|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Method Name | HTTP Method | Parameters | Returns | Explanation |
| l | /users/login | POST | Username,  Password | Token | For Security reasons login cannot pass in GET method.  True=success  False=fail |
| 2 | /users/register | POST | firstName,  lastName, city, country, Email, verifyQuestion1, verifyAnswer1,  verifyQuestion2,  verifyAnswer2,  categories[], userName,  userPassword | Success/eror | Register the User to the system, and save his info in the DB |
| 3 | /POI/getPOI/:id | GET | id | POI | Returns this information about a specific point of view represents by ID. This function also update amountOfWatchers+1 |
| 4 | /user/getVerificationQuestions | GET | userName | Verify\_Questions | Get the 2 verify\_Question for specific user |
| 5 | /user/getPassword | POST | username,  verifyAnswer1  verifyAnswer2 | Password | If the Verify\_Answer match the Verify\_Question of specific user in the DB, get password  Else return error message |
| 6 | /POI/getPopularPOI | GET | amount (int),  rank | POI [] | Get number (=amount) of random popular points of interest ranked higher then rank parameter from DB |
| 7 | /POI/getLastUploadPOI/:amount | GET | amount (int) | POI [] | Get number (=amount) of last uploaded points of interest from DB |
| 8 | /user/reg/getUserFavoritCategories | GET | token | Categories[] | Get the favorit categories of specific user from DB |
| 9 | /POI/getPOIbyCategory/:category | GET | category, | POI[] | Get POI by user favorit categories from DB |
| 10 | /POI/reg/getSavedPOI | GET | token | POI[] | Get ALL POI by user saved from DB |
| 11 | /POI/reg/savePOIToDb | POST | token id(poiid) | Success/error | save new point of interest as favorit at db. |
| 12 | /POI/getPOIrank/:id | GET | POI\_ID | RANK | GET POI rank from DB |
| 13 | /POI/getPOIbyName/:name | GET | POI\_name | POI | Get POI by its name |
| 14 | /POI/reg/deletePOIfromFavorits | DELETE | token id(poiID) | Success/error | Delete POI from user favorits in DB |
| 15 | /POI/reg/postPOIReviwe | POST | id string: reviwe  token | Success/error | Post user review to POI and calculate its new rank and add the reviwe |
| 16 | /POI/reg/postPOIRank | POST | id  Int:rank  token | Success/error | Add user reank and Calculate rank avarage |
| 17 | /POI/reg/get2PopularByUserCat/ | GET | token | POI[] | get 2 popular POI by 2 different user categories |
| 18 | /POI/updateViews/:poiID | UPDATE | poiID | Success/error | Update the number of views of POI |
| 19 | /POI/reg | USE | token | username | Verify token for methods to register users only |
| 20 | /POI/getPicture | get | ID | picture | Return all the pictures of specific poi |
| 21 | /POI/getPoiReview | get | id | Reviews[] | Returns all the reviews of specific poi |
| 22 | /POI/getAllPOI | get | - | POI[] | Returns all the POI in the db |

הסברים לגבי הוספת שלושת השיטות האחרונות :

מס' 20 : הוספנו זאת מכיוון שהחלטנו שאם נרצה בעתיד להוסיף עוד תמוונות עבור כל נקודת ענין אז נצטרך טבלת תמונות ולכן נצטרך פונקציה שמחזירה עבור נקודת עניין מסוימת את כל התמונות שלה .

מס' 21 : הוספנו זאת מכיוון ששכחנו להוסיף את הפונקציה לפני כן ( לא תמיד חושבים עד הסוף ) תפקידה להביא את כל הביקורות על נקודה ספציפית ( ביקורת הנמצאות בטבלה נפרדת )

מס' 22 : גם כן הוספנו זאת רק לאחר שהתחלנו לבנות את האתר . מטרת הפונקציה להחזיר את כל נקודות העניין – בשביל להציג אותם במסך הנקודות עניין .

POI{

'Id':int,

'Name':string,

'Pictures':[string],

'Description':string,

'Views': int,

'Rank':int  
}