting Stored Private Prospects Address Filter

Response codes

- Status 200 - OK. There is at least one result matching your request parameters in the body.
- Status 204 - OK. There are no settings matching your request parameters.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section)
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if you have been issuing get by ID and query with given ID does not exist.

Getting information about your particular person prospects query progress

As the part of response for query resource you are also getting information about the status of query execution. Please refer Getting information about your queries in order to find out how to get information about query. You can also refer Private Prospects Query object definition to get info about the structure which is used for holding query status info.

Getting information about your query results

As an authorized user you can get information about Private Persons Prospects Results. You can get information about all available results or use request parameters in order to make more precise selection. In scope of this request you are not getting actual query results, just metadata describing results being produced. In order to find out how to get access to the results you should refer

- Getting all your private prospects query results
- Getting your query results in chunks

Syntax

GET /privateProspects/results - get all results produced so far for your user

GET /privateProspects/results[?queryId={queryId}&startIndex={startIndex}&count={count}] - get all results produced so far for your user

GET /privateProspects/results/{id} - get results with given ID

Request Format

Parameters:

- startIndex, count (see Paged Reading)

X-Auth-Token: <your_token>

Accept: application/json

Accept-Charset: utf-8

Body: <no_body_expected>

Response format

Content-Type: application/json

Body: Collection of Private Prospects Query Result

Content-Range: <startIndex>-<endIndex>/<totalCount>

Response codes

- Status 200 - OK. There is at least one result matching your request parametersin the body.
- Status 204 - Ok. There are no results matching your request parameters.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section)
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if you have been issuing get by ID and result with given ID does not exist.

Getting all your private prospects query results

As an authorized user you can get results of query which has been completed successfully. Results are returned as text/csv. Also you can request data in necessary encoding using 'Accept-charset' header or 'charset' parameter of 'Accept' header (like "Accept: text/csv;charset=iso-8859-1").

Syntax

GET /privateProspects/results/{id}/valuesStream

Request Format

X-Auth-Token: <your_token>

Accept : text/csv OR application/json OR application/zip OR application/gzip ('charset' parameter value is used if there is no Accept-Charset header)

With this header you specify which output format you prefer to get in response. Please note that if you have requested zip/gzip content than values will be returned in CSV format zipped/gzipped

Accept-Charset: <encoding> (ex. 'iso-8859-1', 'utf8'. UTF-8 is assumed if not specified)

Response format

Content-Type: text/csv

Body: Stream in format defined by Accept header value. For format refer Private Prospects Query Result Value.

Response codes

- Status 200 - Ok. There is at least one answer produced for your query.
- Status 204 - Ok. There are no answers for your query.
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section).
- Status 404 - In case if results with given ID does not exist.
- Status 403 - In case if your result is locked.

Getting your query results in chunks

As an authorized user you can get results of query which has been completed successfully. Results are returned directly within the body. This way of getting results differs from on described in Getting all your private prospects query results because it allows you reducing output set. Using that method could be useful for organizing paged review of results produced in response to your query. You can also find useful using Private Prospects Query Result's **answerCount** property for paging needs. Also you can get results in different formats. Returned data format depends on "Accept" header value.

Syntax

GET /privateProspects/results/{id}/values[?startIndex={index_value}&count={count_value}]

Request Format

Parameters:

- startIndex, count (see Paged reading)

X-Auth-Token: <your_token>

Accept : text/csv or application/json

Accept-Charset: utf-8

Response format

Content-Type: text/csv or application/json (depending on accept header value)

Body: Collection of Private Prospects Query Result Value constrained according to parameters of request

Response codes

- Status 200 - Ok. There is at least one answer produced for your query.
- Status 204 - Ok. There are no answers for your query.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section).
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if results with given ID does not exist.
- Status 403 - In case if your result is locked.

Updating private prospects query results related attributes

Query results have a few parameters which affects consequent queries you issue. As an authorized user you are able to modify the attributes using API.

Syntax

PUT /privateProspects/results/{id}

Request Format

X-Auth-Token: <your_token>

Content-Type: application/json

Accept-Charset: utf-8

Accept: application/json

Body: Private Prospects Query Result

If you supply ID different from one used in URL, you'll get error. This is done to prevent from occasional modification of your settings. At the same time if you did not supplied ID within the body - error will not be issued.

Response format

Content-Type: application/json

Body: Private Prospects Query Result

Response codes

- Status 200 - OK. Result is in response body.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section)
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if result with given ID is not present.
- Status 409 - In case of Conflict (see Handling conflicts)

Getting Private Persons Exclusions used for query

As an authorized user you can get private persons exclusions used while running a particular query. Please note that this could be composition of persistent exclusions associated with setting/query and additional exclusions supplied with query. Only those rows which were recognized in input are present in result of this request. You are always getting exclusions in form of application/octet-stream. Although format of response depends on value of **Accept** header you have supplied in request. Also you can request data in necessary encoding using 'Accept-charset' header or 'charset' parameter of 'Accept' header.

Syntax

GET /privateProspects/settingsInstance/{id}/exclusions

Request Format

X-Auth-Token: <your_token>

Accept-Charset: <encoding> (ex. 'iso-8859-1', 'utf8'. UTF-8 is assumed if not specified)

Accept :text/csv OR application/json OR application/zip OR application/gzip ('charset' parameter value is used if there is no Accept-Charset header)

With this header you specify which output format you prefer to get in response. Please note that if you have requested zip/gzip content than values will be returned in CSV format zipped/gzipped

Response format

Content-Type: depends on Accept header value

Body: Stream in format defined by Accept header value. For format refer Private Persons Exclusions.

Response codes

- Status 200 - OK. There is at least one exclusion associated with your settings.
- Status 204 - Ok. There are no exclusions associated with your settings.
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if settings with given ID does not exist.

Checking correctness of supplied Private Persons Exclusions

As an authorized user you can check whether the exclusions you are going to provide are able to be considered by the system. For that you should post them to a special URL in Private Persons Exclusions format. In response you'll get Private Persons Exclusions Check Result/Private Persons Exclusions Parse Error report which contains parse errors and information about recognized entries. When getting exclusions parse report as text/csv you can define encoding in which report is going to be provided. Use 'Accept-Charset' header or 'charset' parameter of request for that.

Encodings
Please note that we do our best to detect encoding of content you have supplied in body, but if we fail - there will be used ISO-8859-1. So if you see some issues with parsing or getting corrupt output when requesting uploaded results, please let us know and consider using ISO-8859-1 encoding for your input till we are working on the solution of the problem.

Syntax

POST /privateProspects/exclusions[?maxErrorsCount={maxErrorsCount}]

GET /privateProspects/exclusionsParseReport?fileId={fileId}[&maxErrorsCount={maxErrorsCount}]

Request Format

Parameters:

- maxErrorsCount = maximum number of errors to put into response. Non mandatory. Total number of errors detected in supplied input is returned in response as separate field
- fileId - Allowed only for GET exclusionsParseReport endpoint. Defines ID of pre-loaded file (see Uploading input files) holding exclusions.

X-Auth-Token: <your_token>.

Accept-Charset: <encoding> (ex. 'iso-8859-1', 'utf8'. UTF-8 is assumed if not specified)

Content-Type: multipart/form-data, where part with text should have text/csv content type

Accept: application/json or text/csv

- In application/json you get up to maxErrorsCount error entries. If not defined you get up to maximum supported by the platform now (currently = 1000)
- In text/csv you get ONLY error items in format <errorLineNumber><TAB><rawValueTakenFromFile>. maxErrorsCount is not considered in that case.

Body: Private Persons Exclusions

Response format

Content-Type: application/json or text/csv (depending on Accept header value). Body is encoded in UTF-8

Body: Private Persons Exclusions Check Result/Private Persons Exclusions Parse Error report

Response codes

- Status 200 - OK. Parse report is n body
- Status 404 - In case if file with supplied ID could not be found (for GET request)
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).

Getting person query instance settings by query id

As an authorized user you can get settings instance used with a specific query.

Syntax

GET /privateProspects/queries/{id}/settings

Request Format

X-Auth-Token: <your_token>

Accept: application/json

Accept-Charset: utf-8

Body: <no_body_expected>

Response format

Content-Type: application/json

Body: Person Query Instance

Response codes

- Status 200 - OK. There is a result matching your request in the body.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section)

- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if you have been issuing get by ID and query with given ID does not exist.

Getting person query instance settings by its id

As an authorized user you can get settings instance by its id.

Syntax

GET /privateProspects/settingsInstance/{id}

Request Format

X-Auth-Token: <your_token>

Accept: application/json

Accept-Charset: utf-8

Body: <no_body_expected>

Response format

Content-Type: application/json

Body: Person Query Instance

Response codes

- Status 200 - OK. There is a result matching your request in the body.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section)
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if you have been issuing get by ID and instance with given ID does not exist.

Getting Private Prospects Address Filter used for query

As an authorized user you can get a list of allowed addresses used while running particular query. Only those rows which were recognized in input are present in result of this request. You are always getting addresses in form of application/octet-stream. Although format of response depends on value of **Accept** header you have supplied in request. Also you can request data in necessary encoding using 'Accept-charset' header or 'charset' parameter of 'Accept' header.

Syntax

GET /privateProspects/settingsInstance/{id}/allowedAddresses

Request Format

X-Auth-Token: <your_token>

Accept-Charset: <encoding> (ex. 'iso-8859-1', 'utf8'. UTF-8 is assumed if not specified)

Accept :text/csv OR application/json OR application/zip OR application/gzip ('charset' parameter value is used if there is no Accept-Charset header)

With this header you specify which output format you prefer to get in response. Please note that if you have requested zip/gzip content than values will be returned in CSV format zipped/gzipped

Response format

Content-Type: depends on 'Accept' header value

Body: Stream in format defined by Accept header value. For format refer Address Criteria List.

Response codes

- Status 200 - OK. There is at least one address Filter entry associated with your settings.
- Status 204 - Ok. There are no address Filter entries associated with your settings.
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if settings with given ID does not exist.

Checking correctness of supplied Private Prospects Address Criteria List

As an authorized user you can check whether the address criteria list you are going to provide are able to be considered by the system. For that you should post them to a special URL in Address Criteria List format. In response you'll get Address Criteria List Check Result / Addres criteria list parse error report which contains parse errors and information about recognized entries. When getting address filter parse report as text/csv you can define encoding in which report is going to be provided. Use 'Accept-Charset' header or 'charset' parameter of request for that.

Encodings
Please note that we do our best to detect encoding of content you have supplied in body, but if we fail - there will be used ISO-8859-1. So if you see some issues with parsing or getting corrupt output when requesting uploaded results, please let us know and consider using ISO-8859-1 encoding for your input till we are working on the solution of the problem.

Syntax

POST /privateProspects/allowedAddresses[?maxErrorsCount={maxErrorsCount}]

GET /privateProspects/allowedAddressesParseReport?fileId={fileId}[&maxErrorsCount={maxErrorsCount}]

Request Format

Parameters:

- maxErrorsCount = maximum number of errors to put into response. Non Mandatory. Amount of returned parse errors + invalid items will not exceed given parameter value. Total numbers of errors and invalid items detected in supplied input are returned in response as separate fields
- fileId - Allowed only for GET allowedAddressesParseReport endpoint. Defines ID of pre-loaded file (see Uploading input files) holding address filter.

X-Auth-Token: <your_token>

Accept-Charset: <encoding> (ex. 'iso-8859-1', 'utf8'. UTF-8 is assumed if not specified)

Content-Type: multipart/form-data, where part with text should have text/csv content type

Accept: application/json or text/csv

- In application/json you get up to maxErrorsCount error entries. If not defined you get up to maximum supported by the platform now (currently = 1000)
- In text/csv you get ONLY error items in format <errorLineNumber><TAB><rawValueTakenFromFile>. maxErrorsCount is not considered in that case.

Body: Address Criteria List

Response format

Content-Type: application/json or text/csv (depending on Accept header value). Body is encoded in UTF-8 by default.

Body: Address Criteria List Check Result/Address criteria list parse error report

Response codes

- Status 200 - OK. Parse report is n body
- Status 404 - In case if file with supplied ID could not be found (for GET request)
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).

