Specification for the Trafiklab Key API version 1

Specification v1.12, Andreas Krohn, 2018-03-02

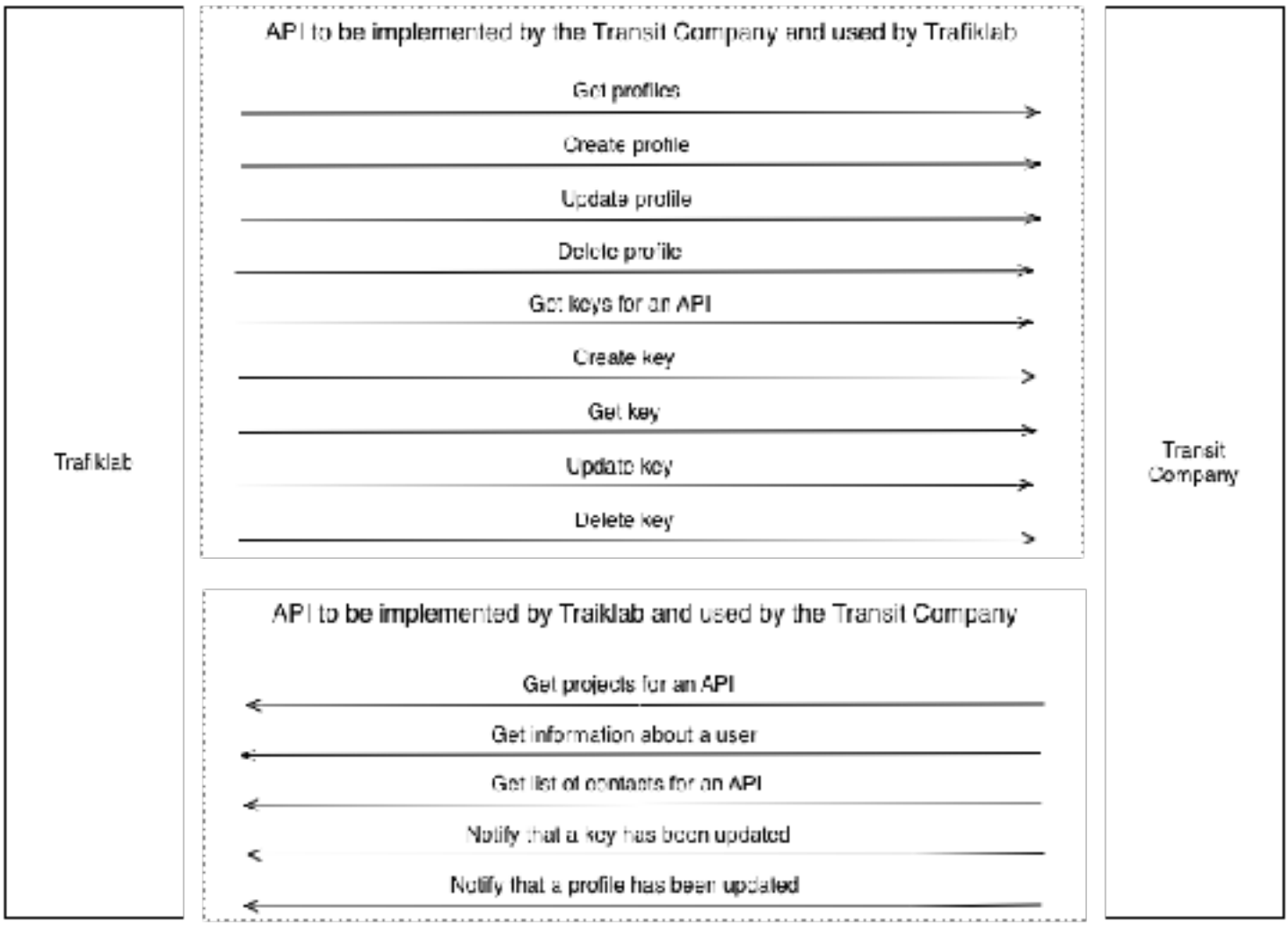
Version	Date	Change	By
1.7	2014-02-27	-	Andreas Krohn
1.8	2014-04-10	• Skickar inte med profil ID eller API i bodyn när man skapar en ny profil (API skickas med i URL, ID ska bestämmas av Trafikföretagen) • Förtydligade att det är Trafikföretagens ansvar att sätta ID på en ny profil • Skickar inte med profil ID i bodyn när man uppdaterar en profil (skickas redan med i URLen) • API skickas inte med när man uppdaterar en profil eftersom detta värde inte kan ändras • Projektstatus är nu en string array istället för en string eftersom ett projekt kan ha flera statusar samtidigt. • "Active" flaggan skickas ej med när man uppdaterar en nyckel. Radera metoden sätter active till false och uppdatera metoden kan ej återaktivera en inaktiverad nyckel. • Korrigera profile ID i alla exempel för att göra dokumentet mer konsekvent	Andreas Krohn
1.9	2014-05-26	• Korrigerar så att projektstatus verkligen är en Array of Strings i alla exempel och förklaringar.	Andreas Krohn
1.10	2015-01-14	• Nytt felmeddelande returneras när man försöker skapa en nyckel för ett projekt som redan har en nyckel	Andreas Krohn
1.11	2015-03-04	• Translation from Swedish to English. The document will only be maintained and updated in its English version from now on.	Andreas Krohn
1.12	2018-03-02	• Adapted the Key API to GDPR by not sending any data about individuals and require that deleted data is really deleted • Delete a key should really delete a key and all related data • Removed the method "Get contact list for an API" • Removed user name and email from method listing projects for an API and from the method getting information about a user • Removed user name and email from all key methods	Andreas Krohn

