|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| EXPLANTION | RETURNS | PARAMTERS | HTTP Method | Method Name | Id |
| Cannot pass by Get because it is not secure return if the user is validate (is identify) in the server: if yes , true=success ,else: false=fail.   * באם התקבל ) fail כתוצאה מחוסר אימות , או משתמש לא קיים במערכת שרת ) מוחזר מהשרת , אז הלקוח בהתאמה ידע להציג את המילה "guest " לview בלקוח אל המשתמש , אחרת (true ) , הוא יציג את הערך האחרון שמילא המשתמש (בצד לקוח ) כ"שם המשתמש " החוקי שלו (זו הנחה , שאינה דורשת להחזיר כתשובה מהשרת את השם משתמש שניכנס למערכת, אלא זה כבר שמור בצד לקוח בעת מילוי הפרטי logIn . ) | User json or false if password or user name not god | Username, password, | Post | Login | 1 |
| Cannot pass by Get because it is not secure. Post is write a new user in the system (and the server) . If the user is registered now in the server: if registration Process Success so, return true, else: false=fail.   * מניחים : יש שדה id של משתמש לצורך אימות פרטים משמעותיים לצורך קנייה עתידית . * lastEntry- זו חותמת זמן של כניסה בצד הלקוח – שיש לשמור בצד השרת ( ניתן למקרה הצורך באבטחה שהם יסכימו על סנכרון מסוים בינהם לגבי פרטי כניסה וזמנים ) * ייתכן כי יוחזר מהשרת ערך שלילי, במקרה שיש חוסר אפשרות לרישום משתמש –כמו :סף משתמשים רשומים לאתר, או לקות של נתון באחד השדות, ערכים חוזרים ולא אפשריים וכו'. | False- if not succeed( in fault)  True- in success  Exist- if already exist | Username ,First name , last name , address , city, country , phone , mail ,credit card number ,gender , password, lastEntry | Post | Register | 2 |
| Cannot by Get because its not secure. returns value is the password (in cipher text)   * אנו מניחים כי בצד לקוח המשתמש ישאל בשאלות (שהן מוסכמות על השרת והלקוח מראש בעת אחזור סיסמה ) והתשובות עליהן הן בהכרח כל הפרמטרים: , country ,city , phone, address – אשר מוסכמים מראש על שני הצדדים, כלומר השרת יודע שהוא הולך לקבל עבור אחזור סיסמה את הפרמטרים הללו * ערך המחרוזת החוזרת מכילה את הסיסמה המאוחזרת מהשרת והיא מוצפנת וסופית – עבור הכניסה הבאה של המשתמש למערכת. * אם הפרמטרים שנשלחים לשרת, אינם תואמים לאימות השרת את המשתמש, אז השרת יחזיר מחרוזת אשר מכילה הודעה ש"לא ניתן לאמת את המשתמש " והלקוח יצטרך לבדוק זאת ולדעת אם מדובר בסיסמה מאוחזרת או הודעת שגיאה . | False if wrong input  else return the password | User name , country ,city , phone, address | Post | restore Password | 3 |
| Returns top five items list for new users, we using Get because there is no information to deliver/change to/in the server. Items are related to the popularity of them, not to the client.   * מניחים שהמוצרים הכי חמים, אינם תלויים בנתוני המשתמש הרשום במערכת , ולכן אין פרמטרים על המשתמש שנישלח לשרת . * נניח כי הצד לקוח שומר את הפרטים הללו שמקבל מהשרת אצלו. | [5] items | - | Get | HotFive | 4 |
| Returns new items from last month when login is made , we using Get because there is no information to deliver/change to the server   * מניחים שהמוצרים החדשים , אינם תלויים בנתוני המשתמש הרשום במערכת , ולכן אין פרמטרים על המשתמש שנישלח לשרת. * נניח כי הצד לקוח שומר את הפרטים הללו שמקבל מהשרת אצלו. * נניח כי המוצרים החדשים יכולים להיות ממספר קטגוריות שונות. | List<> items | - | Get | NewItemsfrom LastMonth | 5 |
| Returns the last time the user was connected when login is made , we using Get because there is no information to deliver/change to/in the server   * הפונקציה נועדה לאימות מוסף המקנה בטיחות שימוש גבוהה יותר : מניחים כי הערך שחוזר Date הינו ערך הפרמטר שנישלח לשרת בעת חיבור לאתר של המשתמש הרשום , והערך הזה חוזר , לשם השוואה בין ערך המוחזר הזה לבין הcookie ( שמניחים שבהכרח קיים cookie בלקוח) – לצורך מנגנון אבטחה יותר בטיחותי בלבד, המסתמך גם על נתוני השרת ולא רק על הcookies (אלא בנוסף!) . * פקודה זו אינה חובה כלל ( בהכרח רק אם לא נבצע את השימוש בשדה " התחברות לאחרונה" שנשלחת בעת התחברות לאתר בפקודה API id=1 ) , כל זאת בהנחה שצד לקוח מנהל את מנגנון המלא עם הCookies והוא מספק די מספיק לאימות בטיחותי רק בצד לקוח , ללא צורך בשרת כלל , ולכן אין צורך בפקודה זו במקרה הזה. | False if user not exist in the system  Else return Date | User name | Get | LastConnected | 6 |
| מחזיר רשימה של כל הקטגוריות האהובות של המשתמש (מניחים שהקטגורויות שהמשתמש בוחר הן אלו שאהובות עליו ) – נועד לצורך המלצה | List[] favorite categories | Username | get | /recommendedItemsForUser/GetUserCategories | 7 |
| מוסיף עוד מוצרים הכי מומלצים (לפי כמות מכירות גבוהה) לפי הקטגוריה אהובה על המשתמש | List[] items | Username , categoryId | get | /recommendedItemsForUser/AddMoreRecommendedItemsByCategory | 8 |
| תבדוק האם המשתמש כבר סימן שהוא אהב את קטגוריה מסוימת | Boolean | Username , categoryId | get | /SetFavoriteCategory/CheckExistCategoryId | 9 |
| הוספת קטגוריה אהובה למשתמש   * מניחים כאן כי בהכרח לא קיים categoryId עבור אותו משתמש , בטבלת users\_categories . * לפני שימוש בפקודה זו – השרת מניחים כי הקלט הינו בהרכח תקין מן הלקוח וכי לפני ביצוע פקודה זו נעשית הפקודה id=9 | Boolean | Username , categoryId | get | /SetFavoriteCategory/AddNewCategory | 10 |
| Returns list of items presented by One category , we using Get because there is no information to deliver/change to/in the server   * אין חובה בשיטה זו – באם אנו נבצע את עניין הסינון לפי קטגוריות אצל צד הלקוח בלבד ללא התערבות של צד שרת ( וכן ,כן תהיה חובה על שימוש בפקודה id=7 לשם הבאת מוצרים לפי קטגוריות אצלך השרת ). * נניח כי הצד לקוח שומר את הפרטים הללו שמקבל מהשרת אצלו. | []List items | category , maxProductsToReturn | get | ItemsByCatgoryget)) | 11 |
| Returns items sorted by choose Category , we using Get because there is no information to deliver/change to/in the server   * אין חובה כלל בפקודה זו , באם מניחים כי לשם המיון העתידי הצד לקוח ישתמש בפקודה id=7 , ואז יביא מהשרת את כל המוצרים לפי קטגוריות, ואז לצד הלקוח בלבד תהיה האפשרות לבצע מיון , ללא התערבות של השרת. והוא יבצע את המיון ללא הצורך בפקודה זו כלל. כי צד הלקוח בעצמו יותר נוח ונכון לדעתנו שצד הלקוח יעשה את זה כי הוא בעצמו יכול (לשמור את הרשימה שמקבל מהשרת בתוכו ) להסתכל על כל רשימה שמגיעה מהשרת בפקודה id=7 ואז לבצע על פר רשימה מיון עצמאי, לפי הקטגוריה שהמשתמש מבקש בview וזה יהיה מיידי כי מניחים השרת כבר מביא רשימות מוצרים פר קטגוריה , ורק נותר למיין אותם. * נשתמש בפקודה זו אך ורק , אם אין ביכולת הצד לקוח לבצע מיון על רשימת מוצרים בודדה ששייכת לאותה קטגוריה. | List<> items | Parm(like size , date) | Get | ItemSortBy(*Size or added item date*  ) | 12 |