Getting extended Private Prospects Query Info

As an authorized user you can get extended information about your queries as list of Person Query Info objects. List is ordered from the most recent to most eldest. As far as Query info object encapsulates information about several related objects, it is in general a convenience method which should be used for efficient retrieval of information. Before using this method, please consider usage of separate resources for Queries and results as they are more efficient when you don't need extended information

Syntax

GET /privateProspects/queryInfo?

startIndex={index_value}&count={count_value}

&queryName={qName}

&queryType={qType}

&queryPostDateFrom={dateFrom}

&queryPostDateTo={dateTo}

&queryStatus={NOT_STARTED|IN_PROGRESS|COMPLETED}, …

&resultConsidered={true/false}

&resultHidden={yes/no}

Request Format

Parameters:

- startIndex, count (see Paged Reading)
- queryName - query name. There will be returned all the queries which name matches the given one. Non mandatory. Parameter supports wildcards (* and ?)

- queryPostDateFrom - minimal date of when the query was posted. Returned queries are older then or have the same date as defined by this parameter. Non mandatory
- queryPostDateTo - maximal date of when the query was posted. Returned queries are older then or have the same date as defined by this parameter. Non mandatory
- queryStatus - one or more statuses of queries which should be returned. If not specified - queries set is not filtered by the status.
- resultConsidered - return only results which are considered/not considered in excludes. Not specifying this parameter disables filtering by this criteria
- resultHidden - return only results which are hidden/not hidden. Not specifying this parameter disables filtering by this criteria.

X-Auth-Token: <your_token>

Content-Type: application/json

Accept-Charset: utf-8

Accept: application/json

Response format

Content-Type: application/json

Content-Range: <startIndex>-<endIndex>/<totalCount>

Body: Person Query Info

Response codes

- Status 200 - Ok. There is at least one answer matching given parameters.
- Status 204 - Ok. There are no answers for your query.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section).
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).

Company Prospects

Creating new Company Prospects settings

As an authorized user you can create settings templates for further usage with company prospects. In result of request you will be supplied with created template settings parameters.

> **Encodings**
> Please note that we do our best to detect encoding of content you have supplied in body (for exclusions/addresses parts of request), but if we fail - there will be used ISO-8859-1. So if you see some issues with parsing or getting corrupt output when requesting uploaded results, please let us know and consider using ISO-8859-1 encoding for your input till we are working on the solution of the problem.

Syntax

POST /companyProspects/settingsTemplate

POST /companyProspects/settingsTemplate[?exclusionsFileId={fileId}][&addressFilterFileId={fileId}]

POST /companyProspects/settingsTemplate (multipart, parts: setting, exclusions, addresFilter)

Request Format

Parameters:

- exclusionsFileId - ID of pre-loaded file (see Uploading input files) holding exclusions. Non Mandatory. If not defined template related exclusions are used.
- addressFilterFileId - ID of pre-loaded file (see Uploading input files) holding address filter. Non Mandatory.

X-Auth-Token: <your_token>

Content-Type: application/json

Accept: application/json

Accept-Charset: utf-8

Body: Company Prospects Settings

You should not supply ID as the ID will be generated for you once setting is created. Supplied ID will be ignored. If you are going to supply exclusions or address filter you need to issue multipart request. Also **createdAt** and **modifiedAt** fields values will be ignored even if supplied.

Please note, when posting as multipart it is expected one mandatory part called **setting** which holds your settings values in json and filters (called **exclusions** and **addresFilter**) none of which are mandatory. Also you could post with file IDs of pre-loaded files.

Response format

Content-Type: application/json

Body: Company Prospects Settings

Only settings in json format will be returned. For accessing exclusions or address filter associated with your setting please use separate methods:

- Getting Stored Company Exclusions
- Getting Stored Company Address Filter

Response codes

- Status 200 - OK. Result is in response body.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section)
- Status 401 - Authorization failure (sub-codes are described in Common Error Codes section)

Updating Company Prospects settings

As an authorized user you can update company settings templates which were previously stored. In return you should get actual state of settings you have been modified.

Syntax

PUT /companyProspects/settingsTemplate/{id}

PUT /companyProspects/settingsTemplate/{id}[?exclusionsFileId={fileId}][&addressFilterFileId={fileId}]

Request Format

Parameters:

- exclusionsFileId - ID of pre-loaded file (see Uploading input files) holding exclusions. Non Mandatory. If not defined template related exclusions are used. Posting empty exclusions will not remove associated exclusions. Use DELETE endpoints in order to get rid of them.
- addressFilterFileId - ID of pre-loaded file (see Uploading input files) holding address filter. Non Mandatory. Posting empty address filter will not remove associated address filter items. Use DELETE endpoints in order to get rid of them.

X-Auth-Token: <your_token>

Content-Type: application/json

Accept: application/json

Accept-Charset: utf-8

Body: Company Prospects Settings

If you supply ID different from one used in URL, you'll get error. This is done to prevent from occasional modification of your settings. At the same time if you did not supplied ID within the body - error will not be issued.

Please note that this endpoint does not support multipart requests due to HTTP semantics

Response format

Content type: application/json

Body: Company Prospects Settings

Only settings template in json format will be returned. For accessing exclusions or address filter associated with your setting please use separate methods:

- Getting Stored Company Exclusions
- Getting Stored Company Address Filter

Response codes

- Status 200 - OK. Result is in response body.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section)
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if settings with given ID is not present.
- Status 409 - In case of Conflict (see Handling conflicts)

Copying Company Prospects settings

As an authorized user you can create a copy of your settings templates for further usage with company prospects. In result of request you will be supplied with created copy of template template settings parameters.

Syntax

POST /companyProspects/settingsTemplate/{id}/copy

Request Format

X-Auth-Token: <your_token>

Content-Type: application/json

Accept: application/json

Accept-Charset: utf-8

Body: Copy Action Parameters

You should supply ID of original entity in URL. In case if this entity does not exists or does not belongs to user issuing request there will be returned 404 status

Response format

Content-Type: application/json

Body: Company Prospects Settings

Only settings in json format will be returned. For accessing exclusions or address filter associated with your setting please use separate methods:

- Getting Stored Company Exclusions
- Getting Stored Company Address Filter

Response codes

- Status 200 - OK. Result is in response body.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section)
- Status 401 - Authorization failure (sub-codes are described in Common Error Codes section)
- Status 404 - Entity with deined id is either not exists or does not belongs to user issued request

Deleting Company Prospects settings

As an authorized user you can delete previously stored settings template. Result of operation is determined by returned status code. Please note, that while dropping the settings template you are also dropping such settings attributes like Company Exclusions and Address Criteria List . After issuing request you'll be getting 404 status responses for following operations:

- DELETE /companyProspects/settingsTemplate/{id}
- DELETE /companyProspects/settingsTemplate/{id}/exclusions
- DELETE /companyProspects/settingsTemplate/{id}/allowedAddresses

Syntax

DELETE /companyProspects/settingsTemplate/{id}

Request Format

X-Auth-Token: <your_token>

Accept: application/json

Accept-Charset: utf-8

Body: <no_body_expected>

Response format

Content-Type: application/json

In case of error content type could be application/json with error in body

Body: <no_body_expected>

Response codes

- Status 204 - OK. Setting was located and successfully deleted.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section).
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if settings with given ID does not exist or was already deleted.

Getting information about your Company Prospects settings

As an authorized user you can get information about Company Prospects settings associated with your user. You can get all your settings, or filter by ID or name. The single resource is responsible for the querying. It supports the set of parameters which could be specified in order to make your search more precise.

Syntax

GET /companyProspects/settingsTemplate[?startIndex={startIndex}&count={count}&name={name_pattern}] - get all settings for user issuing request matching given parameters

GET /companyProspects/settingsTemplate/{id} - get settings with given ID

Request Format

Parameters:

- startIndex, count (see Paged reading)
- name - template name. Supports wildcards (? and *)

X-Auth-Token: <your_token>

Accept: application/json

Accept-Charset: utf-8

Body: <no_body_expected>

Response format

Content-Type: application/json

Content-Range: <startIndex>-<endIndex>/<totalCount>

Body: Collection of Company Prospects Setting

Note that in return we will not issue multipart response. Only settings in json format will be returned. For accessing exclusions or address filter associated with your setting please use separate methods:

- Getting Stored Company Exclusions
- Getting Stored Company Address Filter

Response codes

- Status 200 - OK. There is at least one result matching your request parameters in the body.
- Status 204 - Ok. There are no settings matching your request parameters.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section)
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if you have been issuing get by ID and settings with given ID does not exist.

Getting Stored Company Exclusions

As an authorized user you could submit exclusions which should be used when performing your query associated to the settings. Despite you can define exclusions list directly within your POST/PUT request body JSON part, system does not make any difference in exclusions supplied in such a way and supplied as the separate part of multipart request. You are always getting exclusions in form of application/octet-stream. Although format of response depends on value of **Accept** header you have supplied in request. Also you can request data in necessary encoding using 'Accept-charset' header or 'charset' parameter of 'Accept' header.

Syntax

GET /companyProspects/settingsTemplate/{id}/exclusions

Request Format

X-Auth-Token: <your_token>

Accept-Charset: <encoding> (ex. 'iso-8859-1', 'utf8'. UTF-8 is assumed if not specified)

Accept :text/csv OR application/json OR application/zip OR application/gzip ('charset' parameter value is used if there is no Accept-Charset header)

With this header you specify which output format you prefer to get in response. Please note that if you have requested zip/gzip content then values will be returned in CSV format zipped/gzipped

Response format

Content-Type: depends on Accept header value

Body: Stream in format defined by Accept header value. For format refer Private Persons Exclusions.

Response codes

- Status 200 - OK. There is at least one exclusion associated with your settings.
- Status 204 - Ok. There are no exclusions associated with your settings.
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if settings with given ID does not exist.

Updating Stored Company Exclusions

As an authorized user you can update exclusions which can be used with specific settings in future queries. If settings are already associated with some exclusions, it will be replaced with a new set of exclusions.

Syntax

POST /companyProspects/settingsTemplate/{id}/exclusions

POST /companyProspects/settingsTemplate/{id}/exclusions?fileId={preloaded_file_id}

Request Format

Parameters:

- one of the next respectively:
  - exclusionsFilterFile - file with allowed addresses (see Company Exclusions)
  - storedFileId - ID of pre-loaded file (see Uploading input files) holding address filter.

X-Auth-Token: <your_token>

Content-Type: application/json

Accept: application/json

Accept-Charset: utf-8

Response format

Content-Type: application/json

Body: <no_body_expected>

Response codes

- Status 200 - OK. Exclusions are updated.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section)
- Status 401 - Authorization failure (sub-codes are described in Common Error Codes section)
- Status 404 - Entity with provided id is either does not exist or does not belong to user issued request

Deleting Stored Company Exclusions

As an authorized user you can delete exclusions associated with a template.

Syntax

DELETE /companyProspects/settingsTemplate/{id}/exclusions

Request Format

X-Auth-Token: <your_token>

Content-Type: application/json

Accept: application/json

Accept-Charset: utf-8

Response format

Content-Type: application/json

Body: <no_body_expected>

Response codes

- Status 204 - OK. Settings were located and exclusions were successfully deleted.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section).
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if settings with given ID does not exist or it doesn't have exclusions. Also error will be returned if exclusions were already deleted.

Getting Stored Company Address Filter

As an authorized user you could submit list of allowed addresses which should be used when performing your query associated with the settings. Despite you can define allowed addresses list directly within your POST/PUT request body JSON part, system does not make any difference in addresses supplied in such a way and supplied as the separate part of multipart request. You are always getting addresses in form of application/octet-stream. Although format of response depends on value of **Accept** header you have supplied in request. Also you can request data in necessary encoding using 'Accept-charset' header or 'charset' parameter of 'Accept' header.

Syntax

GET /companyProspects/settingsTemplate/{id}/allowedAddresses

Request Format

X-Auth-Token: <your_token>

Accept-Charset: <encoding> (ex. 'iso-8859-1', 'utf8'. UTF-8 is assumed if not specified)

Accept :text/csv OR application/json OR application/zip OR application/gzip ('charset' parameter value is used if there is no Accept-Charset header)

With this header you specify which output format you prefer to get in response. Please note that if you have requested zip/gzip content than values will be returned in CSV format zipped/gzipped.