Introduction

This is a specification of the Key API that Trafiklab uses to create and maintain API keys for connected APIs as well as the Key API that Transit Companies can use to retrieve information from Trafiklab about the users of their API. The Key API is divided into two parts:

- **The Transit Company Key API** - this API should be implemented by the Transit Companies that connect their APIs to Trafiklab. Trafiklab will be using this API to enable the Trafiklab users to create API Keys for the Transit Companies APIs as well as to do some basic administration of the API Keys.
- **The Trafiklab Key API** - this API should be implemented by Trafikalb and should be used by the Transit Companies that distribute their APIs via Trafiklab to retrieve information about the users of their APIs.

Through these two APIs Trafiklab can get the information needed for the Trafiklab users from the Transit Companies as well as manage the keys if need be. The Transit Companies can get information about how their APIs are used by the Trafiklab users as well as manage the API Keys.



Basic Concepts

- **API** - a separate Transit Company API identified byt a unique *API ID* (string) that is defined by Trafiklab. APIs are not created via an API call but are instead created manually in the backend.
- **Profile** - a profile belongs to an API and keys belong to a profile. A profile contains information about the rate limits that are enforced for the keys belonging to that profile. There can be one, and only one, default profile for each API. This profile is automatically used for new keys. Identified with a *Profile ID* (string) that is unique within the API it belongs to.
- **Key** - a key belongs to a profile and an API and is used to limit the amount of API requests a users can make. Note that the only thing limited is the amount of requests and not what API method or data that the user has access to. A key is identified with *KEY* (string) that is unique. Valid characters for a key are A-Za-z0-9.
- **Project** - a project is an application or similar that is using one or more of the APIs provided by Trafiklab. One or more users are connected to a project. Identified with a *PROJECT ID* (string) that is unique. Note that all information about a project is stored at Trafiklab and can be updated at any point in time. Information about projects are sent to Transit Companies to inform them about how their APIs are used.
- **User** - a person registered at Trafiklab. Can be connected to several projects. Identified by a *USER ID* (string) that is unique.

The master information about profiles and keys are stored at the Transit Companies and it is the responsibility of Trafiklab to keep any copies of this information up to date. The master information about projects and users is stored at Trafiklab and it is the responsibility of the Transit Companies to keep any copies of this information up to date.

