

<u>ID</u>	<u>Method Name</u>	<u>HTTP Method</u>	<u>Parameters</u>	<u>Returns</u>	<u>Explanation</u>	
1	/users/	GET	-	returns all users: users[]	Using GET because a resource from server is needed and there isnt any wish to change the state of the server	we need those functions to the restore password part, we dont have the Token at that time. these functions dont need to be under the token middleware
2	/users/id/:id	GET	user id	Returns user	Using GET because a resource from server is needed and there isnt any wish to change the state of the server	we need those functions to the restore password part, we dont have the Token at that time. these functions dont need to be under the token middleware
3	/users/username/:username	GET	username	Returns user	Using GET because a resource from server is needed and there isnt any wish to change the state of the server	we need those functions to the restore password part, we dont have the Token at that time. these functions dont need to be under the token middleware
4	/users/categories/:id	GET	user_id, Token	returns user's categories: userCategories[]	Using GET because a resource from server is needed and there isnt any wish to change the state of the server	
5	/users/add	POST	firstName, lastName, city, country, username, password, email, q_restore_password[], a_restore_password[], categories[]	True: string: msg_success, False: msg_failure	using POST because we want to create another user in DB with all necessary details	
6	/users/login/authenticate/	POST	Username, Password	true: Token False: failed_msg	Login cannot pass in GET True= success False=fail	

7	/users/favoritesPoints/userId/:id	GET	user_id,Token	True: Returns all favorites points of user: favoritesPoints[], False: Returns failure_msg	Using GET because a resource from server is needed and there isnt any wish to change the state of the server	
8	/users/favoritesPoints/add	POST	user_id, order_num, point_id,Token	True: string: msg_success, False: msg_failure	using POST because we want to create another point of interest in the users favorites, in DB	
9	/users/favoritesPoints/remove/userId/:id/pointId/:id	DELETE	pointId,userId, Token	True: string: msg_success, False: msg_failure	using DELETE because we want to delete the requested point from the favorites points of interests list of the user.	
10	/users/favoritesPoints/update/	PUT	pointId, orderNum, userId,Token	True: string: msg_success, False: msg_failure	using PUT because we want to update the order of the favorites points of interests of the user	
11	/users/favoritesPoints/2Latest/userId/:id	GET	userId,Token	returns 2 latest favorite points : latest2SavedPoints []	Using GET because a resource from server is needed and there isnt any wish to change the state of the server	
12	/users/qaRestorePassword/:id	GET	id	Returns all restored password questions with user_id: restoredPassword Questions[]	Using GET because a resource from server is needed and there isnt any wish to change the state of the server	
13	/users/qaRestorePassword/check	POST	id, question, answer	True: username, password. False: failure_msg	using POST because if the answer is correct then the password is sent to the user and we want to send it in the body of the request for security reasons	

14	/users/validToken	POST	-	True: success_msg, payload of the token. false: failure_msg	using POST because if the result is true then the username along with the id of the user is sent to the clientside and we did it for security reasons	we added this, so that we could check that the token is valid or not using the middleware we already had. its for keeping the user connected of the token as long as the token is valid.
15	/pointsOfInterests/	GET	-	True: Returns all points: PointsOfInterests[] false: error_msg	Using GET because a resource from server is needed and there isnt any wish to change the state of the server	
16	/pointsOfInterests/id/:id	GET	pointID	Returns spcific point or error_msg	Using GET because a resource from server is needed and there isnt any wish to change the state of the server	
17	/pointsOfInterests/popularsInCategory/: category	GET	category	all popular points from the category or error_msg	Using GET because a resource from server is needed and there isnt any wish to change the state of the server	
18	/pointsOfInterests/category/:category	GET	category	Returns all points with the specified category: categoryPoints[] or failed_msg	Using GET because a resource from server is needed and there isnt any wish to change the state of the server	
19	/pointsOfInterests/addView	PUT	point_id	string: success_msg or error_msg	using PUT because we want to update the value of views of the point.	
20	/pointsOfInterests/name/:name	GET	point_name	specific point or error_msg	Using GET because a resource from server is needed and there isnt any wish to change the state of the server	

21	/pointsOfInterests/populars/	GET	-	True: points with rating>=80% False: err_msg	Using GET because a resource from server is needed and there isnt any wish to change the state of the server	
22	/pointsOfInterests/addPoint	POST	name, category, rating, description, picture	string: success_msg or error_msg	using POST because we want to create a new point of interest	
23	/reviews/userId/:id	GET	user_id,Token	Returns all reviews of specefic user: reviews[]	Using GET because a resource from server is needed and there isnt any wish to change the state of the server	
24	/reviews/add/rate	POST	userId,pointId, rate,Token	True: string: msg_success, False: msg_failure	using POST because we want to create another review on a point of interest in DB	
25	/reviews/add/reviewMsg	POST	userId,pointId, reviewMsg, reviewDate, Token	True: string: msg_success, False: msg_failure	using POST because we want to create another review on a point of interest in DB	
26	/reviews/pointId/:id	GET	point_id,Token	Returns all reviews of specefic point: reviews[]	Using GET because a resource from server is needed and there isnt any wish to change the state of the server	
27	/reviews/2Latest/pointId/:id	GET	point_id	Returns 2 latest reviews of point: latest2Reviews []	Using GET because a resource from server is needed and there isnt any wish to change the state of the server	this funcrion dont need the user to be loggedin because Guest can see the last revies as well. He just cant review by himself.
28	/categories/	GET	-	Returns all categories: categories[]	Using GET because there is no information to deliver to the server or any wish to change any information.	

29	/categories/delete/	DELETE	-	True: string: msg_success, False: msg_failure	using DELETE because we want to delete all categories from the DB	
30	/categories/delete/:category	DELETE	category	True: string: msg_success, False: msg_failure	using DELETE because we want to delete the requested category	
31	/categories/init	POST	-	True: string: msg_success	using POST because we want to create all categories in the DB from the xml	
32	/countries/	GET	-	Returns all countries: countries []	Using GET because there is no information to deliver to the server or any wish to change any information.	
33	/countries/delete/	DELETE	-	True: string: msg_success, False: msg_failure	using DELETE because we want to delete all countries from the DB	
34	/countries/delete/:country	DELETE	country	True: string: msg_success, False: msg_failure	using DELETE because we want to delete the requested country	
35	/countries/init	POST	-	True: string: msg_success	using POST because we want to create all countriesin the DB from the xml	
36	/reviews/update/rate	PUT	userId,pointId, rate,Token	True: string: msg_success, False: msg_failure	using PUT because we want to update the review message of the point.	we thought that the user can update his rate any time he wants. so we added a put request
37	/reviews/update/reviewMsg	PUT	userId,pointId, reviewMsg, reviewDate, Token	True: string: msg_success, False: msg_failure	using PUT because we want to update the review message of the point.	we thought that the user can update his review any time he wants. so we added a put request