Response format

Content-Type: depends on 'Accept' header value

Body: Stream in format defined by Accept header value. For format refer Address Criteria List.

Response codes

- Status 200 - OK. There is at least one address Filter entry associated with your settings.
- Status 204 - O?. There are no address Filter entries associated with your settings.
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if settings with given ID does not exist.

Updating Stored Company Address Filter

As an authorized user you can update a list of allowed addresses associated with specific settings.

Syntax

POST /companyProspects/settingsTemplate/{id}/allowedAddresses

POST /comanyProspects/settingsTemplate/{id}/allowedAddresses?fileId={preloaded_file_id}

Request Format

Parameters:

- one of the next respectively:
  - addressFilterFile - file with allowed addresses (see Address Criteria List)
  - storedFileId - ID of pre-loaded file (see Uploading input files) holding address filter. Non Mandatory.

X-Auth-Token: <your_token>

Content-Type: application/json

Accept: application/json

Accept-Charset: utf-8

Response format

Content-Type: application/json

Body: <no_body_expected>

Response codes

- Status 200 - OK. Address filter is successfully updated.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section)
- Status 401 - Authorization failure (sub-codes are described in Common Error Codes section)
- Status 404 - Entity with provided id either does not exist or does not belong to user issued request

Deleting Stored Company Address Filter

As an authorized user you can delete previously stored list of allowed addresses. Result of operation is determined by returned status code. After issuing request you'll be getting 404 status responses for following operations:

- DELETE /companyProspects/settingsTemplate/{id}/allowedAddresses

Syntax

DELETE /companyProspects/settingsTemplate/{id}/allowedAddresses

Request Format

X-Auth-Token: <your_token>

Accept: application/json

Accept-Charset: utf-8

Body: <no_body_expected>

Response format

Content-Type: application/json

In case of error content type could be application/json with error in body

Body: <no_body_expected>

Response codes

- Status 204 - OK. List of addresses was located and successfully deleted.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section).
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if list of addresses with given ID does not exist or was already deleted.

Estimating number of answers for Company Prospects Query

As an authorized user you can estimate how many answers you are going to get if you run a query with specified settings and additional overrides. For that you should POST your query parameters to the special resource. In response you should get the Dry Run Result which holds results of estimation. The estimation is preformed in scope of single request-response cycle (unlike when posting to /companyProspects/queries resource), so you should consider that performing could take some time. Please, be advised about that when setting up your client connection timeout values.

> **Encodings**
> Please note that we do our best to detect encoding of content you have supplied in body (for exclusions/addresses parts of request), but if we fail - there will be used ISO-8859-1. So if you see some issues with parsing or getting corrupt output when requesting uploaded results, please let us know and consider using ISO-8859-1 encoding for your input till we are working on the solution of the problem.

Syntax

POST /companyProspects/queries/dryRun

POST /companyProspects/queries/dryRun[?exclusionsFileId={fileId}][&addressFilterFileId={fileId}]

Request Format

Parameters:

- exclusionsFileId - ID of pre-loaded file (see Uploading input files) holding exclusions. Non Mandatory. If not defined then template related exclusions are used. If defined then exclusions get merged with template exclusions for the run
- addressFilterFileId - ID of pre-loaded file (see Uploading input files) holding address filter. Non Mandatory. If not defined then template related address filter is used. If defined then given address filter is used. Please note that supplied address filter has priority over GEO filter. So if your template has Geo filter defined than it will not be considered.

X-Auth-Token: <your_token>

Content-Type: multipart/form-data if you are uploading exclusions/address filters in body; application/json if you are referring pre-loaded files

Accept: application/json

Accept-Charset: utf-8

Body: Private Prospects Query

Response format

Content-Type: application/json

Body: Dry Run Result

Response codes

- Status 200 - OK. Results of estimation are in the body.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section)
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if settings with given ID is not present. In case if file with defined ID does not exists for exclusions/address filters

Posting Company Prospects Query for execution

As an authorized user you can post a query for execution. While posting the query you must supply the reference to settings which you would like to use. Additionally you can supply settings parameters which would override corresponding parameters within the settings.

> **Encodings**
> Please note that we do our best to detect encoding of content you have supplied in body (for exclusions/addresses parts of request), but if we fail - there will be used ISO-8859-1. So if you see some issues with parsing or getting corrupt output when requesting uploaded results, please let us know and consider using ISO-8859-1 encoding for your input till we are working on the solution of the problem.

Syntax

POST /companyProspects/queries

POST /companyProspects/queries[?exclusionsFileId={fileId}][&addressFilterFileId={fileId}]

Request Format

Parameters:

- exclusionsFileId - ID of pre-loaded file (see Uploading input files) holding exclusions. Non Mandatory. If not defined then template related exclusions are used. If defined then exclusions get merged with template exclusions for the run
- addressFilterFileId - ID of pre-loaded file (see Uploading input files) holding address filter. Non Mandatory. If not defined then template related address filter is used. If defined then given address filter is used. Please note that supplied address filter has precedence over GEO filter. So if your template has Geo filter defined then it will not be considered.

X-Auth-Token: <your_token>

Content-Type: multipart/form-data if you are uploading exclusions/address filters in body; application/json if you are referring pre-loaded files

Accept: application/json

Accept-Charset: utf-8

Body: Prospects Query

Please note, that if you have supplied Address Criteria List this would replace the one defined in associated query settings. If you have supplied Company Exclusions they will be merged with exclusions in settings and both exclusions are going to be used when calculating query result.

Response format

Content-Type: application/json

Body: Prospects Query

In response you will not get associated exclusions or allowed addresses. Supplied exclusions and addresses are associated with settings instance created automatically from your template and overrides (and mentioned in query object under settingsId field). They could be retrieved using separate resources:

- Getting query settings instance related exclusions
- Getting query settings instance related addresses limitations

Response codes

- Status 200 - OK. Posted query metadata is in the body. Metadata also contains assigned ID.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section).
- Status 404 - In case if file with defined ID does not exist for exclusions/address filters
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if settings with given ID is not present. In case if file with defined ID does not exist for exclusions/address filters.

Getting information about your Company Prospects queries

As an authorized user you can get information about Company Prospects queries you have posted. You can get all your settings, or filter by ID or name. The single resource is responsible for the querying.

Syntax

GET /companyProspects/queries[?startIndex={startIndex}&count={count}] - get all queries for user has issued.

GET /companyProspects/queries/{id} - get query with given ID

Request Format

Parameters:

- startIndex, count (see Paged reading)

X-Auth-Token: <your_token>

Accept: application/json

Accept-Charset: utf-8

Body: <no_body_expected>

Response format

Content-Type: application/json

Body: Collection of Prospects Query

Content-Range: <startIndex>-<endIndex>/<totalCount>

In response you will not get associated exclusions or allowed addresses. Supplied exclusions and addresses are associated with results and could be retrieved using separate resources:

- Getting Stored Company Exclusions
- Getting Stored Company Address Filter

Response codes

- Status 200 - OK. There is at least one result matching your request parameters in the body.
- Status 204 - Ok. There are no settings matching your request parameters.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section)
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if you have been issuing get by ID and query with given ID does not exist.

Getting information about your particular query progress

As the part of response for query resource you are getting information about the status of query execution. Please refer Getting information about your Company Prospects queries in order to find out how to get information about query. You can also refer Company Prospects Query object definition to get info about the structure which is used for holding query status info.

Getting information about your Company Prospects Query results

As an authorized user you can get information about Company Prospects Results. You can get information about all available results or use request parameters in order to make more precise selection. In scope of this request you are not getting actual query results, just metadata describing results being produced. In order to find out how to get access to the results you should refer

- Getting all your Company Prospects query results
- Getting your Company Prospects query results in chunks

Syntax

GET /companyProspects/results - get all results produced so far for your user

GET /companyProspects/results[?queryId={queryId}&startIndex={startIndex}&count={count}] - get all results produced so far for your user

GET /companyProspects/results/{id} - get results with given ID

Request Format

Parameters:

- startIndex, count (see Paged reading)

X-Auth-Token: <your_token>

Accept: application/json

Accept-Charset: utf-8

Body: <no_body_expected>

Response format

Content-Type: application/json

Body: Collection of Company Prospects Query Result

Content-Range: <startIndex>-<endIndex>/<totalCount>

Response codes

- Status 200 - OK. There is at least one result matching your request parameters in the body.
- Status 204 - Ok. There are no results matching your request parameters.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section)
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if you have been issuing get by ID and result with given ID does not exist.

Getting all your Company Prospects query results

As an authorized user you can get results of query which has been completed successfully. Results are returned as text/csv. Also you can request data in necessary encoding using 'Accept-charset' header or 'charset' parameter of 'Accept' header (like "Accept: text/csv;charset=iso-8859-1").

Syntax

GET /companyProspects/results/{id}/valuesStream

Request Format

X-Auth-Token: <your_token>

Accept : text/csv ('charset' parameter value is used if there is no Accept-Charset header)

Accept-Charset: <encoding> (ex. 'iso-8859-1', 'utf8'. UTF-8 is assumed if not specified)

Response format

Content-Type: text/csv

Body: Stream in format defined by Accept header value. For format refer Company Prospects Query Result Value.

Response codes

- Status 200 - Ok. There is at least one answer produced for your query.
- Status 204 - Ok. There are no answers for your query.
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section).
- Status 404 - In case if results with given ID does not exist.
- Status 403 - In case if your result is locked.

Getting your Company Prospects query results in chunks

As an authorized user you can get results of query which has been completed successfully. Results are returned directly within the body. This way of getting results differs from one described in Getting all your Company Prospects query results because it allows you reducing output set. Using that method could be useful for organizing paged review of results produced in response to your query. You can also find useful using Private Prospects Query Result's **answerCount** property for paging needs. Also you can get results in different formats. Returned data format depends on "Accept" header value.

Syntax

GET /companyProspects/results/{id}/values[?startIndex={index_value}&count={count_value}]

Request Format

Parameters:

- startIndex, count (see Paged Reading)

X-Auth-Token: <your_token>

Accept : text/csv or application/json

Accept-Charset: utf-8

Response format

Content-Type: text/csv or application/json (depending on accept header value)

Body: Collection of Company Prospects Query Result Value constrained according to parameters of request

Response codes

- Status 200 - Ok. There is at least one answer produced for your query.
- Status 204 - Ok. There are no answers for your query.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section).
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if results with given ID does not exist.
- Status 403 - In case if your result is locked.

Updating Company Prospects query results related attributes

Query results have a few parameters which affects consequent queries you issue. As an authorized user you are able to modify the attributes using API.

Syntax

PUT /companyProspects/results/{id}

Request Format

X-Auth-Token: <your_token>

Content-Type: application/json

Accept-Charset: utf-8

Accept: application/json

Body: Company Prospects Query Result

If you supply ID different from one used in URL, you'll get error. This is done to prevent from occasional modification of your settings. At the same time if you did not supplied ID within the body - error will not be issued.

Response format

Content-Type: application/json

Body: Company Prospects Query Result

Response codes

- Status 200 - OK. Result is in response body.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section)
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if result with given ID is not present.
- Status 409 - In case of Conflict (see Handling conflicts)

Getting Company Query Instance settings by query id

As an authorized user you can get settings instance used with a specific query.

Syntax

GET /companyProspects/queries/{id}/settings

Request Format

X-Auth-Token: <your_token>

Accept: application/json

Accept-Charset: utf-8

Body: <no_body_expected>

Response format

Content-Type: application/json

Body: Company Query Instance

Response codes

- Status 200 - OK. There is a result matching your request in the body.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section)
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if you have been issuing get by ID and query with given ID does not exist.

Getting Company Query Instance settings by its id

As an authorized user you can get settings instance by its id.

Syntax

/companyProspects/settingsInstance/{id}

Request Format

X-Auth-Token: <your_token>

Accept: application/json

Accept-Charset: utf-8

Body: <no_body_expected>

Response format

Content-Type: application/json

Body: Company Query Instance

Response codes

- Status 200 - OK. There is a result matching your request in the body.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section)
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if you have been issuing get by ID and instance with given ID does not exist.