Authentication

All API requests require HTTPS as well as an authenticated user. Transit Companies should provide Trafiklab with a correct username and password to use when calling the API.

Basic authentication is used and that requires the client to send an Authorization HTTP header in each request. The value of the Authorisation header is constructed as follows:

1. Username ad password combines in a string "username:password".
2. The string is encoded in Base64.
3. Authentication method and a space, ie "Basic ", is added in front of the encoded string.

Example: A user with the username 'TheUsername' and the password 'ThePassword' results in the following HTTP Header…
Authorization:Basic VGhlVXNlcm5hbWU6VGhlUGFzc3dvcmQ=

Anyone with a valid username/password has the right to create/manage profiles for one or more APIs.

Response & Error Handling

All methods return a response wrapper with the following attributes:

Attribute	Type	Description
StatusCode	integer	Status code unique to the Transit Company and it's API implementation. Used for debugging etc.
Message	string	Optional error message (empty or missing if no error has occurred).
ExecutionTime	integer	The execution time of the method in ms. Optional field.
ResponseData	objekt	The methods response data, might be empty of an error has occurred.

Example:

```
{
      "StatusCode" : 123,
      "Message" : "",
      "ExecutionTime" : 72,
      "ResponseData" : {
         …
      }
}
```

All examples for all methods in this document just show the data that is included in the "ResponseData" object.

If anything goes wrong an error message should be returned in the "Message" attribute as well as using the correct HTTP status code for the response.

This is a list of the errors all methods must implement:

Error	HTTP Status Code	Error Message
HTTPS is not used	403	HTTPS Required
Incorrect or missing Authorization header	401	Valid authorization header required

Note that this is not a complete list of all the possible error messages. It is just a list of required error messages.

Attributes

The orders of all attributes that are sent in API requests or returned in HTTP bodies can vary. The order in this document is just an example.

All dates should be in the ISO 8601 format, for example “2013-09-19T11:29:17.828Z”

All attributes and data used should be UTF-8 encoded.

Transit Company Key API

This API should be implemented by Transit Companies that are using Trafiklab to distribute their APIs. All API methods has an relative URL. Which domain and port to use should be specified by the Transit Company that implements the API.

Function	HTTP Method	Relativ URL
Get all profiles for an API	GET	/trafiklab/v1/apikeys/apis/{API ID}/profiles
Create a new profile for an API	POST	/trafiklab/v1/apikeys/apis/{API ID}/profiles
Update a profile	PUT	/trafiklab/v1/apikeys/profiles/{Profile ID}
Delete a profile	DELETE	/trafiklab/v1/apikeys/profiles/{Profile ID}
Get all keys for an API	GET	/trafiklab/v1/apikeys/apis/{API ID}/keys
Create a new key for an API	POST	/trafiklab/v1/apikeys/apis/{API ID}/keys
Get a key	GET	/trafiklab/v1/apikeys/keys/{KEY}
Update a key	PUT	/trafiklab/v1/apikeys/keys/{KEY}
Delete a key	DELETE	/trafiklab/v1/apikeys/keys/{KEY}

Error Handling

The following errors are specific for the Transit Company Key API:

Error	HTTP Status Code	Error message
Default profile is missing when a new key is to be created	500	Unable to create a key since no default profile is set for the API
Incorrectly formated body when one tries to create or update a key or profile.	400	Incorrectly formatted body
The HTTP Header Content-type is not set to "application/json" when one tries to create/update a profile or a key	400	Content-type header not set to application/json
Tries to craete a new profile with a non-unique Id	409	A profile with the specified Name already exists for the API
Tries to create a new profile that is not connected to any existing API.	400	A profile needs to be connected to a valId API

Error	HTTP Status Code	Error message
Tries to create a new key that is not connected to any existing API.	400	A key needs to be connected to a valid API
Tries to delete the default profile	400	Can not delete the default profile
Tries to delete a profile that has more than zero API keys connected to it.	400	Can not delete a profile with attached API keys
Tries to create a new key connected to a project that already has a key for the specified API. Only one key/API/ project is allowed.	400	A key already exists for the project. Key: "[INSERT THE ALREADY EXISTING KEY]"

Profiles

All profile methods are using the same data model and all fields are required unless otherwise specified.

