RESTful API

This document contains information about the RESTful API built into the Visio web server.

Author: Visio Team 1

About

The main goal of the API is to create a safe bridge between the mobile application and the web portal. A example of the usage of the API would be to get more information about a user such as age or gender. The mobile application only needs a valid token in order to get this information from a user. Another example would be to get more information about a nearby beacon by using the beacon its id.

The reason why we use an API is to prevent stored database information in the mobile application. This prevents that our database gets hacked and that we can control the flow of information that the user gets to see.

Security

In order to obtain certain data from the API the requester will need to provide a unique MD5 hashed token. It’s the job of the mobile application to save the token given by the server when creating a new user account so that it may be used in the future. When requesting information about a user the mobile application may search for it based on the unique token.

The token prevents hackers from obtaining personal information from the API. Every user has a different token so they only can request information about themselves. In order for a user to be compromised the hacker would need access to the mobile device itself.

Actions

The API has a root path which all the actions connect to; ‘**/api/**’. So if the name of the website is ‘visio’ then the root path for the API would be; ‘**www.visio.com/api/**’.

ExternalUser

JSON ExternalUser Object Example:

{"externalUserId":1, "age":20, "gender":"0", "vitid": {"description":"test", "vitid":1}}

Action: get user information

Path: /user/

Method: GET

Parameter: (int) id of the external user

Responses: SUCCES (1), DOESNT_EXIST (3)

Example URL: www.visio.com/api/user/1

Output: JSON ExternalUser Object

Action: add external user

Path: /user/

Method: POST

Parameter: (ExternalUser) ExternalUser object

Responses: SUCCES (1), ALREADY_EXIST (2), WRONG_FORMAT (4), UNKNOWN ERROR (5)

Example URL: www.visio.com/api/user/

Input: JSON ExternalUser Object

Output: JSON ExternalUser Object **WITH the generated token**

Beacon

JSON Beacon Object Example:

```
{"beaconId":1,"uuid":"aaaca6c1-ca80-4c98-8a46-c81dd974e9e6","major":1,"minor":2,"dangerLevel":1,"longitude":1.0,"latitude":2.0,"beaconGroupId":{"beaconGroupId":1,"description":"test"}}
```

Action: get beacon information

Path: /beacon/

Method: GET

Parameter: (String) uuid, (int) major, (int) minor

Responses: SUCCES (1), DOESNT_EXIST (3)

Example URL:

www.visio.com/visio/api/beacon/aaaca6c1-ca80-4c98-8a46-c81dd974e9e6/1/2

Output: JSON Beacon Object

Responses

Here we will list all the possible responses from our API along with their HTTP status.

1. SUCCES

HTTP Status: Accepted
Code: 202
Description: Everything went good without any issue’s.

2. ALREADY_EXIST

HTTP Status: Conflict
Code: 409
Description: The object that is being added already exists in the database

3. DOESNT_EXIST

HTTP Status: No Content
Code: 204
Description: The requested object does not exist

4. WRONG_FORMAT

HTTP Status: Bad Request
Code: 400
Description: Wrong or missing input

5. UNKNOWN ERROR

HTTP Status: Conflict
Code: 409
Description: Error caught that is not detected by the system