Getting Company Prospects Exclusions used for query

As an authorized user you can get company used to calculate particular result. Please note that this could be composition of persistent exclusions associated with setting/query and additional exclusions supplied with query. Only those rows which were recognized in input are present in result of this request. You are always getting exclusions in form of application/octet-stream. Although format of response depends on value of **Accept** header you have supplied in request. Also you can request data in necessary encoding using 'Accept-charset' header or 'charset' parameter of 'Accept' header.

Syntax

GET /companyProspects/results/{id}/exclusions

Request Format

X-Auth-Token: <your_token>

Accept-Charset: <encoding> (ex. 'iso-8859-1', 'utf8'. UTF-8 is assumed if not specified)

Accept :text/csv OR application/json OR application/zip OR application/gzip ('charset' parameter value is used if there is no Accept-Charset header)

With this header you specify which output format you prefer to get in response. Please note that if you have requested zip/gzip content than values will be returned in CSV format zipped/gzipped

Response format

Content-Type: depends on Accept header value

Body: Stream in format defined by Accept header value. For format refer Company Exclusions.

Response codes

- Status 200 - OK. There is at least one exclusion associated with your settings.

- Status 204 - Ok. There are no exclusions associated with your settings.
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if settings with given ID does not exist.

Checking correctness of supplied Company Exclusions

As an authorized user you can check whether the exclusions you are going to provide are able to be considered by the system. For that you should post them to a special URL in Company Exclusions format. In response you'll get Company Exclusions Check Results which contains parse errors and information about recognized entries. When getting exclusions parse report as text/csv you can define encoding in which report is going to be provided. Use 'Accept-Charset' header or 'charset' parameter of request for that.

> **Encodings**
> Please note that we do our best to detect encoding of content you have supplied in body, but if we fail - there will be used ISO-8859-1. So if you see some issues with parsing or getting corrupt output when requesting uploaded results, please let us know and consider using ISO-8859-1 encoding for your input till we are working on the solution of the problem.

Syntax

POST /companyProspects/exclusions[?maxErrorsCount={maxErrorsCount}]

GET /companyProspects/exclusionsParseReport?fileId={fileId}[&maxErrorsCount={maxErrorsCount}]

Request Format

Parameters:

- maxErrorsCount = maximum number of errors to put into response. Non mandatory. Total number of errors detected in supplied input is returned in response as separate field
- fileId - Allowed only for GET exclusionsParseReport endpoint. Defines ID of pre-loaded file (see Uploading input files) holding exclusions.

X-Auth-Token: <your_token>.

Accept-Charset: <encoding> (ex. 'iso-8859-1', 'utf8'. UTF-8 is assumed if not specified)

Content-Type: multipart/form-data, where part with text should have text/csv content type

Accept: application/json or text/csv

- In application/json you get up to maxErrorsCount error entries. If not defined you get up to maximum supported by the platform now (currently = 1000)
- In text/csv you get ONLY error items in format <errorLineNumber><TAB><rawValueTakenFromFile>. maxErrorsCount is not considered in that case.

Body: Company Private Persons Exclusions

Response format

Content-Type: application/json or text/csv (depending on Accept header value). Body is encoded in UTF-8

Body: Company Exclusions Check Results

Response codes

- Status 200 - OK. Parse report is in the body
- Status 404 - In case if file with supplied ID could not be found (for GET request)
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).

Getting Company Prospects Address Filter used for query

As an authorized user you can get a list of allowed addresses used to run a particular query. Only those rows which were recognized in input are present in result of this request. You are always getting addresses in form of application/octet-stream. Although format of response depends on value of **Accept** hea der you have supplied in request. Also you can request data in necessary encoding using 'Accept-charset' header or 'charset' parameter of 'Accept' header.

Syntax

GET /companyProspects/settingsInstance/{id}/allowedAddresses

Request Format

X-Auth-Token: <your_token>

Accept-Charset: <encoding> (ex. 'iso-8859-1', 'utf8'. UTF-8 is assumed if not specified)

Accept :text/csv OR application/json OR application/zip OR application/gzip ('charset' parameter value is used if there is no Accept-Charset header)

With this header you specify which output format you prefer to get in response. Please note that if you have requested zip/gzip content than values will be returned in CSV format zipped/gzipped

Response format

Content-Type: depends on 'Accept' header value

Body: Stream in format defined by Accept header value. For format refer Address Criteria List.

Response codes

- Status 200 - OK. There is at least one address Filter entry associated with your settings.
- Status 204 - Ok. There are no address Filter entries associated with your settings.
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if settings with given ID does not exist.

Checking correctness of supplied Company Address Criteria List

As an authorized user you can check whether the address criteria list you are going to provide are able to be considered by the system. For that you should post them to a special URL in Address Criteria List format. In response you'll get #Address Criteria List Check Result which contains parse errors and information about recognized entries. When getting address filter parse report as text/csv you can define encoding in which report is going to be provided. Use 'Accept-Charset' header or 'charset' parameter of request for that.

> **Encodings**
> Please note that we do our best to detect encoding of content you have supplied in body, but if we fail - there will be used ISO-8859-1. So if you see some issues with parsing or getting corrupt output when requesting uploaded results, please let us know and consider using ISO-8859-1 encoding for your input till we are working on the solution of the problem.

Syntax

POST /companyProspects/allowedAddresses[?maxErrorsCount={maxErrorsCount}]

GET /companyProspects/allowedAddressesParseReport?fileId={fileId}[&maxErrorsCount={maxErrorsCount}]

Request Format

Parameters:

- maxErrorsCount = maximum number of errors to put into response. Non Mandatory. Amount of returned parse errors + invalid items will not exceed given parameter value. Total numbers of errors and invalid items detected in supplied input are returned in response as separate fields
- fileId - Allowed only for GET allowedAddressesParseReport endpoint. Defines ID of pre-loaded file (see Uploading input files) holding address filter.

X-Auth-Token: <your_token>

Accept-Charset: <encoding> (ex. 'iso-8859-1', 'utf8'. UTF-8 is assumed if not specified)

Content-Type: multipart/form-data, where part with text should have text/csv content type

Accept: application/json or text/csv

- In application/json you get up to maxErrorsCount error entries. If not defined you get up to maximum supported by the platform now (currently = 1000)
- In text/csv you get ONLY error items in format <errorLineNumber><TAB><rawValueTakenFromFile>. maxErrorsCount is not considered in that case.

Body: Address Criteria List

Response format

Content-Type: application/json or text/csv (depending on Accept header value). Body is encoded in UTF-8

Body: Address Criteria List Check Result

Response codes

- Status 200 - OK. Parse report is n body
- Status 404 - In case if file with supplied ID could not be found (for GET request)
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).

Getting extended Company Prospects Query Info

As an authorized user you can get extended information about your queries as list of Company Query Info objects. List is ordered from the most recent to most eldest. As far as Query info object encapsulates information about several related objects, it is in general a convenience method which should be used for efficient retrieval of information. Before using this method, please consider usage of separate resources for Queries and results as they are more efficient when you don't need extended information

Syntax

GET /companyProspects/queryInfo?

startIndex={index_value}&count={count_value}

&queryName={qName}

&queryType={qType}

&queryPostDateFrom={dateFrom}

&queryPostDateTo={dateTo}

&queryStatus={NOT_STARTED|IN_PROGRESS|COMPLETED}, …

&resultConsidered={true/false}

&resultHidden={yes/no}

Request Format

Parameters:

- startIndex, count (see Paged Reading)
- queryName - query name. There will be returned all the queries which name matches the given one. Non mandatory. Supports wildcards (* and ?)
- queryPostDateFrom - minimal date of when the query was posted. Returned queries are older then or have the same date as defined by this parameter. Non mandatory
- queryPostDateTo - maximal date of when the query was posted. Returned queries are older then or have the same date as defined by this parameter. Non mandatory
- queryStatus - one or more statuses of queries which should be returned. If not specified - queries set is not filtered by the status.
- resultConsidered - return only results which are considered/not considered in excludes. Not specifying this parameter disables filtering by this criteria
- resultHidden - return only results which are hidden/not hidden. Not specifying this parameter disables filtering by this criteria.

X-Auth-Token: <your_token>

Content-Type: application/json

Accept-Charset: utf-8

Accept: application/json

Response format

Content-Type: application/json

Content-Range: <startIndex>-<endIndex>/<totalCount>

Body: Company Query Info

Response codes

- Status 200 - Ok. There is at least one answer matching given parameters.
- Status 204 - Ok. There are no answers for your query.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section).
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).

Group Search

Getting information about your Group Search settings

As an authorized user you can get information about Group Search settings associated with your user. You can get all your settings, or filter by ID or name. The single resource is responsible for the querying. It supports the set of parameters which could be specified in order to make your search more precise.

Syntax

GET /groupSearch/settingsTemplate[?startIndex={startIndex}&count={count}&name={name_pattern}] - get all settings for user issuing request matching parameters

GET /groupSearch/settingsTemplate/{id} - get settings with given ID

Request Format

Parameters:

- startIndex, count (see Paged Reading)
- name - pattern of template name. Supports wildcards (* and ?)

X-Auth-Token: <your_token>

Accept: application/json

Accept-Charset: utf-8

Body: <no_body_expected>

Response format

Content-Type: application/json

Content-Range: <startIndex>-<endIndex>/<totalCount>

Body: Collection of Group Search Settings

Response codes

- Status 200 - OK. There is at least one result matching your request parameters in the body.
- Status 204 - Ok. There are no settings matching your request parameters.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section)
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if you have been issuing get by ID and settings with given ID does not exist.

Creating new Group Search settings

As an authorized user you can create settings templates for further usage for your group searches. In result of request you will be supplied with created template settings parameters. Please note that when you are creating settings template you don't need to supply your actual input as you are working out reusable settings set. Input will be required when you'll decide to use your the template for posting the query.

Syntax

POST /groupSearch/settingsTemplate

Request Format

X-Auth-Token: <your_token>

Content-Type: application/json

Accept: application/json

Accept-Charset: utf-8

Body: Group Search Settings

You should not supply ID as the ID will be generated for you once setting is created. Supplied ID will be ignored If you are going to supply exclusions or address filter you need to issue multipart request. Also **createdAt** and **modifiedAt** fields values will be ignored even if supplied

Response format

Content-Type: application/json

Body: Group Search Settings

Created settings JSON will be returned with ID assigned.

Response codes

- Status 200 - OK. Result is in response body.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section)
- Status 401 - Authorization failure (sub-codes are described in Common Error Codes section)

Copying Group Search settings

As an authorized user you can create a copy of your settings templates for further usage with private person prospects. In result of request you will be supplied with created copy of template template settings parameters.

Syntax

POST /groupSearch/settingsTemplate/{id}/copy

Request Format

X-Auth-Token: <your_token>

Content-Type: application/json

Accept: application/json

Accept-Charset: utf-8

Body: Copy Action Parameters

You should supply ID of original entity in URL. In case if this entity does not exists or does not belongs to user issuing request there will be returned 404 status

Response format

Content-Type: application/json

Body: Group Search Settings

Created settings JSON will be returned

Response codes

- Status 200 - OK. Result is in response body.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section)
- Status 401 - Authorization failure (sub-codes are described in Common Error Codes section)
- Status 404 - Entity with deined id is either not exists or does not belongs to user issued request

Updating Group Search settings

As an authorized user you can update settings templates which were previously stored. In return you should get actual state of settings you have been modified.

Syntax

PUT /groupSearch/settingsTemplate/{id}

Request Format

X-Auth-Token: <your_token>

Content-Type: application/json

Accept: application/json

Accept-Charset: utf-8

Body: Group Search Settings

If you supply ID different from one used in URL, you'll get error. This is done to prevent from occasional modification of your settings. At the same time if you did not supplied ID within the body - error will not be issued.

Response format

Content type: application/json

Body: Group Search Settings

Created settings JSON will be returnedwith new ID assigned.