Attribute	Type	Description
Id	string	An ID that is unique among all the Transit Companies profiles (i.e. not only the ones connected to one API). It is the responsibility of the Transit Company to assign every new profile a unique ID. This attribute is not included in the body when one creates or updates a profile.
Name	string	A descriptive name of the profile
Api	string	API ID for the API the profile belongs to. This attribute is not included when one creates a new profile.
RateLimit/ Month	integer	The number of API requests allowed to be made per 30 day period for any API key connected to this profile.
RateLimit/ Minute	integer	The number of API requests allowed to be made per 60 second period for any API key connected to this profile.
Default	boolean	Specifies if a profile is the default profile for the API. Only a profile at a time can be the default profile. Note that this attribute is optional when updating a profile and it is recommended that "default" is only included when one really wants to change the value of this attribute.

Attribute	Type	Description
CreatedDate	date	The date the profile was created
UpdatedDate	date	The date the profile last was updated

Example:

```
    {
        "Id": "TrafiklabExportAPI_Gold",
        "Name": "Guldnivån",
        "Api": "TrafiklabExportAPI",
        "RateLimit": {
            "Month": 10000,
            "Minute": 15
        },
        "Default": false,
        "CreatedDate": "2013-09-19T11:29:17.828Z",
        "UpdatedDate": "2013-09-19T11:29:17.828Z"
    }
```

The Default Profile

All APIs have one and only one default profile. When a new profile gets "Default" set to "true" the old default profile automatically gets "Default" set to "false". When a new key is created it automatically belongs to the default profile. To change so that the new key belongs to another profile one has to update the key in a separate API call. It is not possible to create a key for an API unless a default profile is set.

Get all profiles for an API

HTTP Method: GET
URL: /trafiklab/v1/apikeys/apis/{API ID}/profiles
Returns an array of profiles for the specified API.

Example:

```
GET /trafiklab/v1/apikeys/apis/TrafiklabExportAPI/profiles

ResponseData:
[
    {
        "Id": "Profile1",
        "Name": "Silver",
        "Api": "TrafiklabExportAPI",
        "RateLimit": {
            "Month": 5000,
            "Minute": 10
        },
        "Default": true,
        "CreatedDate": "2013-09-19T11:29:17.828Z",
        "UpdatedDate": "2013-09-19T11:29:17.828Z"
    },
```

```
    {
        "Id": "ProfileB",
        "Name": "Gold",
        "Api": "TrafiklabExportAPI",
        "RateLimit": {
            "Month": 10000,
            "Minute": 15
        },
        "Default": false,
        "CreatedDate": "2013-09-19T11:29:17.828Z",
        "UpdatedDate": "2013-09-19T11:29:17.828Z"
    }
]
```

Create a new profile for an API
HTTP Method: POST
URL: /trafiklab/v1/apikeys/apis/{API ID}/profiles
The body has to contain a complete JSON object for the new profile. The HTTP Header Content-Type must be set to "application/json". The method returns the newly created profile.

Example:

```
POST /trafiklab/v1/apikeys/apis/TrafiklabExportAPI/profiles
{
    "Name": "Platina",
    "RateLimit": {
        "Month": 200000,
        "Minute": 100
    },
    "Default": false
}
```

Update a profile
HTTP Method: PUT
URL: /trafiklab/v1/apikeys/profiles/{Profile ID}
The body must contain a complete JSON object for the new profile (Id and API cannot be updated for a profile and should not be included in the body). The HTTP Header Content-Type must be set to "application/json". The method returns the updated profile.

