Shani Houri 204508477
Eyal Arviv 311264592

API

**Updates are marked in yellow

ID	**Method Name**	**HTTP Method**	**Parameters**	**Returns**	**Explanation**
1	Register	POST	Username, Password, ConfirmedPassword, First name, Last name, City, Country, Email, Categories[size] (size > 1), answersForRecovery[2]		Register cannot pass in GET Using POST method because a resource of a user and his details is created in the DB. True= success False=fail
2	Login	POST	Username, Password	Token	Login cannot pass in GET. Also we create a token in the login session (new resource) which is returned to the client. The token is stored on the client side (and not on the server – stateless) True= success False=fail
3	PasswordRetrieval	POST	Username, user's answers	{"password"}	The user's details are stored in the DB, so only the server can get it. Return the password of the given user. Using POST because of the sensitive data that is being transferred to the client.
4	GetSinglePointOfInterest	GET	name	{"numOfViews", {"description", "rank", "twoLastReviews", "Category", "name", "picture"}	Returns one point of interest. Using GET because there is no wish to change any information.
5	GetAllPointsOfInterest	GET		[{~~"numOfViews",~~ ~~"description",~~ ~~"rank",~~ ~~"twoLastReviews",~~ ~~"Category",~~ "name", "picture"}]	Returns all the points of interest in the city. Using GET because there is no information to deliver to the server or any wish to change any information. Return only relevant fields which are name of poi and picture path.
6	SavePointOfInterest	POST	name, TOKEN		Using POST because we need to deliver the name of the point of interest to the server for adding it to the favorites of the user in the DB. The change must be made on the server so the client can see it even when his local storage is deleted.
7	~~SaveFavorites~~	~~POST~~	~~Names[], TOKEN~~		~~Using POST because we need to deliver the names of the points of~~

					~~interest to the server for adding it to the favorites of the user in the DB. The change must be made on the server so the client can see it even when his local storage is deleted or when he logs on a different computer.~~
8	UnsavePointOf Interest	DELETE	Name, TOKEN		Using DELETE because we need to deliver the name of the point of interest to the server for removing it from the favorites of the user in the DB.
9	GetFavoritePoi ntsOfInterest	GET	TOKEN	[{~~“numOfViews”, {“description”, “rank”, “twoLastReviews”, “Category”,~~ “name”, “~~picture~~”,"position "}]	Returns all favorite points of interest of the logged user. Using Get because no resource is created – only display of existing resources to the user.
10	Get2MostPop oularPointsOfI nterestForUse r	GET	TOKEN	[{“numOfViews”, “description”, “rank”, “twoLastReviews”, “Category”, “name”, “picture”}] (size 2)	Returns two points of interest in two different categories chosen by the user in registration. Using GET method because there is no resource created.
~~11~~	~~Get2RecentPoi ntsOfInterest~~	~~GET~~	~~TOKEN~~	~~[{“numOfViews”, “description”, “rank”, “twoLastReviews”, “Category”, “name”, “picture”, “CreatedAt”}] (size 2)~~	~~Returns point of interest of the given name. Using Get because no resource is created.~~ ~~True= find False=alert~~
12	ReviewPointOf Interest	PUT	Review, name		Using PUT because we create here a resource of the review to the given point of interest.
13	RankPointOfIn terest	PUT	Rank, name		Using PUT because we update the rank of the chosen point of interest in the DB.
~~14~~	~~UpdatePoints OfInteres~~	~~GET~~		~~[{“numOfViews”, “rank”, “twoLastReviews”,}]~~	~~Returns the dynamic fields of points of interest to update them in the client side to be the most relevant. This method will be called before the client will be displayed the detailed points of interest. Using GET method because there is no resource created.~~
15	GetCategories	Get		[{“Category”}]	Return all the categories. Using GET method because there is no resource created.

Shani Houri 204508477
Eyal Arviv 311264592

16	Get3PopRand			[{"numOfViews", "description", "rank", "twoLastReviews", "Category", "name", "picture"}] (size 3)	Return three popular and random points of interest. Using GET method because there is no resource created.
17	reorder	PUT	[{"name"}], token		Change the order of the favorite POIs according to the order of the array. Using Put because we update the positions of the existing favorite POIs.

Assumptions:

-When the client first loads the page, all of the static resources such as HTML, CSS and JS files, pictures, etc., are loaded from the server. We don't mention this in the API above as it was said in class it is not necessary.

-The names of points of interest are unique.

-All of the functions that can be called only by a logged user are given the user's TOKEN as a parameter.

-Writing review to a point of interest is possible to any kind of users (registered or just visitors)

-Since our database contains relatively low number of points of interest (5 points in each of the 4 categories), actions like sorting by user's preference will be implemented in client side only (after retrieving all points of interest from the server). Therefore there is no need to point these actions in the API above. Displaying filtered, sorted and random points of interest will be executed in the client side in order to save requests to the server and enable faster communication and better user's experience when he roams the website.

-When the user adds a point of interest to the favorites, the server will update it together with a time stamp of the time updated for retrieving later on the most recent added to favorite's points of interest.

-In 16 – We assume that points of interest that got more than 70% rank are considered popular.

-In 17 – We assume that the wanted order determined according to the order of the array's elements.

Link to Git:

https://github.com/Eyal8/TripMe