|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Explanation** | **Returns** | | **Parameters** | **HTTP method** | **Method name** | ID |
| השתמשנו ב-Get מכיוון שלא נרצה לשנות שום מקור בשרת, רק לקבל מהשרת פרמטרים.  יתר על כן – אין חשיבות להעביר פרמטרים בצורה מאובטחת. | JSON | | - | Get | readXml | 0 |
| המערכת מספקת פונקציונאליות של הרשמה- פונקציה אשר מקבל את פרטי המשתמש/ת, ואת תשובה לשאלת האימות , ומחזירה ערך אמת אם המשתמש/ת הצליח/ה להירשם. אחרת, הפונקציה תחזיר שקר.  אנו נשתמש ב-POST כי אנו משנים מקור מסוים בשרת. | String | | Username,  Password,  Firstname,  Lastname,  City,  Country,  Email,  Category,  Answer1,  Answer2 | POST | Register | 1 |
| בהינתן שם המשתמש והסיסמה המערכת בודקת האם במאגר קיימים שם משתמש וסיסמה כדלקמן – במידה וכן, המערכת תחבר את המשתמש אל המערכת ותחזיר ערך אמת. במידה ולא, המערכת תחזיר שקר.  נשתמש ב-POST כי נרצה להעביר את הפרמטרים בצורה מאובטחת. | String  Or  JSON | | Username  Password | POST | Login | 2 |
| המערכת תאפשר ללקוח לאחזר את סיסמתו – לאחר ששלח למערכת את הtoken שלו. המערכת תציג ללקוח שאלה – במידה והלקוח יענה נכון על השאלה סיסמתו תוצג על המסך לאחר שהפונקציה החזירה את המחרוזת המייצגת את הסיסמה. אחרת – הסיסמה אינו תוצג, ותוחזר מחרוזת ריקה.  נשתמש ב-POST כי נרצה להעביר את הפרמטרים בצורה מאובטחת. | String  Or  JSON | | token,  Answer1,  Answer2 | POST | GetPassword | 3 |
| בהינתן token, הפונקציה תחזיר את כלל הפרטים אשר משויכים לאותו משתמש המערכת.  השתמשנו ב-Get מכיוון שלא נרצה לשנות שום מקור בשרת, רק לקבל מהשרת פרמטרים.  יתר על כן – אין חשיבות להעביר פרמטרים בצורה מאובטחת. | String  Or  JSON | | token | POST | GetUser /accounts | 4 |
| הפונקציה תחזיר את מספר הנקודות אשר המשתמש הוסיף לרשימת הנקודות האהובות שלו – אורך הרשימה, לאחר שהמשתמש שלח לשרת את הtoken המתאים.  השתמשנו ב-Get מכיוון שלא נרצה לשנות שום מקור בשרת, רק לקבל מהשרת פרמטרים.  יתר על כן – אין חשיבות להעביר פרמטרים בצורה מאובטחת. | JSON | | token | GET | /accounts GetCounterPointsOfUser | 5 |
| בהינתן הנקודה, הטוקן [המזהה של המשתמש] והמחרוזת הרצויה, המערכת תעדכן את ההערה אשר קיימת בבסיס הנתונים.  שימוש ב-PUT לצורך עדכון פרטים. אין –יתר על כן | Status  Or  String | | Token,  Comment,  PointID | PUT | AddComment/accounts | 6 |
| לאחר ביצוע דירוג מספרי [בין 1 ל-5] של משתמש רשום כלשהו באתר לנקודה כלשהי, המערכת תשקלל את הציון של הנקודה באחוזים ע"פ כלל הציונים אשר קיבלה המערכת באתר.  השתמשנו ב-PUT מכיוון שרצינו לעדכן את מסד הנתונים.  יתר על כן – אין חשיבות להעביר פרמטרים בצורה מאובטחת. | Status  Or  String | | PointID,  integer,  token | PUT | AddRank/accounts | 7 |
| המערכת תציג ללקוח 3 נקודות עניין רנדומליות ופופולריות מעל דירוג מסוים [ערך קבוע], מתוך מאגר קבוע של נקודות פופולריות.  השתמשנו ב-Get מכיוון שלא נרצה לשנות שום מקור בשרת, רק לקבל מהשרת פרמטרים. | Jason | | Rank | GET | Display3RandomPopPoints/accounts | 8 |
| הפונקציה תחזיר מספר שלם אשר ייצג את מספר הפעמים אשר משתמשי המערכת צפו בנקודה זו.  השתמשנו ב-Get מכיוון שלא נרצה לשנות שום מקור בשרת, רק לקבל מהשרת פרמטרים.  יתר על כן – אין חשיבות להעביר פרמטרים בצורה מאובטחת. | Jason | | PointID | GET | getNumberofTimesViewsAtPoint/accounts | 9 |