Response codes

- Status 200 - OK. Result is in response body.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section)
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if settings with given ID is not present.
- Status 409 - In case of Conflict (see Handling conflicts)

Deleting Group Search settings

As an authorized user you can delete previously stored settings template. Result of operation is determined by returned status code.

Syntax

DELETE /groupSearch/settingsTemplate/{id}

Request Format

X-Auth-Token: <your_token>

Accept: application/json

Accept-Charset: utf-8

Body: <no_body_expected>

Response format

Content-Type: application/json

In case of error content type could be application/json with error in body

Body: <no_body_expected>

Response codes

- Status 204 - OK. Setting was located and successfully deleted.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section).
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if settings with given ID does not exist or was already deleted.

Estimating number of answers for group query

As an authorized user you can estimate how many answers you are going to get if you run a query with specified settings and additional overrides. For that you should POST your query parameters to the special resource. In response you should get the Dry Run Result which holds results of estimation. The estimation is preformed in scope of single request-response cycle (unlike when posting to /groupSearch/queries resource), so you should consider that performing could take some time. Please, be advised about that when setting up your client connection timeout values.

> **Encodings**
> Please note that we do our best to detect encoding of content you have supplied in body (for exclusions/addresses parts of request), but if we fail - there will be used ISO-8859-1. So if you see some issues with parsing or getting corrupt output when requesting uploaded results, please let us know and consider using ISO-8859-1 encoding for your input till we are working on the solution of the problem.

Syntax

POST /groupSearch/queries/dryRun

POST /groupSearch/queries/dryRun?inputFileId={fileId}

Request Format

Parameters

- inputFileId - ID of pre-loaded file (see Uploading input files) holding input for your query.
- queryInput - this should be a name of a part holding your query input if you are posting input in scope of request.
- setting - in case if the queryInput part is defined, this should be the name of part holding the settings overrides for the query as well as name and settingTemplateId.

X-Auth-Token: <your_token>

Content-Type: multipart/form-data if you are uploading exclusions/address filters in body; application/json if you are referring pre-loaded files

Accept: application/json

Accept-Charset: utf-8

Body: Private Prospects Query

Response format

Content-Type: application/json

Body: Dry Run Result

Response codes

- Status 200 - OK. Results of estimation are in the body.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section)
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if settings with given ID is not present. In case if file with defined ID does not exists for exclusions/address filters

Getting information about your group queries

As an authorized user you can get information about Group queries you have posted. You can get all your settings, or filter by ID or name. The single resource is responsible for the querying.

Syntax

GET /groupSearch/queries[?startIndex={startIndex}&count={count}] - get all queries for user has issued.

GET /groupSearch/queries/{id} - get query with given ID

Request Format

Parameters:

- startIndex, count (see Paged reading)

X-Auth-Token: <your_token>

Accept: application/json

Accept-Charset: utf-8

Body: <no_body_expected>

Response format

Content-Type: application/json

Body: Collection of Group Query

Content-Range: <startIndex>-<endIndex>/<totalCount>

In response you will not get associated input. Supplied input is associated with query settings and could be retrieved using separate resources:

- Getting your Group Query Input

Response codes

- Status 200 - OK. There is at least one result matching your request parameters in the body.
- Status 204 - Ok. There are no settings matching your request parameters.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section)
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if you have been issuing get by ID and query with given ID does not exist.

Getting information about your particular group query progress

As the part of response for query resource you are getting information about the status of query execution. Please refer Getting information about your Company Prospects queries in order to find out how to get information about query. You can also refer Private Prospects Query object definition to get info about the structure which is used for holding query status info.

Posting a group query for execution

As an authorized user you can post a group query for execution. While posting the query you must supply the reference to settings which you would like to use. Additionally you can supply settings parameters which would override corresponding parameters within the settings.

> **Encodings**
> Please note that we do our best to detect encoding of content you have supplied in body (for queryInput part of request), but if we fail - there will be used ISO-8859-1. So if you see some issues with parsing or getting corrupt output when requesting uploaded results, please let us know and consider using ISO-8859-1 encoding for your input till we are working on the solution of the problem.

Syntax

POST /groupSearch/queries

POST /groupSearch/queries?inputFileId={fileId}[&inputFileName={fileName}]

Request Format

Parameters:

- inputFileId - ID of pre-loaded file (see Uploading input files) holding input for your query.
- inputFileName - name of file you post as input. Non Mandatory. Will be used when you'll try to download input. If not specified - will be generated.
- queryInput - this should be a name of a part holding your query input if you are posting input in scope of request. File name parameter will be taken in this case from 'filename' key supplied along with the part in content-disposition section
- setting - in case if the queryInput part is defined, this should be the name of part holding the settings overrides for the query as well as name and settingTemplateId.

X-Auth-Token: <your_token>

Content-Type: multipart/form-data if you are uploading input in body; application/json if you are referring pre-loaded files

Accept: application/json

Accept-Charset: utf-8

Body: Group Query Definition

Response format

Content-Type: application/json

Body: Group Query

In response you will not get associated input. Supplied input is associated with settings instance created automatically from your template and overrides (and mentioned in query object under settingsId field). They could be retrieved using separate resources:

- Getting your Group Query Input

Response codes

- Status 200 - OK. Posted query metadata is in the body. Metadata also contains assigned ID.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section).
- Status 404 - In case if file with defined ID does not exist for exclusions/address filters
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if settings with given ID is not present. In case if file with defined ID does not exist for query input.

Getting group query instance settings by query id

As an authorized user you can get settings instance used with a specific query.

Syntax

GET /groupSearch/queries/{id}/settings

Request Format

X-Auth-Token: <your_token>

Accept: application/json

Accept-Charset: utf-8

Body: <no_body_expected>

Response format

Content-Type: application/json

Body: Group Search Settings Instance

Response codes

- Status 200 - OK. There is a result matching your request in the body.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section)
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if you have been issuing get by ID and query with given ID does not exist.

Getting group query instance settings by its id

As an authorized user you can get settings instance by its id.

Syntax

GET /groupSearch/settingsInstance/{id}

Request Format

X-Auth-Token: <your_token>

Accept: application/json

Accept-Charset: utf-8

Body: <no_body_expected>

Response format

Content-Type: application/json

Body: Group Search Settings Instance

Response codes

- Status 200 - OK. There is a result matching your request in the body.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section)
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if you have been issuing get by ID and instance with given ID does not exist.

Getting information about your group query results

As an authorized user you can get information about Group Search Results. You can get information about all available results or use request parameters in order to make more precise selection. In scope of this request you are not getting actual query results, just metadata describing results being produced. In order to find out how to get access to the results you should refer

- Getting all your group query results
- Getting your group query results in chunks

Syntax

GET /groupSearch/results - get all results metadata produced so far for your user

GET /groupSearch/results[?startIndex={index_value}&count={count_value}] - get all results metadata produced so far for your user

GET /groupSearch/results/{id} - get results metadata with given ID

GET /groupSearch/results/{queryId} - get results metadata with given ID

Request Format

Parameters:

- startIndex, count (see Paged reading)
- queryId - set this to id of your query to get corresponding results metadata

X-Auth-Token: <your_token>

Accept: application/json

Accept-Charset: utf-8

Body: <no_body_expected>

Response format

Content-Type: application/json

Body: Collection of Group Query Result

Content-Range: <startIndex>-<endIndex>/<totalCount>

Response codes

- Status 200 - OK. There is at least one result matching your request parameters in the body.
- Status 204 - Ok. There are no results matching your request parameters.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section)
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if you have been issuing get by ID and result with given ID does not exist.

Getting all your group query results

As an authorized user you can get results of query which has been completed successfully. Results are returned as text/csv. Also you can request data in necessary encoding using 'Accept-charset' header or 'charset' parameter of 'Accept' header (like "Accept: text/csv;charset=iso-8859-1").

Syntax

GET /groupSearch/results/{id}/valuesStream

Request Format

X-Auth-Token: <your_token>

Accept : text/csv ('charset' parameter value is used if there is no Accept-Charset header)

Accept-Charset: <encoding> (ex. 'iso-8859-1', 'utf8'. UTF-8 is assumed if not specified)

Response format

Content-Type: text/csv

Body: Stream in format defined by Accept header value. For format refer Group Query Result Value.

Response codes

- Status 200 - Ok. There is at least one answer produced for your query.
- Status 204 - Ok. There are no answers for your query.
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section).
- Status 404 - In case if results with given ID does not exist.
- Status 403 - In case if your result is locked.

Getting your group query results in chunks

As an authorized user you can get results of query which has been completed successfully. Results are returned directly within the body. This way of getting results differs from one described in Getting all your group query results because it allows you reducing output set. Using that method could be useful for organizing paged review of results produced in response to your query. You can also find useful using Group Query Result's **answerCount** property for paging needs. Also you can get results in different formats. Returned data format depends on "Accept" header value.

Syntax

GET /groupSearch/results/{id}/values[?startIndex={index_value}&count={count_value}&transpose={true/false}&outNonParsed={true/false}]

Request Format

Parameters:

- startIndex, count (see Paged Reading)
- transpose - a CSV format related Boolean flag. If transposed = true, all the results related to a single question are returned in one line. See [transposed format link goes here]. Please note that this flag is ignored for JSON values.
- outNonParsed - if defined then lines which contains input lines with parse errors will be present in output. Matching template for such lines will hold 'parse error' text. If line was formed with correct separator, then only ID of error rows will be outputted else entire line is considered to be ID and will be placed to output.

X-Auth-Token: <your_token>

Accept : text/csv or application/json

Accept-Charset: utf-8

Response format

Content-Type: text/csv or application/json (depending on accept header value)

Body: Collection of Group Query Result Value constrained according to parameters of request

Response codes

- Status 200 - Ok. There is at least one answer produced for your query.
- Status 204 - Ok. There are no answers for your query.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section).
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if results with given ID does not exist.
- Status 403 - In case if your result is locked.

Updating group query results related attributes

Query results have a few parameters which affect consequent queries you issue. As an authorized user you are able to modify the attributes using API.

Syntax

PUT /groupSearch/results/{id}

Request Format

X-Auth-Token: <your_token>

Content-Type: application/json

Accept-Charset: utf-8

Accept: application/json

Body: Group Query Result

If you supply ID different from one used in URL, you'll get an error. This is done to prevent from occasional modification of your settings. At the same time if you did not supply ID within the body - an error will not be issued.

Response format

Content-Type: application/json

Body: Group Query Result

Response codes

- Status 200 - OK. Result is in response body.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section)
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if result with given ID is not present.
- Status 409 - In case of Conflict (see Handling conflicts)

Checking correctness of supplied Questions for Group Search

As an authorized user you can check whether the input you are going to provide is able to be handled by the system. For that you should post it to a special URL in Tab Separated values format. In response you'll get Group Questions Check Result/Group Query Parse error report which contains parse errors and information about recognized entries. When getting input parse report as text/csv you can define encoding in which report is going to be provided. Use 'Accept-Charset' header or 'charset' parameter of request for that.

Also if you have already posted a query for execution you can get a parse report on questions you have used if you know settings id used for posting your query. As far as the query input is a part of your query settings the parse report for input is extracted using groupSearch/settingsInstance resource (see syntax section).

> **Encodings**
> Please note that we do our best to detect encoding of content you have supplied in body, but if we fail - there will be used ISO-8859-1. So if you see some issues with parsing or getting corrupt output when requesting uploaded results, please let us know and consider using ISO-8859-1 encoding for your input till we are working on the solution of the problem.

Syntax

POST /groupSearch/questions[?maxErrorCount=<>]

GET /groupSearch/questionsParseReport?fileId={preloaded_file_id}[&maxErrorCount=<>]

GET /groupSearch/settingsInstance/{id}/queryInput/questionsParseReport

Request Format

Parameters:

- maxErrorsCount = maximum number of errors to put into response. Non mandatory. Total number of errors detected in supplied input is returned in response as separate field
- fileId - Allowed only for GET questionsParseReport endpoint. Defines ID of pre-loaded file (see Uploading input files) holding your input.

X-Auth-Token: <your_token>

Accept-Charset: <encoding> (ex. 'iso-8859-1', 'utf8'. UTF-8 is assumed if not specified)

Content-Type: multipart/form-data, where part with text should have text/csv content type

Accept: application/json or text/csv

- In application/json you get up to maxErrorsCount error entries. If not defined you get up to maximum supported by the platform now (currently = 1000)
- In text/csv you get ONLY error items in format <errorLineNumber><TAB><rawValueTakenFromFile>. maxErrorsCount is not considered in that case.

Body: Group query input

Response format

Content-Type: application/json or text/csv (depending on Accept header value). Body is encoded in UTF-8 by default

Response codes

- Status 200 - OK. Parse report is n body
- Status 404 - In case if file with supplied ID could not be found (for GET request)
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).

Getting extended Group Query Info

As an authorized user you can get extended information about your queries as list of Group Query Info objects. List is ordered from the most recent to most eldest. As far as Query info object encapsulates information about several related objects, it is in general a convenience method which should be used for efficient retrieval of information. Before using this method, please consider usage of separate resources for Queries and results as they are more efficient when you don't need extended information

Syntax

GET /groupSearch/queryInfo?

startIndex={index_value}&count={count_value}

&queryName={qName}

&queryType={qType}

&queryPostDateFrom={dateFrom}

&queryPostDateTo={dateTo}

&queryStatus={NOT_STARTED|IN_PROGRESS|COMPLETED}

Request Format

Parameters:

- startIndex, count (see Paged Reading)
- queryName - query name. There will be returned all the queries which name matches the given one. Non mandatory. Parameter supports wildcards (* and ?)
- queryPostDateFrom - minimal date of when the query was posted. Returned queries are older then or have the same date as defined by this parameter. Non mandatory
- queryPostDateTo - maximal date of when the query was posted. Returned queries are older then or have the same date as defined by this parameter. Non mandatory
- queryStatus - one or more statuses of queries which should be returned. If not specified - queries set is not filtered by the status.

X-Auth-Token: <your_token>

Content-Type: application/json

Accept-Charset: utf-8

Accept: application/json

Response format

Content-Type: application/json

Content-Range: <startIndex>-<endIndex>/<totalCount>

Body: Group Query Info

Response codes

- Status 200 - Ok. There is at least one answer matching given parameters.
- Status 204 - Ok. There are no answers for your query.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section).
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).

Getting your Group Query Input

As an authorized user you can get an input you have posted while runing the query. Only those rows which were recognized in input are present in result of this request. You are always getting addresses in form of application/octet-stream. Although format of response depends on value of **Accept** header you have supplied in request. Also you can request data in necessary encoding using 'Accept-charset' header or 'charset' parameter of 'Accept' header.

Syntax

GET /groupSearch/settingsInstance/{id}/queryInput

Request Format

X-Auth-Token: <your_token>

Accept-Charset: <encoding> (ex. 'iso-8859-1', 'utf8'. UTF-8 is assumed if not specified)

Accept :text/csv OR application/json OR application/zip OR application/gzip ('charset' parameter value is used if there is no Accept-Charset header)

With this header you specify which output format you prefer to get in response. Please note that if you have requested zip/gzip content than values will be returned in CSV format zipped/gzipped

Response format

Content-Type: depends on 'Accept' header value

Body: Stream in format defined by Accept header value.

Response codes

- Status 200 - OK. There is at least one address Filter entry associated with your settings.
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if settings with given ID does not exist.

Getting available Features List for Group Search

As an authorized for Group searches user you can get a list of special features if there are some available for your user. By default there is no any non-standard features available to Itesco clients, but f you have some specific requirements to input and.or output which you would like us to fulfil, you can contact Itesco Sales and do a feature request. If you have no special features endpoints below will always result empty responses.

You can get a list of available features or refer some specific one by ID. IDs of features are usually human readable names specified in UPPER_CASE. Down below you can find description of endpoints. Please note that features endpoint is capable to supply localized answers to your requests. That means that if you follow the Localization section you will be able to get values of feature fields in one of supported languages

Syntax

GET /groupSearch/features/{id}

GET /groupSearch/features[?startIndex={index_value}&count={count_value}&withInFeatures={true/false}&withOutFeatures={true/false}]

Request Format

Parameters:

- startIndex, count (see Paged Reading)
- withInFeatures (true by default) - specify as false if you don't want to get features for processing user input
- withOutFeatures (true by default) - specify as false if you don't want to get features for processing user output

X-Auth-Token: <your_token>

Accept: application/json

Accept-Charset: utf-8

Accept-Language: see Localization

Body: <no_body_expected>

Response format

Content-Type: application/json

Body: List of Features descriptions

Response codes

- Status 200 - OK. There is a result matching your request in the body.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section)
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if you have been issuing get by ID and instance with given ID does not exist.

Getting available Matching Templates

As an authorized for Group searches user you can get a list of the standard matching templates which are supported by iCatch3 application engine. Matching template represents an identifiable set of rules which are applied to each row of your input and information we have in database to decide whether it "matches" or not. Positive decision causes our information to be placed to result (as separate item by default or replacing existing information if running in "update" mode).

You can get a list of available templates or refer some specific one by ID. IDs of standard templates are usually 4 digits numbers. Down below you can find description of endpoints. After that comes more details about Standard templates and matching rules.

Syntax

GET /groupSearch/matchingTemplates/{id}

GET /groupSearch/matchingTemplates[?startIndex={index_value}&count={count_value}]

Request Format

Parameters:

- startIndex, count (see Paged Reading)

X-Auth-Token: <your_token>

Accept: application/json

Accept-Charset: utf-8

Body: <no_body_expected>

Response format

Content-Type: application/json

Body: Group Search Settings

Response codes

- Status 200 - OK. There is a result matching your request in the body.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section)
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if you have been issuing get by ID and instance with given ID does not exist.

Matching Templates List

In the table below you can find the complete set of standard matching templates available to be used. You can refer them by ID when create settings template or posting a group match query for execution. The table besides ID holds also the fields which are matched in scope of the template. We are applying different kinds of matching fo each of fields. These kinds are described in the separate section.

ID	FIRST_NAME	LAST_NAME	ORG_NAME	BIRTH_DATE	SSN_ORG_NUM	STREET	HOUSE_NUM		FLAT	ZIP_CODE	LOCATION	PHONE
1100	✓	✓			✓	✓	✓				✓	
1110	✓				✓	✓					✓	
1120	✓	✓		✓		✓	✓				✓	
1130		✓		✓		✓	✓				✓	
1200	✓	✓	✓		✓						✓	✓
1210		✓	✓		✓						✓	✓
1220	✓	✓	✓								✓	✓
1230		✓	✓								✓	✓
1240	✓	✓			✓						✓	✓
1250		✓			✓						✓	✓
1260	✓	✓	✓		✓							✓
1270	✓	✓	✓									✓
1280	✓	✓			✓							✓
1300			✓		✓	✓					✓	✓
1310			✓		✓							✓
1320			✓									✓
1330					✓							✓
1400	✓	✓				✓					✓	✓
1410	✓	✓									✓	✓

1420		✓				✓					✓	✓
1430		✓									✓	✓
1440	✓	✓										✓
1450		✓										✓
1500												✓
2000			✓		✓	✓	✓	✓	✓		✓	
2010			✓		✓	✓	✓	✓			✓	
2020			✓		✓	✓	✓				✓	
2030			✓			✓	✓	✓	✓		✓	
2040			✓			✓	✓	✓			✓	
2050			✓			✓	✓				✓	
2060			✓		✓						✓	
2070			✓			✓					✓	
2080			✓								✓	
2100			✓		✓							
2120			✓									
3000	✓	✓	✓		✓	✓					✓	
3010	✓	✓	✓			✓					✓	
3020	✓	✓			✓	✓					✓	
3030		✓	✓		✓	✓					✓	
3040		✓	✓			✓					✓	
3050		✓			✓	✓					✓	
3060	✓	✓	✓		✓						✓	
3070	✓	✓	✓								✓	
3100		✓	✓								✓	
3105	✓	✓			✓						✓	
3110		✓			✓						✓	
3120					✓						✓	
3200	✓	✓	✓		✓							
3210	✓	✓	✓									
3220	✓	✓			✓							
3230		✓	✓		✓							
3240		✓	✓									
3250		✓			✓							
3333					✓							
4000	✓	✓				✓	✓	✓	✓		✓	
4010	✓	✓				✓	✓	✓			✓	
4020	✓	✓				✓	✓				✓	
4030		✓				✓	✓	✓	✓		✓	
4040		✓				✓	✓	✓			✓	
4050		✓				✓	✓				✓	
4060	✓	✓				✓					✓	
4070	✓	✓									✓	

5000						✓	✓	✓	✓		✓	
5010						✓	✓	✓			✓	
5020						✓	✓				✓	
5030						✓						

Matching Rules for Templates

For each of the fields mentioned below we are applying one or more match operation to decide whether the value we have matches given one or not. Field is considered as matched if **any of match operations** gave positive result.

Following matches are available

- EXACT - Exact match of field value to information in our database.
- ALIAS - Alias match - not exact but quite near and used in every day live. Ex. if we use ALIAS match rule on "Tobbias" name we can have positive match for such word as "Tobbie".
- PHONETIC - Phonetically matches.

Field	Matchig Kinds
FIRST_NAME	EXACT, ALIAS, PHONETIC
LAST_NAME	EXACT, ALIAS, PHONETIC
ORG_NAME	EXACT, PHONETIC
BIRTH_DATE	EXACT
SSN_ORG_NUM	EXACT
STREET	EXACT, ALIAS, PHONETIC
HOUSE_NUM	EXACT
HOUSE_NUM_SUFFX	EXACT
FLAT_NUMBER	EXACT
ZIP_CODE	EXACT
LOCATION	EXACT, ALIAS, PHONETIC
PHONE	EXACT

NIX Queries

Getting information about your NIX queries

As an authorized user you can get information about NIX Phone queries you have posted. You can get all your settings, or filter by ID or name. The single resource is responsible for the querying.

Syntax

GET /nixSearch/queries[?startIndex={startIndex}&count={count}] - get all queries for user has issued.

GET /nixSearch/queries/{id} - get query with given ID

Request Format

Parameters:

- startIndex, count (see Paged reading)

X-Auth-Token: <your_token>

Accept: application/json

Accept-Charset: utf-8

Body: <no_body_expected>

Response format

Content-Type: application/json

Body: Collection of NIX Query

Content-Range: <startIndex>-<endIndex>/<totalCount>

In response you will not get associated input. Supplied input is associated with query settings and could be retrieved using separate resources:

Getting your NIX Query Input

Response codes

- Status 200 - OK. There is at least one result matching your request parameters in the body.
- Status 204 - Ok. There are no settings matching your request parameters.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section)
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if you have been issuing get by ID and query with given ID does not exist.

Posting a NIX query for execution

As an authorized user you can post a NIX phones query for execution. While posting the query you must supply the reference to settings which you would like to use. Additionally you can supply settings parameters which would override corresponding parameters within the settings.

> **Encodings**
> Please note that we do our best to detect encoding of content you have supplied in body (for queryInput part of request), but if we fail - there will be used ISO-8859-1. So if you see some issues with parsing or getting corrupt output when requesting uploaded results, please let us know and consider using ISO-8859-1 encoding for your input till we are working on the solution of the problem.

Syntax

POST /nixSearch/queries

POST /nixSearch/queries?inputFileId={fileId}[&inputFileName={fileName}]

Request Format

Parameters:

- inputFileId - ID of pre-loaded file (see Uploading input files) holding input for your query.
- inputFileName - name of file you post as input. Non Mandatory. Will be used when you'll try to download input. If not specified - will be generated.
- queryInput - this should be a name of a part holding your query input if you are posting input in scope of request. File name parameter will be taken in this case from 'filename' key supplied along with the part in content-disposition section

X-Auth-Token: <your_token>

Content-Type: multipart/form-data if you are uploading input in body; application/json if you are referring pre-loaded files

Accept: application/json

Accept-Charset: utf-8

Body: NIX Query Definition

Response format

Content-Type: application/json

Body: NIX Phones Query

In response you will not get associated input. Supplied input is associated with settings instance created automatically from your template and overrides (and mentioned in query object under settingsId field). They could be retrieved using separate resources:

Getting your NIX Query Input

Response codes

- Status 200 - OK. Posted query metadata is in the body. Metadata also contains assigned ID.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section).
- Status 404 - In case if file with defined ID does not exist for exclusions/address filters
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if settings with given ID is not present. In case if file with defined ID does not exist for query input.

