**API:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Explanation** | **Returns** | **Parameters** | **HTTP Method** | **Method name** | **ID** |
| Register cannot pass in GET, must use Post. | Creates user, saves all info to DB. | {  "user\_name" : "zivto",  "password" : "a12345",  "first\_name" : "Ziv",  "last\_name" : "Tomarov",  "city" : "Beer-Sheva",  "country" : "Israel",  "email" : "zivtomarov@gmail.com"  "question": "What is Your Elementary School?",  "answer" : "Shazar",  "category": "sport", "food"  } | Post | /guests/register | 1 |
| Login cannot pass in GET, must use Post. | Creates Token | {  "Username" : "zivto",  "Password" : "a12345"  } | Post | /guests/login | 2 |
| A method used to get the users question | Returns the users security question**.** | {  " user\_name": "zivto"  } | Get | /guests/forgetPassword  /:user\_name | 3 |
| A method used to verify the answers to privacy questions. Using POST because we don’t want the question and answer to be shown in the url. | Sends an email to user | {  "user\_name": "zivto"  " Email" :"zivtomarov@gmail.com",  "Question": "What is Your Elementary School?",  "Answer" : "Shazar"  } | Post | /guest/forgetPassword  Answer | 4 |
| There is no information to deliver to the server or any wish to change any information. The Interests are stored in the DB so we'll use GET. | Returns list of three **random interests by certain rank:**  List<interest> | --------- | Get | /poi/threeRandomPOI | 5 |
| Recommended interests has no information to deliver to the server or any wish to change any information. The Interests are stored in the DB so we'll use GET. | Returns list of two **last saves (chooses by date) :**  List<interest> | --------- | Get | /users/twoLastPOI | 6 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Recommended interests are for the client.  There is no information to deliver to the server or any wish to change any information. The Interests are stored in the DB so we'll use GET. | **Returns list of two favorite Interests by User ID:**  List<Interest> | --------- | Get | /users/twoUserPOI | 7 |
| Category interests connected to server side, using Get because we have data that flows to the server no updates. | **Returns list of Interests by category:**  List<Interest> | POI category | Get | /poi/searchPOIcatagory  /:POI\_category | 8 |
| The interests are connected to server side, using Post because we have data that flows to the server no updates. | **Returns interest by specific name.** | POI Name | Get | /poi/searchPOIname  /:POI\_name | 9 |
| Using POST because we need to deliver the favorite interest to the server by adding it to the DB. Return “failed” if failed. | **Saves POI yo users favorites.** | {  "POI\_name": "Central Park"  } | Post | /users/saveFavoritePOI | 10 |
| Using POST because we need to deliver the favorite interest to the server by removing (deleting) it to the DB. Return “failed” if failed. | **Removes POI from users favorites.** | POI name | Delete | /users/removeFavoritePOI  /:POI\_name | 11 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Num of favorites is connected to server side, using Get because we have data that flows to the server no updates. | **Returns number of Interests in List:** int | --------- | Get | /users  /numberOfFavoritePOI | 12 |
| Using POST because we need to add a review to an interest through the server to the DB. Return “failed” if failed. | **Returns POI's reviews by users:**  List<reviews> | {  "user\_name": "zivto"  "POI\_name": "liberty statue",  "review": "Amazing"  "rating": "4",  "created": "29/05/2018 17:00:12"  } | Post | /users/reviewPOI | 13 |
| Favorites are connected to server side, using Get because we have data that flows to the server no updates. | **Returns favorite Interests saved by user:**  List<Interest> | --------- | Get | /users/FavoritesPOI | 14 |