| הפעולה תחזיר את התיאור המתאים של נקודה, מספר אשר מייצג את הדירוג המספרי המשוקלל של הנקודה באחוזים, את התמונה המתאימה לנקודה, את הקטגוריה של הנקודה ואת ממוצע בדירוגים שלה, בהינתן אותה הנקודה.  השתמשנו ב-Get מכיוון שלא נרצה לשנות שום מקור בשרת, רק לקבל מהשרת פרמטרים.  יתר על כן – אין חשיבות להעביר פרמטרים בצורה מאובטחת | Jason | | PointID | GET | getPointInfo | 10 |
| המערכת מספקת פונקציונאליות כדלקמן: בהינתן הטוקן של המשתמש, הפונקציה תחזיר את 2 הנקודות האחרונות אשר המשתמש שמר.  השתמשנו ב-Get מכיוון שלא נרצה לשנות שום מקור בשרת, רק לקבל מהשרת פרמטרים.  יתר על כן – אין חשיבות להעביר פרמטרים בצורה מאובטחת. | Jason | | token | GET | Get2LastPointSaved/accounts | 11 |
| הפונקציה תחזיר מחרוזת אשר תכלול בתוכה 2 הערות מילוליות אחרונות אשר כתבו המשתמשים באתר בנוגע לאותה הנקודה. לכל ביקורת יצורף גם התאריך בה היא נרשמה.  השתמשנו ב-Get מכיוון שלא נרצה לשנות שום מקור בשרת, רק לקבל מהשרת פרמטרים.  יתר על כן – אין חשיבות להעביר פרמטרים בצורה מאובטחת. | Jason | | PointID | GET | Get2LastReviewWithDate | 12 |
| הפונקציה תחזיר נקודה אשר המערכת מצאה אותה כמומלצת למשתמש, בהינתן הקטגוריה המומלצת של המשתמש וה-token של המשתמש.  השתמשנו ב-Get מכיוון שלא נרצה לשנות שום מקור בשרת, רק לקבל מהשרת פרמטרים.  יתר על כן – אין חשיבות להעביר פרמטרים בצורה מאובטחת | Jason | | Category,  token | GET | GetRecommendPoint/accounts | 13 |
| המערכת תאפשר פונקציונליות של הוספת נקודה אל רשימה הנקודות המועדפות של המשתמש, בהינתן הנקודה הרצויה והטוקן המזהה. במקרה וההוספה תתבצע המערכת תחזיר ערך אמת, אחרת תחזיר שקר.  שימוש ב-PUT לצורך עדכון פרטים. | Status  Or  String | | PointID,  token | PUT | AddPointToFavoriteList/accounts | 14 |
| המערכת תאפשר פונקציונליות של הסרת נקודה מרשימת הנקודות המועדפות של המשתמש – בהינתן הנקודה הרצויה וה-token של המשתמש. במקרה והמחיקה תתבצע המערכת תחזיר ערך אמת, אחרת תחזיר שקר.  שימוש ב-delete מפני שבמקרה זה נרצה למחוק מידע מהשרת. | Status  Or  String | | token,  PointID | Delete | DeletePointFromFavoriteList/accounts | 15 |
| המערכת תאפשר פונקציונליות של הסרת כל הנקודות של משתמש מרשימת הנקודות המועדפות, בהינתן הטוקן.  שימוש ב-delete מפני שבמקרה זה נרצה למחוק מידע מהשרת. | - | | token | Delete | DeleteAll/accounts | 16 |
| המערכת תאפשר פונקציונליות של הצגת כל הנקודות אשר שמר המשתמש. הפונקציה תקבל כקלט את הטוקן המזהה של המשתמש, ותחזיר רשימה של כל הנקודות אשר המשתמש סימן כנקודות אהובות.  השתמשנו ב-Get מכיוון שלא נרצה לשנות שום מקור בשרת, רק לקבל מהשרת פרמטרים.  יתר על כן – אין חשיבות להעביר פרמטרים בצורה מאובטחת. | JASON | | token | GET | getIntrestingPointsOfuser/accounts | 17 |
| בהינתן קטגוריה, הפעולה תחזיר רשימה של כלל הנקודות אשר משתייכות לאותה קטגוריה.  השתמשנו ב-Get מכיוון שלא נרצה לשנות שום מקור בשרת, רק לקבל מהשרת פרמטרים.  יתר על כן – אין חשיבות להעביר פרמטרים בצורה מאובטחת. | Jason | Category | | GET | getPointsByCategory/accounts | 18 |