Example:

```
PUT /trafiklab/v1/apikeys/profiles/TrafiklabExportAPI_Platinum
{
    "Name": "Updated platina level",
    "RateLimit": {
        "Month": 200001,
        "Minute": 102
    }
}
```

Delete a profile
HTTP Method: DELETE
URL: /trafiklab/v1/apikeys/profiles/{Profile ID}
Only profiles without any keys can be deleted, with the exception of the default profile that can never be deleted.

Keys

All key methods use the following data model where all fields are required unless otherwise specified. For a few methods that data model for a project is expanded as listed below.

Attribute	Type	Description
Key	string	Unique key
Note	string	A note used to remember what operations have been performed on the key. Overwritten when updated so take care to include any old information when the key is updated (preferably separate new and old note rows with a new line character and start each note row with the date the changes was made).
Api	string	The API ID of the API the key belongs to.
Profile	string	The Profile ID for the profile the key belongs to. Do not include it when a new key is created since a new key always belongs to the default profile.
Project	string	Project ID for the project the APIs should be used for. Note that this attribute is not used for the create/update methods since much more project information is then included (see below).
CreatedDate	date	Date when the key was created
UpdatedDate	date	Date when they key was last updated
Active	boolean	Specify if the key is active or not. Only active keys can be used to make API requests to the Transit Companies APIs. Not included when creating new keys since new keys are always active.

In addition to these attributes the methods Create and Update contain the following information:

Attriute	Type	Description
Project/Id	string	Profile ID for the profile the key belongs to.
Project/Name	string	The name of the project the API is used for.
Project/Status	array of strings	Valid values are "Test", "Ongoing", "Launched", "Terminated". This is an optional field and it can have several values in the array.

Attriute	Type	Description
Project/ ShortDescription	string	Short description of the project. Might contain HTML tags.
Project/ LongDescription	string	Long description of the project. Might contain HTML tags.
Project/Users	array	All users connected to this project
Project/Users/Id	string	The User ID of the user

Get all keys for an API
HTTP Method: GET
URL: /trafiklab/v1/apikeys/apis/{API ID}/keys
Returns an array of keys for the specified API.

Example:
```
GET /trafiklab/v1/apikeys/apis/TrafiklabExportAPI/keys

ResponseData:
[
     {
          "Key": "730a655dd2ae44bb94c9c244a01cca2b",
          "Note": "",
          "Api": "TrafiklabExportAPI",
          "Profile": "ProfileB",
          "Project": "1234",
          "CreatedDate": "2013-09-19T11:29:17.828Z",
          "UpdatedDate": "2013-09-19T11:29:17.828Z",
          "Active": true
     },
     {
          "Key": "810a655dd2ae44bb94c9c244a01cca3a",
          "Note": "",
          "Api": "TrafiklabExportAPI",
          "Profile": "Profile1",
          "Project" : "12937123",
          "CreatedDate": "2013-09-19T11:29:17.828Z",
          "UpdatedDate": "2013-09-19T11:29:17.828Z",
          "Active": true
     }
]
```

Create a new key for anAPI
HTTP Method: POST
URL: /trafiklab/v1/apikeys/apis/{API ID}/keys
The body must contain a JSON object for the new key, not including the key, API and dates. Information about what project the key belongs to and what users that are connected to that project is included in order to inform the Transit Company who is using the key. The HTTP Header Content-Type has to be set to "application/json".

The method returns a newly created key.

Example:

```
POST /trafiklab/v1/apikeys/apis/TrafiklabExportAPI/keys
{
    "Note": "2014-01-01: Key created",
    "Project" : {
        "Id" : "23134",
        "Name" : "New cool app",
        "Status" : ["Test", "Terminated"],
        "ShortDescription" : "We do stuff",
        "LongDescription" : "We do a lot of stuff",
        "Users" : [
           {
              "Id" : "87987"
           },
           {
              "Id" : "42"
           }
        ]
     },
}

ResponseData:
{
    "Key": "730a655dd2ae44bb94c9c244a01cca2b",
    "Note": "2014-01-01: Skapats",
    "Api": "TrafiklabExportAPI",
    "Profile": "Silver",
    "CreatedDate": "2014-01-01T11:29:17.828Z",
    "UpdatedDate": "2014-01-01T11:29:17.828Z",
    "Active": true
}
```

