

עבודה API-3.1

<u>ID</u>	<u>Method Name</u>	<u>HTTP Method</u>	<u>Parameters</u>	<u>Return</u>	<u>Explanation</u>
1	/Guests/Information	GET	Int point_id	'Number_of_views': int 'description': String Rating: double { Review[] }	Using Get because the information sent to the server is not private
2	/Guests/Login	POST	String username String password	If success: { Token }	Using POST because user's password is a private parameter. Retrieve users's session Token onlt if login is matched, else, error message will be thrown and no change to the database
3	/Guests/RegisterUser	POST	String firstName String lastName String UserName String City String Country String Email String question String answer String[] Categories	If error: { 'error_message' }	Using POST because the parameters we send are private and we want to write them into the database. if one or more of the parameters are invalid, error message will show.
4	/Guests/VerificationQuestions	GET	String username	If success: { 'question1': String 'question2': String }	Using Get because we will get the question by username(if exists) without any change to the database
5	/Guests/Password	POST	String username String answer	If success: { 'password: String }	Willing to retrieve user's password only if the answer is matched else, error will be thrown , without any change to the database.
6	/tokenAuthentication/Users/PopularPoints	GET	Token token	If success: { 'points': Point[] }	Using GET because we are willing to get 2 popular points, by username,

					without any change in the database
7	/Guests/RandomPoints	GET	Double rating	{ 'points': Point[] }	Willing to get 3 random points without any change in the database.
8	/tokenAuthentication/Users/FavoritePoints	GET	Token token	{ 'points': Point[] }	Using GET because we are willing to get 2 of user's last saved points, by username, without any change in the database
9	/Guests/PointsbyCategorys	GET	Category: String	{ 'points': Point[] }	Using GET because there is no information to deliver or change the database.
10	/tokenAuthentication/Users/AddFavoritePoint	POST	Token token int pointid	Number of favorites	Posting new favorite point to a user in database. If not added, error is shown.
11	/tokenAuthentication/Users/RemoveFavoritePoint	DELETE	Token token int pointid	If error: { 'error_message' }	Deleting favorite point from a user in database. If not added, error is shown.
12	/tokenAuthentication/Users/Review	POST	Token token Int pointid Review review	If error: { 'error_message' }	Posting(adding) new review to a point in the database. If not added, error is shown.
13	/Guests/Points	GET	-	{ 'points': Point[] }	Using GET because there is no information to deliver or change the database.
14	/Guests/Categories	GET	-	{ 'categories': Category[] }	Using GET because there is no information to deliver or change the database.
15	/tokenAuthentication /Users/Rate	Post	Token token Int pointid double rank	If error: { 'error_message' }	Posting(adding) new rate to a point in the database. If not added, error is shown.
16	/tokenAuthentication/Users/AllFavoritePoints	GET	Token token	{ 'points': Point[] }	Using GET because there is no information to deliver or change the database.
17	/Guests/Countries	GET	-	{ Coutries:String[] }	Reading the countries xml file

רונית קייס 318551454
בנואל דוגיימן 207521113

פונקציה 16- כשהתחלנו את חלק 3.3 הבחנו בטעות במימוש שכן הייתה חסרה לנו הפונקציה של השגת כל נקודות העיניין המועדפות של המשתמש ולכן הוספנו אותה עכשיו ועדכנו את ה API

פונקציה 17- טעינת המדינות מקובץ ה XML- לאחר שמימשנו את כל העבודה, החלטנו שפונקציה זו מקומה בצד השרת משום שזו אינה פעולה שהלקוח אמור לדעת שקיימת או בכלל אמורה להיות בצד לקוח. ה Xml נמצא בצד השרת ולכן עליו לטעון את המדינות.

והנה זוג סטודנטים חרוצים שעושים את העבודה בראיון אישי:

