**מטלה 3.1 – הצעת API / אילן סרנוב**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Method Name** | **HTTP Method** | **Parameters** | **Returns** | **Explanation** |
| 1 | Users/Login | POST | {  “Username”:,  “Password”:,  } | Token  User details:  {  “fname”: ,  “interests”: ,“favoritepoi”:  } | Login cannot pass in GET True= success False = fail  Get the interests to show relevant POI, get the favorites to show him his favorites |
|  |  |  |  |  |  |
| 3 | POI/POIDetails/:id | GET | token | Returns POI details: {  “POIid”: ,“name”: , “watchCount” :, “POIDesc”:, “rank”: ,”reviews”: [ “reviewID”: {“username”:, “review”:, “date”: , “rank”:} ]  } | Using GET since the id of the POI doesn’t need to be hidden and no other parameters needed |
| 4 | Users/register | POST | {“username”: ,  “password”:,  “fName”: , “lName”:, “city”:, “country”:, “email”: ,“secretquestion”: ,”secretanswer”:  ,”interests”:  } | Returns status and message :  200 : “You are successfully registered to the site and will be redirected to the login page  400 : “Invalid data: ${error}” | POST since we need the details to be passed in body and be secured |
| 5 | POI/poidetails | GET | No Sent Parameters | Returns list of POI: {  “PoiID”:,  “WatchCount”:,  “name”: ,  “PoiDesc”:,  “Category”:, “TotalRank”:  “NumberOfRanks”:  } | Using get since we are not sending any sensitive information |
| 6 | POI/searchPOI/?name | GET | No Sent Parameters | Route to POI interface and filter to specific poi by name  Return error if not found | Using get because no sensitive data was sent and no change is needed |
| ~~7~~ | ~~putFavorite~~  Will be processed in Client side. Will update the list at the end (Users/favorites -> PUT) | ~~PUT~~ | ~~Token in header.~~  ~~{“username”: , “POIid”: }~~  ~~in body~~ | ~~Return request status and the new favorite list of the id~~  ~~Will add poi if poiid is not on the list yet, remove it if it is (Toggle)~~ | ~~PUT is needed to update the database with the new point of interest added or removed~~ |
| 8 | Users/favorites  (Returns in login) | Get | Token in header | Return the list of favorites | Not updating anything |
| 9 | Users/favorites | PUT | Token in header  {  “Favorites”:  } ids of the sorted poi list } | - | Will be read and update the list in database to show it correctly next time its pulled  Using PUT since we are updating the data base |
| 10 | Users/retrievePassword | POST | {  “username”:  } | Return information for user validation  {  “username”: ,”secretquestion”: ,”secretanswer”: , “password”:  } | Using post to send the username in body, all the validation will happen in front end using the details sent from DB |
| 11 | POI/review | PUT | Token in header{  “PoiID”: ,  “Review”: ,  “Date”: ,  “Rank”: ,  } | Return success if the review was added to the database  Will make sure only one review per username per PoiID will be sent on the same day | Using PUT since we are updating the data base with new entry of review |
| 12 | Users/SuggesterPOI  Added during client development since I decided that it will be right to do the JOIN in SERVER side and not CLIENT | GET | Token in header | Returns list of POI after JOIN table of Interests and POI by Category Where Username = Token.username: {  “PoiID”:,  “WatchCount”:,  “name”: ,  “PoiDesc”:,  “Category”:, “TotalRank”:  “NumberOfRanks”:  } | Using get because no body information is needed in send |