Getting NIX query settings by query ID

As an authorized user you can get settings instance used with a specific query.

Syntax

GET /nixSearch/queries/{id}/settings

Request Format

X-Auth-Token: <your_token>

Accept: application/json

Accept-Charset: utf-8

Body: <no_body_expected>

Response format

Content-Type: application/json

Body: NIX Query Settings

Response codes

- Status 200 - OK. There is a result matching your request in the body.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section)
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if you have been issuing get by ID and query with given ID does not exist.

Getting NIX query settings by instance ID

As an authorized user you can get settings instance by its id.

Syntax

GET /nixSearch/settingsInstance/{id}

Request Format

X-Auth-Token: <your_token>

Accept: application/json

Accept-Charset: utf-8

Body: <no_body_expected>

Response format

Content-Type: application/json

Body: NIX Query Settings

Response codes

- Status 200 - OK. There is a result matching your request in the body.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section)
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if you have been issuing get by ID and instance with given ID does not exist.

Getting your NIX Query input

As an authorized user you can get an input you have posted while running the query. Only those rows which were recognized in input are present in result of this request. You are always getting addresses in form of application/octet-stream. Although format of response depends on value of **Accept** header you have supplied in request. Also you can request data in necessary encoding using 'Accept-charset' header or 'charset' parameter of 'Accept' header.

Syntax

GET /nixSearch/settingsInstance/{id}/queryInput

Request Format

X-Auth-Token: <your_token>

Accept-Charset: <encoding> (ex. 'iso-8859-1', 'utf8'. UTF-8 is assumed if not specified)

Accept :text/csv OR application/json OR application/zip OR application/gzip ('charset' parameter value is used if there is no Accept-Charset header)

With this header you specify which output format you prefer to get in response. Please note that if you have requested zip/gzip content than values will be returned in CSV format zipped/gzipped

Response format

Content-Type: depends on 'Accept' header value

Body: Stream in format defined by Accept header value.

Response codes

- Status 200 - OK. There is at least one address Filter entry associated with your settings.
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if settings with given ID does not exist.

Getting information about your NIX query results

As an authorized user you can get information about NIX Phones Search Results. You can get information about all available results or use request parameters in order to make more precise selection. In scope of this request you are not getting actual query results, just metadata describing results being produced. In order to find out how to get access to the results you should refer

- Getting all your NIX query results
- Getting your NIX query results in chunks

Syntax

GET /nixSearch/results - get all results metadata produced so far for your user

GET /nixSearch/results[?startIndex={index_value}&count={count_value}] - get all results metadata produced so far for your user

GET /nixSearch/results/{id} - get results metadata with given ID

GET /nixSearch/results/{queryId} - get results metadata with given ID

Request Format

Parameters:

- startIndex, count (see Paged reading)
- queryId - set this to id of your query to get corresponding results metadata

X-Auth-Token: <your_token>

Accept: application/json

Accept-Charset: utf-8

Body: <no_body_expected>

Response format

Content-Type: application/json

Body: Collection of NIX Query Result

Content-Range: <startIndex>-<endIndex>/<totalCount>

Response codes

- Status 200 - OK. There is at least one result matching your request parameters in the body.
- Status 204 - Ok. There are no results matching your request parameters.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section)
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if you have been issuing get by ID and result with given ID does not exist.

Getting all your NIX query results

As an authorized user you can get results of query which has been completed successfully. Results are returned as text/csv. Also you can request data in necessary encoding using 'Accept-charset' header or 'charset' parameter of 'Accept' header (like "Accept: text/csv;charset=iso-8859-1").

Syntax

GET /nixSearch/results/{id}/valuesStream[?combineWithInput={true/false}]

Request Format

Parameters**:**

- combineWithInput- non mandatory Boolean parameter. Define it to true if you would like to have output in CSV combined to your input. By default in answers you'll get only answers for valid questions in input.

X-Auth-Token: <your_token>

Accept : text/csv ('charset' parameter value is used if there is no Accept-Charset header)

Accept-Charset: <encoding> (ex. 'iso-8859-1', 'utf8'. UTF-8 is assumed if not specified)

Response format

Content-Type: text/csv

Body: Stream in format defined by Accept header value. For format refer NIX Query Result Value.

Response codes

- Status 200 - Ok. There is at least one answer produced for your query.
- Status 204 - Ok. There are no answers for your query.
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section).
- Status 404 - In case if results with given ID does not exist.
- Status 403 - In case if your result is locked.

Getting your NIX query results in chunks

As an authorized user you can get results of query which has been completed successfully. Results are returned directly within the body. This way of getting results differs from one described in Getting all your NIX query results because it allows you reducing output set. Using that method could be useful for organizing paged review of results produced in response to your query. You can also find useful using NIX Query Result's **answerCount** property for paging needs. Also you can get results in different formats. Returned data format depends on "Accept" header value.

Syntax

GET /nixSearch/results/{id}/values[?startIndex={index_value}&count={count_value}&transpose={true/false}&combineWithInput={true/false}]

Request Format

Parameters:

- startIndex, count (see Paged Reading)
- transpose - a CSV format related Boolean flag. If transposed = true, all the results related to a single question are returned in one line. See [transposed format link goes here]. Please note that this flag is ignored for JSON values.
- combineWithInput - non mandatory Boolean parameter. Define it to true if you would like to have output in CSV combined to your input. By default in answers you'll get only answers for valid questions in input

X-Auth-Token: <your_token>

Accept : text/csv or application/json

Accept-Charset: utf-8

Response format

Content-Type: text/csv or application/json (depending on accept header value)

Body: Collection of NIX Query Result Value constrained according to parameters of request

Response codes

- Status 200 - Ok. There is at least one answer produced for your query.
- Status 204 - Ok. There are no answers for your query.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section).
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if results with given ID does not exist.
- Status 403 - In case if your result is locked.

Updating NIX query results related attributes

Query results have a few parameters which affect consequent queries you issue. As an authorized user you are able to modify the attributes using API.

Syntax

PUT /nixSearch/results/{id}

Request Format

X-Auth-Token: <your_token>

Content-Type: application/json

Accept-Charset: utf-8

Accept: application/json

Body: NIX Query Result

If you supply ID different from one used in URL, you'll get an error. This is done to prevent from occasional modification of your settings. At the same time if you did not supply ID within the body - an error will not be issued.

Response format

Content-Type: application/json

Body: NIX Query Result

Response codes

- Status 200 - OK. Result is in response body.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section)
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if result with given ID is not present.
- Status 409 - In case of Conflict (see Handling conflicts)

Checking correctness of supplied Questions for Nix Search

As an authorized user you can check whether the input you are going to provide is able to be handled by the system. For that you should post it to a special URL in Tab Separated values format. In response you'll get NIX Input Check Result/NIX Query Parse error report which contains parse errors and information about recognized entries. When getting input parse report as text/csv you can define encoding in which report is going to be provided. Use 'Accept-Charset' header or 'charset' parameter of request for that.

Also if you have already posted a query for execution you can get a parse report on questions you have used if you know settings id used for posting your query. As far as the query input is a part of your query settings the parse report for input is extracted using nixSearch/settingsInstance resource (see syntax section).

> **Encodings**
> Please note that we do our best to detect encoding of content you have supplied in body, but if we fail - there will be used ISO-8859-1. So if you see some issues with parsing or getting corrupt output when requesting uploaded results, please let us know and consider using ISO-8859-1 encoding for your input till we are working on the solution of the problem.

Syntax

POST /nixSearch/questions[?maxErrorCount=<>]

GET /nixSearch/questionsParseReport?fileId={preloaded_file_id}[&maxErrorCount=<>]

GET /nixSearch/settingsInstance/{id}/queryInput/questionsParseReport

Request Format

Parameters:

- maxErrorsCount = maximum number of errors to put into response. Non mandatory. Total number of errors detected in supplied input is returned in response as separate field
- fileId - Allowed only for GET questionsParseReport endpoint. Defines ID of pre-loaded file (see Uploading input files) holding your input.

X-Auth-Token: <your_token>

Accept-Charset: <encoding> (ex. 'iso-8859-1', 'utf8'. UTF-8 is assumed if not specified)

Content-Type: multipart/form-data, where part with text should have text/csv content type

Accept: application/json or text/csv

- In application/json you get up to maxErrorsCount error entries. If not defined you get up to maximum supported by the platform now (currently = 1000)
- In text/csv you get ONLY error items in format <errorLineNumber><TAB><rawValueTakenFromFile>. maxErrorsCount is not considered in that case.

Body: Group query input

Response format

Content-Type: application/json or text/csv (depending on Accept header value). Body is encoded in UTF-8 by default

Body: NIX Input Check Result/NIX Query Parse error report

Response codes

- Status 200 - OK. Parse report is n body
- Status 404 - In case if file with supplied ID could not be found (for GET request)
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).

Getting extended NIX Query Info

As an authorized user you can get extended information about your queries as list of NIX Query Info objects. List is ordered from the most recent to most eldest. As far as Query info object encapsulates information about several related objects, it is in general a convenience method which should be used for efficient retrieval of information. Before using this method, please consider usage of separate resources for Queries and results as they are more efficient when you don't need extended information

Syntax

GET /nixSearch/queryInfo?

startIndex={index_value}&count={count_value}

&queryName={qName}

&queryType={qType}

&queryPostDateFrom={dateFrom}

&queryPostDateTo={dateTo}

&queryStatus={NOT_STARTED|IN_PROGRESS|COMPLETED}

Request Format

Parameters:

- startIndex, count (see Paged Reading)
- queryName - query name. There will be returned all the queries which name matches the given one. Non mandatory. Parameter supports wildcards (* and ?)
- queryPostDateFrom - minimal date of when the query was posted. Returned queries are older then or have the same date as defined by this parameter. Non mandatory
- queryPostDateTo - maximal date of when the query was posted. Returned queries are older then or have the same date as defined by this parameter. Non mandatory
- queryStatus - one or more statuses of queries which should be returned. If not specified - queries set is not filtered by the status.

X-Auth-Token: <your_token>

Content-Type: application/json

Accept-Charset: utf-8

Accept: application/json

Response format

Content-Type: application/json

Content-Range: <startIndex>-<endIndex>/<totalCount>

Body: NIX Query Info

Response codes

- Status 200 - Ok. There is at least one answer matching given parameters.
- Status 204 - Ok. There are no answers for your query.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section).
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).

User files storage

Uploading input files

Some of methods accepts files as parameters values (ex. Posting query accepts Exclusions and Address Filters). The files could be supplied in body of multipart requests. It is not always convenient, especially when client has strict limitations on access to users file system or user computer resources (like modern browsers) and this does not allows in convenient manner operating with user files. For such cases we have created ability to our users to store files on our server. You can upload file using exposed endpoint, get ID in response and then use this ID as parameter for operations defining inputs. You can also download files you have uploaded, delete unnecessary ones or get information about files your store currently holds. Please refer corresponding topics within the document to get more details on particular operations.

> **User space Quota**
> In order to allow you and other user efficiently use our services we provide we have to limit the available space for each user. At the moment the quota for each user is equal to ~100 Mb of available space. In order to make it easier to you keeping your storage clean and always ready for work we recommend
>
> 1. Uploading your files as ***temporary*** ones. Temporary files a kept on your store only limited period of time and are removed automatically by the system. At the moment max lifetime of temporary file is equal to **8h**.
> 2. Deleting unnecessary files right after you don't need them

Syntax

POST /files[?temporary=<true/false>]

Request Format

Parameters:

- temporary - defines whether the file you upload is temporary or not. Default is **false**.

X-Auth-Token: <your_token>

Content-Type: multipart/formdata (content is expected as **file** part)

Accept-Charset: utf-8

Accept: application/json

Response format

Content-Type: application/json

Body: File Info

Response codes

- Status 200 - Ok. File was successfully uploaded - you can refer body for more details on uploaded file.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section).
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 413 - File cannot be uploaded because it's size exceeds free space quota available to current user.

