

סביבות פיתוח- עבודה 3.1

<u>ID</u>	<u>Method Name</u>	<u>HTTP Method</u>	<u>Parameters</u>	<u>Returns</u>	<u>Explanation</u>
1	Users/register	POST	Username, Password, FirstName, LastName, City, Country, Email,	Boolean	We save at the DB private detail. We used POST method because we are passing private details of the client in the message's body. Returns: True=Success, False=Failed
1.1	Users/chooseCategories/: username	POST	categories{Cat(i)} 1<=i<=4	Boolean	User chooses categories added to his category list
1.2	Users/answerQuestions/: username	POST	Questions[]	Boolean	User chooses answer for restoring password
2	Users/log	POST	Token	{ "Username", "FirstName", "LastName", "City", "Country", "Email", "Categories" }	Using token to determine that the user who posts something is the actual user

3	Users/Login	POST	Username, Password	Token	We will Use POST method because we are passing private details, then we will write it at the message's body.
4	Users/restorePassword	POST	Username, ans1, ans2	String: Password	The password stored at the DB (private detail). The server is the only one who can get the data. We use GET method because there is no need to change the DB or transfer any private details.
5	Users/username/:username	POST	username	boolean	If the user exists
6	Users/log/categories/:username	GET	username	Json {UserCategories[]}	Returns user categories
7	Users/log/getFavorites/:username	GET	username	Returns list of favorites points: { "pointId", "pointName", "category", "rating", "description", "pictures", "viewNum", "reviews" }	Using GET Method to receive the list of objects makes by the client. A list is created in the DB and this function returns it.

8	Users/log/insertFavorite/:username/point/:pointId	POST	Username, pointId		Using POST to add to the list of favorites in user Interface. The user adds the object to the list therefore the user needs a post method.
9	Users/log/ratePoint/:pointId	POST	Username, pointId, rate		Rate the pointId by specific username
10	Users/log/reviewPoint/:pointId	POST	Username, pointId, review		Review the pointId by specific username
11	Users/log/removePoint	DELETE	pointId		Remove point from favorites by user
12	Users/countries	GET		List countries[]	Return countries using given XML

13	Points/getPoints	GET		Returns list of Interests: {“pointId”, “pointName”, “category”, “rating”, “description”, “pictures”, “viewNum”, “reviews”}	The points save at the DB. We will return all the points at the DB and the client will be responsible to filter the sites according to the rate he decide and in randomly. We use GET method because there is no need to change the DB or transfer any private details.
14	Points/category/: category	GET	categoryID	Return Json{categoryID,category}	Return category by categoryID
15	Points/getPointbyid/: pointId	GET	pointId	{“ pointId”, “pointName”, “category”, “rating”, “description”, “pictures”, “viewNum”, “reviews”}	All the sites details saved at the Server Side in the DB. We use GET method because there is no need to change in the DB or transfer any private details.
16	Points/getPointbyName/: pointname	GET	pointName	{“ pointId”, “pointName”, “category”, “rating”, “description”, “pictures”, “viewNum”, “reviews”}	All the sites details saved at the Server Side in the DB. We use GET method because there is no need to change in the DB or transfer any private details.
17	Points/removePoint	DELETE	pointID	[]	Remove point from DB
18	Points/viewsIncrease/: pointId	POST	pointID	[]	Increase views in point
19	Points/rate/: pointId	POST	pointID	[]	Calculates the percentage of all rates by users in certain point

מגישים : לינוי קלכמן 305280935
אסתר פבין 203382114

- נקודות העניין הרנדומליות לפי דירוג מסוים יתבצעו בצד הלקוח לאחר שנחזיר את כל נקודות העניין ללקוח. (getPointsOfInterests).