Get a key

HTTP Method: GET
URL: /trafiklab/v1/apikeys/keys/{key}

Example:

```
GET /trafiklab/v1/apikeys/keys/730a655dd2ae44bb94c9c244a01cca2b

ResponseData:
{
    "Key": "730a655dd2ae44bb94c9c244a01cca2b",
    "Note": "2014-01-01: Created, 2014-02-01: Updated to Silver",
    "Api": "TrafiklabExportAPI",
    "Profile": "123",
    "CreatedDate": "2013-09-19T11:29:17.828Z",
    "UpdatedDate": "2013-09-19T11:29:17.828Z",
    "Active": true
```

```
}
```

Update a key

HTTP Method: PUT
URL: /trafiklab/v1/apikeys/keys/{key}
The body must contain a JSON object for the updated key, not including key, API, active and dates. The HTTP Header Content-Type must be set to "application/json". The method returns the updated key.

Example:

```
PUT /trafiklab/v1/apikeys/keys/730a655dd2ae44bb94c9c244a01cca2b
{
    "Note": "",
    "Profile": "Silver"
}
```

Delete a key

HTTP Method: DELETE
URL: /trafiklab/v1/apikeys/keys/{key}

Completely deletes a key and all references to the key from the system. If successful returns HTTP Status Code 200 OK and an empty ResponseData object.

Previously this method did not delete a key from the system, instead it was is a quick way to set "Active" to "false" for a key. Note that this is no longer the case and a deleted key has to be completely removed from the system.

Example:

```
DELETE /trafiklab/v1/apikeys/keys/730a655dd2ae44bb94c9c244a01cca2b

ResponseData:
{}
```

Trafiklab Key API

This API should be implemented by Trafiklab and used by the Transit Company. All API methods have a relative URL, the domain is https://trafiklab.se. Since the master version of the user information is stored by Trafiklab it is the responsibility of the Transit Company to retrieve updated user information when needed.

If a user has deleted their information at Trafiklab or has disconnected from a project this deleted information is no longer returned by this API. In that case the Transit Company has to delete all information saved in connection to the deleted user.

Function	HTTP Method	Relative URL
Get all projects for an API	GET	/system/v1/apikey/apis/{API ID}/Projects
Get information about a user	GET	/system/v1/apikeys/users/{USER ID}
Get the contact list for an API	GET	/system/v1/apikeys/apis/{API ID}/contacts
Inform Trafiklab that an API key has been updated.	GET	/system/v1/apikeys/keys/{KEY}/refresh
Inform Trafiklab that a profile has been updated.	GET	/system/v1/apikeys/apis/{API ID}/profiles/refresh

Error Handling

The following errors are specific to the Trafiklab Key API:

Error	HTTP Status Code	Error Message
Tries to refresh a non-existing API	400	The API does not exist
Tries to refresh a non-existing key	400	The key does not exist
Tries to refresh a key that does not belong to the specified API.	400	The key is not connected to the specified API
The refresh method failed to get updated key information from the Transit Company.	500	Refresh failed
The Transit Company tries to get information about a user that is not using any of their APIs.	403	Access not allowed to the provided user

Error	HTTP Status Code	Error Message
The Transit Company tries to get information about another Transit Companies API.	403	Access not allowed to the provided API

Get all projects for an API

HTTP Method: GET
URL: /system/v1/apikey/apis/{API ID}/Projects
Returns all projects for the specified API, if the specified API belongs to the requesting Transit Company.

Projects are described with the following data model where all attributes are required.

