

עבודה הגשה 3 בסביבות- חלק א'

בעבודה זו, בחרנו במוצר אלבומים. **אדום** - API שירד כי לא רלוונטי יותר. **ירוק** - API שהוספנו (הסבר מדוע הסרנו/הוספנו- בסוף המסמך).

API של צד שרת מתואר בטבלה הבאה:

| ID | Method Name | HTTP Methods | Parameters | Returns | Explanation |
|----|----------------|--------------|---|--------------------------|--|
| 1 | Login | POST | {Username, Password} | bool | התחברות לא יכולה להיות מועברת בGET True = success False = fail השיטה הנ"ל מחברת משתמש למערכת |
| 2 | topFive | GET | | Returns album: Album[5] | נשתמש בGET מפני שאנחנו לא מעדכנים שום דבר, ולא שולחים שום מידע לשרת. השיטה מחזירה 5 האלבומים החמים (כל הרשומה (databasen |
| 3 | getNewAlbums | GET | | Returns albums: Albums[] | נשתמש בGET מפני שאנחנו לא מעדכנים שום דבר, ולא שולחים שום מידע לשרת. השיטה מביאה לנו את כל האלבומים שנוספו בחודש האחרון (כל הרשומה (databasen |
| 4 | registerClient | POST | {FirstName, LastName, Adress, City, Country, Phone, Cellular, Mail, CreditCardNumber, isADmin, Username, Password, favoriteCategory2, favoriteCategory1, Country} | bool | הרישום נעשה בPOST מפני שמדובר בהעברת פרמטרים רבים, כולל מידע רגיש לכן כדאי שיועבר בbody של HTTP Request True = success enter the details to the DB, False = fail (בהנחה שהוולידציה על השדות נעשית בצד לקוח) |

| | | | | | |
|---|---------------------------|------|--|---|--|
| 5 | setClientFilterCategories | POST | JSON For example: List {"Username": "a", "FilterCategoryID": "1"} | bool | קביעת הקטגוריות ללקוח נעשית ב post מכיוון שמדובר בהעברת פרמטרים רבים. True = success enter the details to the DB, False = fail השיטה נקראת במהלך רישום הלקוח למערכת במהלכו הוא מכניס את הקטגוריות שלו. |
| 6 | AvailableUsername | GET | Username | True/False | נשתמש בGET מפני שאנחנו לא מעדכנים שום דבר בשרת. השיטה מחזירה true במידה ושם המשתמש פנוי אחרת יוחזר false. שיטת עזר לוולידציה בצד לקוח. |
| 7 | getFilterCategories | GET | | Return FilterCategories[{"FilterCate gotyID, CategoryName}] | נשתמש בGET מפני שאנחנו לא מעדכנים שום דבר, ולא שולחים שום מידע לשרת. השיטה מחזירה את כל הקטגוריות לפיהם ניתן לסנן אלבומים. (לדוגמא: פופ,רוק וכו') השיטה מחזירה שם ומספר מזהה של בקטגוריה |
| 8 | filterByCategory | GET | FilterCategoryID | Returns albums: Albums[] | נשתמש בGET מפני שאנחנו לא מעדכנים שום דבר. השיטה מחזירה אלבומים לפי הסינון רצוי. (כל הרשומה מ-database) |
| 9 | getSortCategories | GET | | Return SortCategories: [{"SortCategoryID, CategoryName}] | נשתמש בGET מפני שאנחנו לא מעדכנים שום דבר, ולא שולחים שום מידע לשרת. השיטה מחזירה את כל הקטגוריות לפיהם ניתן למיין את האלבומים. |

| | | | | | |
|----|-------------------------|-----|----------------|-----------------------------|---|
| | | | | | (לדוגמא: מחיר, שם אלבום, שם מבצע) השיטה מחזירה שם ומספר מזהה של בקטגוריה. |
| 10 | sortByCategory | GET | SortCategoryID | Returns albums: Albums[] | נשתמש בGET מפני שאנחנו לא מעדכנים שום דבר. השיטה מחזירה אלבומים ממוינים לפי הקטגוריה הרצויה. |
| 11 | getRecommendedPerClient | GET | Username | Returns albums: Albums[] | נשתמש בGET מפני שאנחנו לא מעדכנים שום דבר. השיטה מחזירה אלבומים מומלצים לפי עדיפות לקוח. (כל הרשומה מ-database) |
| 12 | getSearchCategories | GET | | Return category: Category[] | נשתמש בGET מפני שאנחנו לא מעדכנים שום דבר, ולא שולחים שום מידע לשרת. השיטה מחזירה את כל הקטגוריות לפיהם ניתן לחפש את האלבומים |
| 13 | searchByCategory | GET | category | Returns albums: Albums[] | נשתמש בGET מפני שאנחנו לא מעדכנים שום דבר. השיטה מחזירה אלבומים העונים לחיפוש לפי קטגוריות שהמשתמש בחר |
| 14 | SearchByGenre | GET | Genre | Returns albums: Albums[] | נשתמש בGET מפני שאנחנו לא מעדכנים שום דבר. השיטה מחזירה את כל האלבומים העונים לשאילת המשתמש |
| 15 | SearchByArtistName | GET | string | Returns albums: Albums[] | נשתמש בGET מפני שאנחנו לא מעדכנים שום דבר. השיטה מחזירה את כל האלבומים העונים לשאילת המשתמש |
| 16 | searchByAlbumName | GET | string | Returns albums: Albums[] | נשתמש בGET מפני שאנחנו לא מעדכנים שום דבר. השיטה מחזירה את כל האלבומים העונים לשאילת המשתמש |

| | | | | | |
|----|----------------------|------|---|--|--|
| 17 | getAlbums | GET | | Return albums: Albums[] | נשתמש בGET מפני שאנחנו לא מעדכנים שום דבר. השיטה מחזירה את כל פרטי האלבומים במערכת |
| 18 | getAlbumInfo | GET | AlbumID | Return Album | נשתמש בGET מפני שאנחנו לא מעדכנים שום דבר. השיטה מחזירה את כל פרטים על אלבום מסוים |
| 16 | getHistoryOrdersByID | GET | Username | Return orders: Order[] | נשתמש בGET מפני שאנחנו לא מעדכנים שום דבר. השיטה מחזירה את כל הזמנות שהשתמש ביצע בעבר (רכישות). |
| 17 | startNewPurchase | POST | {Username:username} | True / False | אנחנו מעבירים מידע רגיש ולכן נשתמש בPOST הפונקציה הזאת נועדה ליצור עגלת קניות חדשה |
| 18 | getOrderID | POST | {Username, OrderDate} (date like: "2017-06-11 12:12:12.123") | Return orderID | אנחנו מעבירים מידע רגיש ולכן נשתמש בPOST הפונקציה הזאת נועדה להחזרת orderID. |
| 19 | addItemToPurchase | POST | {OrderID,AlbumID, Amount} | String: "add item succeed" Or error string: " no such albumID" or error string:"Amount is bigger than StokAmount" | נשתמש בPOST מפני שמדובר בהעברת מידע רב. במידה ואין כמות למוצר הנ"ל נחזיר שגיאה. כמו כן, ניתן למשתמש את כל מה שיש ברשותנו עבור אותו מוצר. |
| 20 | DeleteltemToPurchase | POST | {OrderID, AlbumID , Amount} | string "update inventory succeed" or "no such albumID" | נשתמש בPOST מפני שמדובר בהעברת מידע רב. השיטה מוחקת פריט שהשתמש מעוניין למחוק מתוך עגלת הקניות. |
| 21 | DeleteOrder | POST | OrderID | String "remove order succeed" | נשתמש בPOST משום שמדובר במידע רגיש (לא רוצים לחשוף את מספר ההזמנה). השיטה מוחקת את ההזמנה במידה ולא התבצעה רכישה. |
| 22 | getOrderDetail | GET | OrderID | Array:{AlbumID, AlbumName, ArtistName , Price, Amount, PicturePath} | נשתמש בGET כי אנחנו לא מעדכנים מידע ואין שום מידע רגיש. השיטה מחזירה את מספרי זיהוי |

