**API מעודכן**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Method Name** | **HTTP Method** | **Parameters** | **Returns** | **Explanation** |
| 1 | /Users/LogIn | POST | {“UserName”, “Password”} | Token | Login cannot pass in GET because it not secured. |
| 2 | /Users/Register | POST | JSON userDetails  {“username”,”Password”,  ”FirstName”,“LastName”,  ”City”,”Country”,”Email”,  “FavoriteCategories”, “questionID“,  questionsAns”} | Token | The user’s details are stored at the DB.  True=success  False=fail |
| 3 | /Users/RestorePassword | POST |  | Token | Using POST because GET is not secured for the user name and the user’s answers transfer together. |
| 4 | /POI/ (Midelware) | Use??? | Token |  | Using for verify the user token.  Return a massage if the authentication of the token failed |
| 5 | /POI/POIName/:POI\_Name | GET |  |  |  |
| 6 | /POI/PopularPOI | GET | - | {“POI\_Name”} | Returns the name of all popular POI **above threshold rank** (60%) from the DB. No changes at the DB. |
| 7 | /POI/Category/: Category\_Name | GET | {“ Category\_Name”} | JSON of POIs  {“POI\_Name”, “Category\_Name, “Num\_Of\_Views”,  ”Rank\_Per”, “POI\_Picture”, “Review\_Desc\_1”, “Review\_Desc\_2”, “Review\_Date\_1”, “Review\_Date\_2”} | Return list of all the POI by the given category. Using GET because no changes in the DB IS required. |
| 8 | /POI/RemovePOIFromUserFav | DELETE | { “POI\_Name”} | - | Delete POI from the user favorites list in the DB. |
| 9 | /POI/AddPOIToUserFavorites | POST | } “POI\_Name”} | - | Using POST because we need to add the POI to user favorites in the DB.  Return “FALSE” if failed |
| 10 | /POI/GetUser2PopularPOI | GET | - | JSON of 2 POIs  {“POI\_Name”, “Category\_Name, “Num\_Of\_Views”,  ”Rank\_Per”, “POI\_Picture”, “Review\_Desc\_1”, “Review\_Desc\_2”, “Review\_Date\_1”, “Review\_Date\_2”} | Return the user two most popular POI by the user favorites categories (one from each category) |
| 11 | /POI/GetUserAllFavPOI | GET | - | JSON of all the user favorites POIs  {“POI\_Name”, “Category\_Name, “Num\_Of\_Views”,  ”Rank\_Per”, “POI\_Picture”, “Review\_Desc\_1”, “Review\_Desc\_2”, “Review\_Date\_1”, “Review\_Date\_2”} | Return list of all the user favorites POI. using GET because there is no information to deliver to the server or to change in the DB |
| 12 | /POI/GetAllPOI | GET | - | {“POI\_Name”} | Return the Name Of all the POIs Using GET because no changes in the DB IS required. |
| 13 | /POI/POIRank | POST | {“POI\_Name”, “Rank”} | - | Using POST because we need to update the POI rank in the DB. |
| 14 | /POI/POIReview | POST | {“POI\_Name”, “Review”} | - | Using POST because we need to update the POI review in the DB. |
| 15 | /POI/RecentSavedPOI | GET | {“User\_Name”} | JSON of 2 POIs  {“POI\_Name”, “Category\_Name, “Num\_Of\_Views”,  ”Rank\_Per”, “POI\_Picture”, “Review\_Desc\_1”, “Review\_Desc\_2”, “Review\_Date\_1”, “Review\_Date\_2”} | Returns the user’s 2 last saved POI from the DB.  True=if found 2 POI  False=2 POI didn’t found |
| 16 | /POI/SaveAllFavPOIByUser | POST | {“POI\_Array: [“POI\_Name”, ”date”]} | - | Using post in order to add new favorites POI to the user favorites list. |
| 17 | /POI/SetlFavPOIOrderByUser | POST | {“POI\_Array: [“POI\_Name”, ”POI\_Order”]} | - | Using post in order to set the order of the favorites POI of the user. |

**הערה:** כל בקשה של משתמש רשום (GET,POST,DELET....) מגיעה עםToken **בHEADER**, כלל הבקשות האלו מגיעות לModule – POI ומגיעות קודם כל ל Midelwareלטובת פיענוח של הtoken ובדיקת הנכונות על מול הDB.  
לכן הפרמטר token צויין רק בMidelware אך מגיעה בכל בקשה לPOI.  
כמו כן, הuser name אינו מועבר כפרמטר בבקשות אלו מפוענח מתוך הtoken.