| Returns item by Id to display for the user , we using Get because there is no information to deliver/change to the server   * נניח כי רק בגלל העניין שבצד שרת המידע על הפריט מעודכן יותר מבחינת המלאי שלו( וגם התיאור שלו ) מאשר צד לקוח, אז רק אז חייב את הפונקציה הזו. * בגלל שמדובר במנשק ספציפי של חיפוש, ייתכן כי בצד לקוח אין ברשותו את פרטי הפריט הספציפי "שנלחץ ע"י המשתמש" שמורים בו ממש , והוא רק מציג מתוך המידע מהשרת שקיבל עד כה משאר הפקודות בAPI (לדוגמא id=11) , את הview . ולכן מניחים שתתיכן צור לבקשה מלאה להעברת קבלת פרטיו המעודכנים ביותר של הפריט הספציפי מתוך צד השרת אל הלקוח. | Item (json) | Item Id | Get | ItemById | 13 |
| * קבלת פרטי מוצר , ע"י מתן שמו בלבד אל השרת | Item (json) | name | get | ItemByName | 14 |
| Add a specific item to the current user cart, we using Post because there is change to make in the server. To Save the new item of the user wants to his cart. : if add Process Success so , return true, else: false=fail.   * נניח כי העגלה היא זהה ואחת עבור לאותו התחברות יחיד של משתמש למערכת , ובעצם הוא יכול למלא אותה בצד לקוח , דרך צג החיפוש . או דרך מנשק הקניות . (הייתה התלבטות אם למחוק את ההעגלה בעת התנתקות – בסוף הוחלט שלא ) – אך בעת קנייה של העגלה – אז העגלה נמחקת , ופרטי הקנייה נשמרים במערכת בצד שרת. * אנו מניחים כי רק בעת קנייה של עגלה – המלאי מתעדכן בהתאם. אחרת , אין שום עדכון של המלאי ( גם בהוספת פריט לעגלה ) . * נניח \_(בסוף הוחלט שלא ): כי אם המשתמש מתנתק מהמערכת ומתחבר שוב , השרת מוחק את נתוני עגלת הקניות של ההתחברות הקודמת. * נניח כי השרת צריך לשמור את המוצר עבור המשתמש (הקיים כבר במערכת כמשתמש ) , אשר נוסף לעגלת קניות שלו . | Boolean | cartId , item id ,amount | get | AddItemToCart | 15 |
| Return a array list of a cart by 2 categories (parameters) and display it's to the user as required , we using Get because there is no information to deliver/change to the server   * נניח שהשרת בהכרח יחזיר את כל המוצרים אשר נמצאים בעגלת הקניות ששמורה על המשתמש הספציפי בשרת , ונניח כי צד הלקוח שומר את רשימת פרטים אלו על המוצרים שהוחזרו, ואז הצד לקוח בלבד מבצע את עניין הסינון עפ"י 2 מאפיין (פרמטרים : paramter1,paramter2) – בהנחה כי לצד לקוח יכולת להיכנס לפרטי הפריט ,ולתת הצגה לפי סינון עפ"י שני פרמטרים (נגיד צבע וגודל חולצה ) אך עדיין להציג את כל רשימת המוצרים בעגלת הקניות – כלומר בשלב הצגת הסינון- אין צורך בהתערבות הצד שרת, והאחריות לסינון על צד הלקוח . * ניתן להניח שהשרת יעשה בעצמו את החישוב של סך העלות של העגלת קניות , ויחזיר את העלות עם רשימת המוצרים בעגלה, או לחילופין : ניתן להניח כי את הסכום של כל הפריטים יחדיו , מחשבת באופן עצמאי הצד לקוח בלבד , ללא התערבות של השרת , בעת קבלת כל רשימת המוצרים , עם המחירים המעודכנים כולל כמויות מכל מוצר ( ואז אין צורך בשדה הנוסף המוחזר בפקודה זו של Long number(represent the price) ) | []List<> cart items, Long number(represent Global Price of cart ) | Cart id | Get | ViewCurrent ItemsInCart withGlobal Price | 16 |
