מטלה 3.1 – API

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Module** | **Method Name** | **HTTP Method** | **Parameters** | **Returns** | **Explanation** |
| 1 | /Users | /authenticate | POST | '{ "userName":, "password": }' | Token | These specific parameters should not pass in the URL – hence the POST. |
| 2 | /Users | /retrievePassword | POST | '{ "Username": , "Q1":, "A1": , "Q2": ,"A2": }' | '{ "Password": }' | The answers to the security questions should not be passed in the URL – so POST is needed. |
| 3 | /Users | /  (addUser) | POST | '{ "Username":, "Password":, "Fname":, "Lname":, "City":, "Country":, "Email":, "categories []":, "Q1":, "A1": , "Q2":, "A2": }' | Success- 200OK  Fail- "choose another Username" | Parameters need to be sent – so POST is used.  True = successful user registration, False = registration fail. |
| 4 | /POI | /reg | USE | '{"token":}' | '{ "Username":,  "categories[]": }' | Check token and details before doing method for register users only |
| 5 | /POI | /getPOIbyID/:id | GET | id | POI object  {“ID:”, "Name":, "Description":,  "Rank":, " CategoryID": ,  "UsersWatching": , "Picture":, "Reviews": lastTwo[] } | Retrieves the POI object by id |
| 6 | /POI | /getPOIbyName/:name | GET | name | POI[]  {“ID”: ,"Name":, "Description":,  "Rank":, " CategoryID": ,  "UsersWatching": , "Picture":, "Reviews": lastTwo[] } | Retrieves the POI object Array by name  (Using LIKE sql) |
| 7 | /POI | /getAllPOI | GET | - | POI[]  {“ID”: ,"Name":, "Description":,  "Rank":, " CategoryID": ,  "UsersWatching": , "Picture":, "Reviews": lastTwo[] } | Retrieves all the POIs in the db. return Array |
| 8 | /POI | /Random3 | GET | - | POI[]  {“ID”: ,"Name":, "Description":,  "Rank":, " CategoryID": ,  "UsersWatching": , "Picture":, "Reviews": lastTwo[] } | Retrieves 3 random POIs from the db. return Array |
| 9 | /POI | /reg/FavoritesByUsername/:num | GET | If num=2- Retrieves 2 last favorites.  Else- Retrieves all favorites | POI[]  {“ID”: ,"Name":, "Description":,  "Rank":, " CategoryID": ,  "UsersWatching": , "Picture":, "Reviews": lastTwo[] } | Retrieves the users favorites, return an array |
| 10 | /POI | /reg/get2byCat/ | GET | - | POI[]  {“ID”: , "Name":, "Description":,  "Rank":, " CategoryID": ,  "UsersWatching": , "Picture":, "Reviews": lastTwo[] } | Retrieves 2 random POIs , one from each random category that the user choose upon registration. Return an array |
| 11 | /POI | /reg/storeFav | POST | Favorites[]  { " favorites:" } | Success/Fail | Stores the user's local favorite list in the database |
| 12 | /POI | /reg/addRank/ | POST | '{"id":, "rank:", "body:" }' | Success/Fail | Store the review in the db and calculate new rank for the POI |