|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Name** | **HTTP Method** | **Parameters** | **Returns** | **Explanation** |
| 1 | login | POST | Username, password | Token | Login cannot pass in GET (not safe) |
| 2 | register | POST | First name o Last name o o City( איפה המשתמש גר( o Country( איפה המשתמש גר( Email o o תחומים המעניינים את הלקוח, לפחות 2 o שאלות אימות לשחזור הסיסמא | Token |  |
| 3 | RestorePassword | POST | username,question  ,answer | new random password | Check if correct -> return password |
| 4 | insertQuestion | GET | username,question,answer | token | Add a new question to user on a city or place in city |
| 5 | getRandomPOI | GET | amount | POI[amount] | Return #amount random points of interest in current city |
| 6 | getAboutInfo | GET | nothing | String | return short summary on the city |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 7 | save\_attraction | GET | User name , attraction o | Token | A peronalised favourite list addition shouln't be public, and we are adding information to the server, so GET is appropriate |
| 8 | get\_attractions\_by\_category\_sort | GET | Category | Returns all attractions in a category  In sort order | Using GET because there is no information to deliver to the server or any wish to change the information. The filtering can be implemented on the client side but this is wasteful In my opinion. |
| 9 | last\_saved | get | Poi ID | Returns top 2 last review on the POI | We using GET because we want to get the reviews and we don’t care about security |
| 10 | get\_attractions\_by\_name\_sort | GET | String s | Returns all attractions whose names correspond to the query in sort order | Using GET because the parameter "attraction\_name" is just specifying what are we searching for, it's not a password |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 11 | get\_saved\_attractions | GET | User name | Returns all saved attractions | Using GET because there is no information to deliver to the server or any wish to change the information |
| 12 | save\_review | POST | attraction o rating o review o  username | Token | reviews aren't personal like login info but still it is safer and more correct to send it to the server with POST instead of GET. This method will be triggered by the user, as normally the attractions are saved in the browser's cache. |
| 13 | get\_attractions\_by\_category | GET | Category c | Returns all attractions in a category | Using GET because there is no information to deliver to the server or any wish to change the information. The filtering can be implemented on the client side but this is wasteful In my opinion. |
| 14 | get\_attractions\_by\_name | GET | attraction\_name | Returns all attractions whose names correspond to the query | Using GET because the parameter "attraction\_name" is just specifying what are we searching for, it's not a password |
| 15 | remove\_saved\_attraction | DELETE | Attraction ID, username | Token | Using DELETE to delete |
| 16 | getCities | Get | nothing | string | Return the website name |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 17 | add\_new\_POI | get | Name, category | string | Api function for manegers, add to the POI DB standard POI with not a lot of meat, will be more efficient un the future |
| 18 | get\_attractions\_by\_rate\_sort | get | nothing | POI[] | Return the POIS in the DB sorted by they avg\_rating |
| 19 | move\_to\_POI | get | POI id | POI | Returns all the info about a specific POI |
| 20 | delete\_review | DELETE | Username,POI ID | token | Delete a review of user if we as a managers of the website want to. |
| 21 | getPOIinfo | GET | POI ID | String of description | Return only the description of given POI |
| 22 | getReviewByPOI | GET | POI ID | Review[] | Return all the reviews on a given POI |
| 23 | 2POIforUser | GET | username | POI[2] | Return the last 2 POIs that the current user saved |
| 23 | POIforUser | GET | username | POI[] | Return all the POIs that the current user saved |