| Returns item by Id to display for the user , we using Get because there is no information to deliver/change to the server   * פקודה זו אינה הכרחית (חובה) , באם מסתמכים על תוצאת הפקודה id=14 (ובהנחה בהכרח שצד לקוח שומר בתוכו את רשימת הפרטים (כולל הפירוט שלהם) בעגלה , וכל עוד עניין העדכניות של פרטי המוצר הינם יותר זניחים . שוב נניח כי פרטי המוצר עדכניים יותר בשרת, ולכן אנו רוצים בעת קבלת פרטים על מוצר ספציפי , לקבל את הפרטים המעודכנים ביותר מהשרת , גם אם המוצר הוא חלק מעגלת קניות. בסופו של דבר, פקודה זו אינה הכרחית , בעת זניחות על עדכניות המידע וכל עוד התבצעה פקודה id=16 . רק אז הצד הלקוח יכול לתת פרטים על המוצר ללא התערבות של השרת. | details about Item (object item include his description) | Item Id | Get | Get Update Details Of Items In Cart | 17 |
| Add a specific item to the current user cart, we using Post because there is change to make in the server. To Save the new item of the user wants to his cart. : if add Process Success so , return true, else: false=fail   * אנו צריכים את פקודה זו לשם תפעול תקין של הפקודה id=16 . * נניח כי בעת הוספת/הסרה מוצר לעגלת קניות, אין תהליך אימות בשרת שמוודא שיש את אותה כמות מספקת של אותו מוצר במלאי. נניח כי בדיקת המלאי , מתבצעת רק בזמן הקנייה עצמה. * אנו מניחים כי עגלת הקניות המעודכנת לפי המשתמש , היא נשמרת בצד השרת . ולכן עניין העגלת הקניות מתנהל דרכו . | Boolean | UserName , productId, amountOfItemToAdd | Post | AddItemToCart | 15 |
| Boolean | CartId, productId, AmountOfItemToRemove | Post | RemoveItem FromCart | 18 |
| Return list of previous orders (by id's orders) , we using Get because there is no information to deliver/change to the server.   * אנו מניחים כי כל ההזמנות שבוצעו קודם לכן שמורות בצד השרת ולא אצל הלקוח . * השרת מחזיר את כל פרטי ההזמנות שבוצעו | List<> previous orders)+amount of prev orders | User name | Get | GetPreviousOrders | 19 |
| Return specific order details by id order ,to display for the user , we using Get because there is no information to deliver/change to the server   * אנו מניחים כי כל ההזמנות שבוצעו קודם לכן שמורות בצד השרת ולא אצל הלקוח . * השרת מחזיר את פרטי ההזמנה לפי הid שלה. | Order object(include his description | Orderid , userName | Get | GetPreviousOrderDetails | 20 |
| Return the exactly amount of the item in the inventory. Get because there is no information to deliver/change to the server.  {If amount return is lowest from the amount\* , so in the client side ( (the client will check that existence of this condition and) will message the user about the absence of the item. Before the user doing the Buy }   * נניח כי פעולה זו מבוצעת לפני ביצוע הבקשה של המשתמש לקנות את כל המוצרים בעגלת הקניות. לפי ביצוע פקודה id=23 . בו צד הלקוח רוצה לדעת בעצמו , האם הפריט המבוקש ע"י המשתמש לשם קנייה , יש את הכמות המעודכנת במלאי לפי המצוין בשרת על הפריט המסוים . | Item description +Double amount of the item in the inventory | productId | Get | GetAmountOfItemInInvetorey | 21 |