| | | | | | |
|----|-------------------|------|---|---|--|
| | | | | | של האלבומים, שמות האלבומים, שמות הזמרים, את מחירם ואת הכמות שהוזמנה מכל אלבום. |
| 23 | getTotalPrice | GET | OrderID | int | נשתמש בget כי אנחנו לא מעדכנים מידע ואין שום מידע רגיש נחזיר את המחיר הכולל של ההזמנה/עגלת הקניות. |
| 24 | makePurchase | POST | {OrderID, DateDeilvery , TotalPrice} | string "update DeliveryDate succeed update TotalPrice succeed, Purchase Succeed!" | נשתמש בPOST מפני שמדובר בהעברת מידע רב. לחיצה על makePurchase מהווה הסכמה של רוכש. |
| 25 | approvalOrder | POST | OrderID | String | נשתמש בPUT מפני שמדובר בעדכון אובייקט order***. השיטה מחזירה הודעה שהרכישה הסתיימה עם מספר הזמנה ופרטיה. |
| 26 | getClients | GET | | Returns clients: Clients[] | נשתמש בGET מפני שאנחנו לא מעדכנים שום דבר, ולא שולחים שום מידע לשרת. השיטה מחזירה את פרטי כל הלקוחות |
| 27 | getSpecificClient | GET | Username | Return Client | נשתמש בGET מפני שאנחנו לא מעדכנים שום דבר. השיטה מחזירה לקוח אחד לפי Username שלו. |
| 28 | getOrdersHistory | GET | | Return Array: Orders[] | נשתמש בGET מפני שאנחנו לא מעדכנים שום דבר, ולא שולחים שום מידע לשרת. השיטה מחזירה את כל ההזמנות שקיימות במערכת |
| 29 | addAlbum | POST | {AlbumName, ArtistName, FilterCategoryID, Description , PicturePath ReleaseDate , Price , ReleaseDate , StokAmount addToSystemDate , usernameAdmin} | Bool or string if not admin: " you are not Admin, you cant remove clients" | נשתמש בPOST מפני שמדובר ביצירת אובייקט Album חדש מסוג ב ומדובר בשליחת פרמטרים רבים. True = success enter the album to the DB, Error 400=Fail |

| | | | | | |
|----|-------------------------------|--------|--|---|--|
| | | | | | השיטה מאפשרת למנהל מערכת להוסיף אלבום למערכת. |
| 30 | removeAlbum | DELETE | {AlbumsID, usernameAdmin} | Bool or string if not admin: "you are not Admin, you cant remove albums" | נשתמש בDelete מפני שמדובר במחיקה מתוך אוסף אלבומים. True = success remove the album from the DB, Error 400=Fail השיטה מאפשרת למנהל מערכת למחוק אלבום מהמערכת. |
| 31 | addClient | POST | {FirstName, LastName, Address, City, Country, Phone, Cellular, Mail, CreditCardN, isAdmin, Username, Password} | bool | נשתמש בPOST מפני שמדובר ביצירת אובייקט חדש מסוג Clients וכולל פרמטרים רגישים כגון פרטי אשראי וסיסמא. True = success enter the client to the DB, False = fail השיטה מאפשרת למנהל מערכת להוסיף לקוח למערכת |
| 32 | removeClient | POST | {Username, usernameAdmin} | Bool or string if not admin: "you are not Admin, you cant remove clients" | נשתמש בPOST מפני שמדובר בעדכון אוסף לקוחות. True = success remove the client from the DB Error 400=Fail השיטה מאפשרת למנהל מערכת למחוק לקוח מהמערכת |
| 33 | updateInventory | POST | {AlbumID, StokAmount usernameAdmin} | Bool or string if not admin: "you are not Admin, you cant update inventory" | נשתמש בPOST מפני שמדובר בעדכון מלאי האלבומים. True = success update the inventory in DB, Error 400=Fail השיטה מאפשרת למנהל לעדכן את המלאי של מוצר מסוים. |
| 34 | getInventory | GET | | Array: [{AlbumID, AlbumName, Amount}] | נשתמש בGET מפני שאנחנו לא מעדכנים שום דבר, ולא שולחים שום מידע לשרת. השיטה מחזירה את כל מלאי האלבומים במערכת. |
| 35 | validationQuestionForPassword | POST | {Username, City, Lastname} | String: Password or "false, you are not the guy!" | נשתמש בPOST משום שמדובר במידע רב. |