**הערות לבודקת:**

1. **ניתן להבחין כי שינינו בשרת את ה – middleware: זאת מכיוון שהיינו צריכים להפריד בין משתמשים רשומים למשתמשים לא רשומים.  
   כדי לבצע את ההפרדה בין שני סוגי המשתמשים השונים הוספנו את הקידומת:   
   "/accounts" לmiddleware.**
2. **זאת ועוד, הוספנו middleware נוסף מכיוון שיש לנו בשרת 2 מודולים נוספים והיינו צריכים לנתב עבור כל מודול.**
3. **ניתן להבחין בכך שמחקנו חלק מהפונקציונאליות מכיוון שבעת מימוש צד הלקוח הגענו למסקנה כי המימוש של הפונקציונאליות מיותר ומחקנו על מנת לייעל.**
4. **הוספנו 2 שיטות שלא היו קודם לכן – 2 ראוטרים:   
   א. readXml- מכיוון שלאחר חשיבה רבה הבנו כי עדיף לממש את הקריאה מקובץ ה-XML בצד השרת ולא בצד הלקוח, כפי שתכננו תחילה.**

**ב. DeleteAll- אמנם לא נתבקשנו למחוק את כלל הנקודות מרשימת המועדפים אך מחקנו את כולם על מנת לממש פונק' אחרות אשר נדרשו במערכת [הערה-שינוי צבע הכוכבים].**

**טיפה הסברים על הקוד:**

**קודם כל השרת הינו רץ בפורט 3000 .**

**הקבצים מחולקים ל2 תיקיות,serverFiles ו- clientFiles ,בחלק של הסרבר שינינו את הדברים שכתובים לעיל, בחלק של השרת בנינו אותו בsingle page application כפי שלמדנו בכיתה על ידי hashbang .**

**בצד הלקוח ישנו myCtrl שהינו הקונטרולר הראשי ויש גם את app שזה בעצם הראוטר שמנתב לכל controller , וישנו index.html שזהו הדף היחיד שמשתמשים בו בhashbang . נוסף על כך ישנה תיקיית components אשר בה יש את כל שאר ה-controllers . וישנו גם קובץ css אשר מחובר לindex , השתמשנו גם בcss inline מכיוון שהיינו צריכים להשפיע רק על אלמנטים בודדים בדף.**

**תודה ☺**