Attibute	Type	Description
Projects/Name	string	Profile ID for the profile the key belongs to.
Projects/Id	string	The name of the project the API is used for.
Project/Status	array of strings	Valid values are "Test", "Ongoing", "Launched", "Terminated". This is an optional field and it can have several values in the array.
Project/ ShortDescription	string	Short description of the project. Might contain HTML tags.
Project/ LongDescription	string	Long description of the project. Might contain HTML tags.
Projects/Users/Id	string	The User ID of the user

Example:

```
GET /system/v1/apikeys/apis/Realtid/Projects

ResponseData:
{
    "Projects" : [
        {
           "Id" : "23134",
           "Name" : "New cool app",
           "Status" : ["Test"],
           "ShortDescription" : "We do stuff",
           "LongDescription" : "We do a lot of stuff",
           "Users" : [
              {
                 "Id" : "87987"
              },
              {
```

```
                "Id" : "42"
            }
         ]
      },
      {
         "Id" : "787",
         "Name" : "Another cool app",
         "Status" : ["Ongoing","Launched"],
         "ShortDescription" : "We do stuff",
         "LongDescription" : "We do a lot of stuff",
         "Users" : [
            {
               "Id" : "87987"
           }
         ]
      }
   ]
}
```


Get information about a user

HTTP Method: GET
URL: /system/v1/apikeys/users/{USER ID}
Returns information about the specified user. To be used if an individual user has to be contacted.

Users are described with the following data model where all attributes are required.

Attribute	Type	Description
User/Projects	array	Array of all the projects the user is connected to, including only the projects that uses at least one of the Transit Company APIs. Returns an empty array if the user is not connected to any projects that uses at least one of the Transit Company APIs.
User/Projects/ Name	string	Profile ID for the profile the key belongs to.
User/Projects/Id	string	The name of the project the API is used for.
User/Project/ Status	array of strings	Valid values are "Test", "Ongoing", "Launched", "Terminated". This is an optional field and it can have several values in the array.
User/Project/ ShortDescription	string	Short description of the project. Might contain HTML tags.
User/Project/ LongDescription	string	Long description of the project. Might contain HTML tags.

Example:

```
GET /system/v1/apikeys/users/812121

ResponseData:
{
    "User" {
       "Projects" : [
           {
              "Id" : "23134",
              "Name" : "New cool app",
              "Status" : ["Test", "Terminated"],
              "ShortDescription" : "We do stuff",
              "LongDescription" : "We do a lot of stuff",
           },
           {
              "Id" : "787",
              "Name" : "Another cool app",
              "Status" : ["Test"],
              "ShortDescription" : "We do stuff",
              "LongDescription" : "We do a lot of stuff",
           }
       ]
    }
}
```

Update keys

HTTP Method: GET
URL: /system/v1/apikeys/keys/{KEY}/refresh
In the case a key is updated directly in the Transit Company systems it is the responsibility of the Transit Company to inform Trafiklab of this. That way Trafiklab can update any stored information about this key. Informing Trafiklab is made by calling this method.

Update profiles

HTTP Method: GET
URL: /system/v1/apikeys/apis/{API ID}/profiles/refresh
In the case a profile is updated directly in the Transit Company systems it is the responsibility of the Transit Company to inform Trafiklab of this by calling this method. That way Trafiklab can update any stored information about this profile.

Summary of Responsibilities

It is the responsibility of Trafiklab to:

- Implement the Trafiklab Key API according to the specifications in this document
- Make the necessary user credentials available to the Transit Company
- Send the Transit Company all the API IDs for the Transit Company's APIs.
- Keep any stored data about keys and profiles synchronised with the master data provided by the Transit Company.

It is the responsibility of the Transit Company to:

- Implement the Transit Company Key API according to the specifications in this document.
- Provide Trafiklab with the host, path and port needed to call the API.
- Make the necessary user credentials available to Trafiklab
- Define all profiles for each API
- Define which profile is the default profile
- Implement calls to the "refresh" methods in the Trafiklab Key API if the Transit Company can update a key or a profile in their own systems.
- Keep any stored data about users and projects synchronised with the master data provided by Trafiklab.