| | | | | | השיטה מחזירה את הסיסמא במידה ופרטי המשתמש שהוזנו נכונים. |
|----|-------------------|-----|------------------|----------------------|---|
| 36 | getFilterCategory | GET | FilterCategoryID | {CategoryName:"R&B"} | נשתמש בGET מפני שאנחנו לא מעדכנים שום דבר, ולא שולחים שום מידע לשרת. השיטה בהינתן מספר של הקטגוריה(זאנר) מחזירה שם הזאנר. |

הסבר על הסרות/הוספות של API:

5. setClientFilterCategories - הסרנו API זה מכיוון שהבנו שניתן לשלוח את הז'אנרים האהובים על המשתמש כבר בעת ההרשמה יחד עם שאר הפרטים שהוא ממלא בJSON- חסכון בקריאות לשרת.

12. getSearchCategories - אין צורך בAPI זה מכיוון שקבענו בצד הלקוח שניתן לחפש לפי "שם אלבום" ולפי "שם מבצע". כמו כן, קטגוריות לחיפוש זה משהו שניתן לקבוע בצד לקוח, גם אם ירצו בהמשך לקבוע עוד קטגוריות חיפוש- זאת משום שבצד הלקוח ניתן לקבל את כל פרטי האלבומים ולבצע חיפוש על איזה פרטים שנרצה.

13+14+15+16. Search API - הוסר מהסיבה שהחיפוש מתבצע בצד לקוח (משום שכמו שנאמר ב12 יש ללקוח את כל פרטי האלבומים). לכן, ניתן לבצע חיפוש זה בצד לקוח בצורה נוחה ומהירה (ספציפית במימוש שלנו אפשרנו חיפוש לפי "שם אלבום" ו"שם מבצע").

21. DeleteOrder - הוספנו שיטה זו מכיוון שבבסיס הנתונים אנו שומרים שתי טבלאות; אחת המפרטת את כל האלבומים שהוזמנו פר הזמנה. וטבלה נוספת המתעדת את כל ההזמנות. קודם לכן היה API שרק מוחק אלבום מהזמנה. אך שמנו לב שיש צורך בשיטה שגם מוחקת את ההזמנה מטבלת ההזמנות במידה והמשתמש לא ביצע רכישה.

25. approvalOrder - הסרנו API זה מכיוון שהAPI makePurchase כבר מחזיר String עם תשובה האם הרכישה הצליחה. לכן, אין צורך בו ולכן הסרנו אותו.

31. AddClient - הסרנו API זה מכיוון שאין צורך בו. ניתן לעשות שימוש בAPI registerClient אשר מבצע אף הוא הוספה/הרשמה של משתמש חדש למערכת.

36. getFilterCategory - הוספנו API זה מכיוון שהוא מסייע לנו בהבאה של שם הז'אנר. כאשר אנו מביאים פרטי אלבום מסוים חוזרת הרשומה כפי שהיא מופיעה בבסיס הנתונים- מופיע שם רק מספר הז'אנר. לכן כאשר אנחנו רוצים להציג את פרטי האלבום איננו רוצים שיופיע מספר ז'אנר אלא שם הז'אנר- לכן נשתמש בAPI זה.