API WEB

GIT: <https://github.com/yosimeltser/server>

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Method Name** | **HTTP Method** | **parameters** | **Returns** | **Explanation** |
| 1 | /register | POST | USER : {  First name:  Last name:  City:  Country:  Email:  Categories[]:  Questions[]:  Verifiers[]:  Username:  Password:  } | Boolean | True = success  False = fail  Register cannot pass with GET.  Create a new user in the DB. |
| 2 | /retrieve/Answers | POST | Username,  Answers [] | Password (if wrong returns false) | Using POST to avoid revealing the answer and check for a match. |
| 3 | Login | POST | Username, password | if login succeed:  Token: {  "success”:  "message":,  "token":  }  else  Return  success: false | Login cannot pass in GET. |
| 4 | /Points | GET | - | List of all interest points.  IP []:{  ID:  PointName:  Category:  Views:  Ratings:  Ref:  } | GET - no change in DB,  Return all the interest points. |
| 5 | Users/saveInterestPoint | POST | Interest point ID,Token | Boolean | POST change in DB – adding interest point to the user's list. |
| 6 | Users/removeInterestPoint/:id | DELETE | Token, Interest point ID | Boolean | True = success  False = fail  DELETE – needs to remove the interest point from the user's list. |
| 7 | Users/Favorites | GET | Token | Saved list of interest points per user | GET- retrieve the favorites page by user id. |
| 8 | Users/Rank | POST | Rank  Point ID | Return true if succeed, return false if error occurs. | True = success  False = fail  POST- change in DB – adding rank to the point's review list. |
| 9 | Users/Comment | POST | Comment  Point ID | Boolean | True = success  False = fail  POST- change in DB – adding comment to the point's review list. |
| 10 | Users/2InterestPoint | GET | Token | 2 Interest point  InterestPoint {  ID:  PointName:  Category:  Views:  Ratings:  Ref:  } | GET- retrieve 2 of the most popular interest points, each one from different category that user favors. |
| 11 | Users/Last2Saved | GET | Token | 2 Interest points. | GET- retrieve the last 2 interest points that the user saved. If there are none – show appropriate message. |
| 12 | /3RandomPoints | GET | - | 3 Interest points. | GET- retrieves 3 random interest points from the DB. There are no change in the DB. |
| 13 | Users/saveOrderedFavoriteList | POST | Token, list of favorite interest points ID's | Boolean | POST we make an update in the DB, saves the favorite list in the order that user asked for. |
| 14 | interestPoint/:id | GET |  | JSON Interest point  As described above. | GET – we don’t make any change in the DB. Retrieve the interest point as described above. |
| 15 | /retrieve/Questions/: Username | GET | - | 2 Verification questions  Else false | GET – we don’t make any change in the DB. Retrieve the verification questions that user choose in registration |
| 16 | /countries | GET | - | Countries[Country]:  Country: {  ID:  Name:  } | GET – we don’t make any change in the DB. Retrieve the countries that user can choose from. |
| 17 | /last2comments/:id | GET | - | Comment{  Text:,  Date:  Author:  } | GET– we don’t make any change in the DB. Retrieve the 2 last comments for a given point. |

הסבר ל- 17 : התגובות הן דינמיות ותמיד מתווספות חדשות. עלינו כל פעם לבדוק מהן 2 התגובות האחרונות שנוספו.

**הנחות :**

* על מנת לצמצם גישות לשרת הוצעה מתודה אחת ויחידה getInterestPoints שכל תפקידה יהיה להביא את רשימת נקודות העניין כאשר המיון, הסינון והחיפוש ע"פ שם יתבצעו בצד הלקוח.
* Username הינו מזהה ייחודי.
* החלטנו כי לאחר כל דירוג או תגובה של משתמש המידע יתעדכן רק לאחר גישה נוספת לדף על מנת לחסוך בפנייה נוספת לשרת ורינדור של הדף דבר אשר יפגע בחוויית המשתמש.