Getting information about files from your storage

You can always get information about files which are currently present at your store. The information could be requested for all the files or for the particular file identified by its ID.

Syntax

GET /files/info - get information about files of user issuing request

GET /files/info/{id} - get info about the file with given ID

Request Format

X-Auth-Token: <your_token>

Accept: application/json

Accept-Charset: utf-8

Body: <no_body_expected>

Response format

Content-Type: application/json

Body: Collection of File Info objects

Note that in return we will not issue multipart response. Only information about file in json format will be returned. For accessing file content please use separate method:

- Downloading files from your storage

Response codes

- Status 200 - Ok. There is at least one result matching your request parameters in the body.
- Status 204 - Ok. There are no files matching your request parameters.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section)
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if you have been issuing get by ID and file with given ID does not exist.

Downloading files from your storage

You can download file currently present at your store. For that you need to request it from described endpoint by specifying its ID.

Syntax

GET /files/{id} - get content of file with given ID

Request Format

X-Auth-Token: <your_token>

Accept: application/octet-stream

Body: <no_body_expected>

Response format

Content-Type: application/octet-stream

Body: Content of file you have requested

Response codes

- Status 200 - Ok. There is at least one result matching your request parameters in the body.
- Status 204 - Ok. Your file is empty.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section)
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).
- Status 404 - In case if we are not able to locate file with supplied ID within the store of user issuing request.

Deleting files from your storage

As an authorized user you can delete previously stored files from your store. Result of operation is determined by returned status code.

Syntax

DELETE /files/{id}

Request Format

X-Auth-Token: <your_token>

Accept: application/json

Accept-Charset: utf-8

Body: <no_body_expected>

Response format

Content-Type: application/json

In case of error content type could be application/json with error in body

Body: <no_body_expected>

Response codes

- Status 204 - OK. File was located and successfully deleted.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section).
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).

- Status 404 - In case if file with given ID does not exist or was already deleted.

Geographical Dictionary

As far as service is accepting some parameters in terms of geographical items like zip/area codes, counties, municipalities, etc, it is often useful to be sure that client supplies correct terms while filtering. For such a purpose there are provided geographical dictionary which consists of several endpoints allowing you to validate your search terms. It is often the case to lookup your term by pattern and ensure that it is supplied to the search criteria accepting endpoints exactly as it was returned by the service. This does not relates to the case of terms, but is still true for terms content.

Please note that endpoints often support parameters called ***pattern**. Such kind of parameters can accept 2 wildcard characters: *(any number of any legal chars) and ? (any legal character). You can use them in order to lookup for terms which spelling you are not quite sure about. All patterns are non mandatory, so if you did not specified it it will be considered to be '*' by default

Within the subsections you will be able to find information about all available endpoints.

Getting information about zip codes

You can get information about geographical areas related to a specific zip code.

Syntax

GET dictionary/zips?

startIndex={index_value}&count={count_value}

&zipCodePattern={zipPattern}

Request Format

Parameters:

- startIndex, count (see Paged Reading)
- zipCodePattern - pattern used during matching. Can hold digits and wildcards (*, ?)

X-Auth-Token: <your_token>

Accept: application/json

Accept-Charset: utf-8

Body: <no_body_expected>

Response format

Content-Type: application/json

Body: Collection of Zip Info objects

Response codes

- Status 200 - Ok. There is at least one result matching your request parameters in the body.
- Status 204 - Ok. There are no items matching your request parameters.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section)
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).

Getting information about area codes

You can get information about area codes you can use in your queries.

Syntax

GET dictionary/areaCodes?

startIndex={index_value}&count={count_value}

&areaCodePattern={pattern}

&topLevel={true/false}

&withGeoReflection={true/false}

Request Format

Parameters:

- startIndex, count (see Paged Reading)
- areaCodePattern - pattern used during matching. Can hold digits and wildcards (*, ?)
- topLevel - accepts `true` or `false` (If not specified, defaults to `true`)
- withGeoReflection - accepts `true` or `false` (If not specified, defaults to `false`)

X-Auth-Token: <your_token>

Accept: application/json

Accept-Charset: utf-8

Body: <no_body_expected>

Response format

Content-Type: application/json

Body: Collection of Strings. Each item is matching area code

Response codes

- Status 200 - Ok. There is at least one result matching your request parameters in the body.
- Status 204 - Ok. There are no items matching your request parameters.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section)
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).

Getting information about Zip cities

You can get information about zip cities (postal areas) you can use in your queries.

Syntax

GET dictionary/zipCities?

startIndex={index_value}&count={count_value}

&namePattern={pattern}

Request Format

Parameters:

- startIndex, count (see Paged Reading)
- namePattern - pattern used during matching. Can hold letters, spaces, hyphens and wildcards (*, ?)

X-Auth-Token: <your_token>

Accept: application/json

Accept-Charset: utf-8

Body: <no_body_expected>

Response format

Content-Type: application/json

Body: Collection of Strings. Each item is matching zip city name

Response codes

- Status 200 - Ok. There is at least one result matching your request parameters in the body.
- Status 204 - Ok. There are no items matching your request parameters.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section)
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).

Getting information about Zip cities by Municipality

You can get information about zip cities (postal areas) you can use in your queries.

Syntax

GET dictionary/municipality2zipCity?

namePattern={pattern}

Request Format

Parameters:

- namePattern - pattern used during matching. Can hold letters, spaces, hyphens and wildcards (*, ?)

X-Auth-Token: <your_token>

Accept: application/json

Accept-Charset: utf-8

Body: <no_body_expected>

Response format

Content-Type: application/json

Body: Collection of Strings. Each item is matching zip city name

Response codes

- Status 200 - Ok. There is at least one result matching your request parameters in the body.
- Status 204 - Ok. There are no items matching your request parameters.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section)
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).

Getting information about Counties and Municipalities

You can get information about counties and their municipalities.

Syntax

GET dictionary/counties?

&countyNamePattern={pattern}

&municipalityNamePattern={pattern}

Request Format

Parameters:

- countyNamePattern - pattern used during matching for names of counties. Can hold letters and wildcards (*, ?). You can define several patterns by supplying several parameters with the same name. in this case entry will be considered a matching one in case if it matches at least one of supplied patterns
- municipalityNamePattern - pattern used during matching for names of municipalities. Can hold letters and wildcards (*, ?). You can define several patterns by supplying several parameters with the same name. in this case entry will be considered a matching one in case if it matches at least one of supplied patterns.

  Please note if this one is used together with countyNamePattern - then you'll get:
  - all counties which are matching county name patterns. All municipalities within matching counties are "filtered" through given municipalities patterns
  - all counties which have at least one municipality matching to at least one municipality name patterns supplied

X-Auth-Token: <your_token>

Accept: application/json

Accept-Charset: utf-8

Body: <no_body_expected>

Response format

Content-Type: application/json

Body: Collection of County Info objects

Response codes

- Status 200 - Ok. There is at least one result matching your request parameters in the body.
- Status 204 - Ok. There are no items matching your request parameters.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section)
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).

SNI Codes Dictionary

As far as service is accepting SNI codes as a prt of filtering conditions, it is often useful to be sure that client supplies correct terms while filtering. For such a purpose there are provided SNI codes dictionary which is exposed by a single endpoint allowing you to validate your search terms. SNI codes could be looked up by pattern which can hold either part/complete SNI code value or part/complete SNI code description. Dictionary works on 2007 year standard.

Please note that endpoints often support parameters called ***pattern**. Such kind of parameters can accept 2 wildcard characters: *(any number of any legal chars) and ? (any legal character). You can use them in order to lookup for terms which spelling you are not quite sure about. All patterns are non mandatory, so if you did not specified it it will be considered to be '*' by default.

Getting information about SNI codes

You can get information about SNI codes corresponding to SNI 2007 standard using described endpoint. For searching you can use either SNI code value pattern or SNI code description pattern. Please note that returned collection of SNI Codes info holds description in the language specified by **Accept-Language** header. The value of header is also used to determine which dictionary to query your pattern to.

Syntax

GET dictionary/sniCodes?

startIndex={index_value}&count={count_value}

&sniCodesPattern={pattern}

Request Format

Parameters:

- startIndex, count (see Paged Reading)
- sniCodesPattern- pattern used during matching. Can hold letters, spaces, hyphens and wildcards (*, ?)

X-Auth-Token: <your_token>

Accept: application/json

Accept-Charset: utf-8

Accept-Language: EN/SE (If not defined EN is used)

Body: <no_body_expected>

Response format

Content-Type: application/json

Body: Collection of Sni Code Info. Each item is matching given constraints

Response codes

- Status 200 - Ok. There is at least one result matching your request parameters in the body.
- Status 204 - Ok. There are no items matching your request parameters.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section)
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).

Locking Query results

Admin users can prevent some results from being extracted. This is called 'locking' of results. Locked results cannot be extracted through the service API. Instead of supplying results there will be returned error response with status 403 and special sub code '403001'. Results could be locked/unlocked by putting locked attribute boolean value to corresponding endpoint.

Syntax

PUT /admin/privateProspects/results/{id}/locked

PUT /admin/companyProspects/results/{id}/locked

Request Format

X-Auth-Token: <your_token>

Accept: application/json

Accept-Charset: utf-8

Body: String value for true/false

Response format

Content-Type: application/json

Body: Actual state of your Result object

Response codes

- Status 200 - Ok. Updated successfully.
- Status 400 - Malformed request. (sub-codes are described in Common Error Codes section).
- Status 404 - Result with given id does not exist.
- Status 401, 412 - Authorization issues (sub-codes are described in Common Error Codes section).

Change Log

1.0 - 2015-12-02

Initial Release of ic3 REST API with Person Search implementation.

- Authentication endpoints added
- Dictionary data endpoints exposed
- Queries management API exposed
- User data files API endpoints exposed
- Persons search API endpoints exposed

1.1 - 2016-08-26

Implementation of Companies Search API.

- Companies Search API endpoints exposed
- Dictionary endpoints extended with SNI codes dictionaries
- Locking results ability added
- /ping endpoint for accessibilty check added

1.2 - 2017-05-04

Implementation of Group Search API.

- Group Search API endpoints exposed
- Demo user Access support
- Exclusions now allow excluding people living in the same building as ones get excluded

1.3 - 2017-09-22

Implementation of NIX registry search by phones

- NIX registry search API endpoints exposed

1.3.1 - 2017-10-25

Minor updates on stability and functionality:

- Updated functionality of /ping endpoint. Now it allows tracking version of API.
- CFAR identifier is added to Company selection results
- In company query settings count related criteria are limited by range [0..1 000 000] instead of previously Java Specific max value of Integer (2^{31} -1)
- Added option for getting NIX answers in CSV set according to input. Parameter 'outNonParsed' is added to /nixSearch/results/{id}/valuesStream endpoint

1.3.2 - 2017-12-01:

- Added duplicate input items count to NIX Input Check Result

1.4 - 2017-12-08:

- Added Flat Number support. Changed Data Structures
  - Group Search Questions
  - Private Prospects Query Result Value
  - Company Prospects Query Result Value / Company Prospects Flat Result Value
  - Address Criteria List
- Added Flat number considering Matching templates. See: Matching Templates List
- Added totalDuplicates field to Nix Query Input
- For NIX queries parameter 'outNonParsed' was renamed to 'combineWithInput' as it better reflects what is done. Details:
  - Getting all your NIX query results

1.5 - 2018-03-09

- Added Quick Search API
  - Doing Quick Search
  - Quick Search Results
  - Quick Search

1.5.1 - 2019-11-13

Minor updates on functionality:

- Added option to retrieve only fixed Area Codes. Parameter 'topLevel' is added to /dictionary/areaCodes endpoint
- Added option to retrieve only Area Codes with Geo Reflection (zip-ranges available). Parameter 'withGeoReflection' is added to /dictionary/areaCodes endpoint

1.5.2 - 2019-12-19

Added new end point for getting zip cities with ability to group by municipalities.

- Zip City by Municipality Info

1.6.0 - 2020-03-01

- Authentication Result is extended with `token` field which holds JWT token, one should be used to access services in further releases