| יצירת הזמנה חדשה  הנחות:   * + בעזרת numOfPrevOrders , השרת קובע את ערך הorderId עבור המשתמש ( עניין זה יכול להיעשות גם בצד הלקוח ולא בצד השרת ).   + DateOfDeliver—ערך זה אנו מניחים כי הוא נקבע בצד הלקוח בלבד, ללא צורך בשרת – בעצם הלקוח יאפשר בחירת תאריכים שהם מתוך לוח השנה , החל משבוע ומעלה (מהתאריך הנוכחי ) , והשרת יקבל את התאריך הזה ועבורו זה, יהיה מתאים לו התאריך שיבחר בהכרח בעוד שבוע ומעלה מהתאריך הנוכחי (זו ההנחה) – באם נחשוב בעתיד לשנות זאת נשנה . | boolean | username ,TypeOfMatbea , GelobalPrice\_total \_cost , DateOfDeliver ,numOfPrevOrders | post | Create New Order | 22 |
| בדיקה האם יש מספיק מלאי מאותו ה- מוצר | Item descripition+amount left after but in the inventory | CartId, itemId | Get | payOrder/CheckEnoughAmountInInvertory | 23 |
| עידכון מצב מלאי למוצר   * מניחים כי לפי הפקודה id=23 , בוצעה בדיקה מלאי, וידוע ללקוח כמה נישאר מאותו מוצר במלאי אחרי הקנייה . ולכן – אפשר לעדכן את הכמות החדשה ( לאחר הקנייה ) בתוך פרטי (המוצר ) המלאי אצל השרת . | boolean | productid , leftInInvertory | post | payOrder/UpdaetTheCureentAmuntInInvertory | 24 |
| יצירת קבלה עבור קניית מוצר   * שמירת פרטי הקנייה בצורה מפורטת (בדומה לכמו שהיה בעגלת הקניות ) = הכוללת את סוג המטבע שנבחר לקנייה , את הסכום המשולם עבור כל פריט, את הכמות מכל פריט, את המוצר , ושמירה : לפי מפתח : שם המשתמש , ומספר ההזמנה שבוצעה ומספר פריט. | boolean | OrderId,username,productid,amount,totalSum,price,,TypeOfMatbea | Post | AddNewSpecificationOrderAndPay | 25 |
| \*Update the status of the order id in the server is "approved". in Success return true , in failure return false from some reason . Post because there is information to update in the server.   * השרת שומר את פרטי ההזמנה אצלו , ומעדכן אותה בהתאם . * ברגע שהמשתמש בצד לקוח , רואה את הסכום של העגלת הקניות , הוא צריך לעבור לתהליך הקנייה – ולכן הוא חייב לאשר את ההזמנה (לפני הקנייה בפועל ) .- רק לאחר מכן הוא יוכל לקנות את העגלה . | Boolean | UserName, orderId, | Post | ApproveOrder | 26 |
| עדכון מצב מלאי | Boolean | productid,leftInInvetory | post | /PayOrderUpdateTheCurrentAmountInInvetory | 27 |
|  | Boolean | OrderId,username,productid,amount,totalSum,price | post | /PayOrder AddNewSpicifectOrderAndPay | 28 |
| * מעדכן במערכת שאכן בוצע התשלום עבור המשתמש . | Boolean | OrderId,userName | post | /PayOrder PayCart | 29 |
| * מניחים כי אין צורך לשמור במערכת את פרטי העגלה של כל המשתמשים, במיוחד לאחר ביצוע הקנייה ( כדי לא להעמיס יתר על המידה על בסיס הנתונים ) – ולכן פרטי העגלה נמחקים לאחר קנייה – וכן לצורך מילוי חוזר של העגלה עבור הקנייה הבאה . | Boolean | userName | post | /PayOrder/DeleteCart | 30 |

דניאל לקוח:

* ישנו קושי בלתכנן את כל המערכת וכל הפונקציות שלה מראש.
* ישנו קושי בלנסות לנסח דרישה נכונה שתייצג את הרצון שלי ולא תיתן מקום למתכנת להבין משהו אחר .
* ישנו קושי בכך שבתור הלקוח שהוא לא בעל ידע טכני, איני יודע מה אפשרי לבקש ומה לא, כמו כן הלקוח אינו חשוף לכלל אפשרויות טכנולוגיות שאולי הוא היה מעוניין להשתמש בהם.

חן- מתכנת:

* ישנו קושי כאשר יש דרישות לא ברורות ואז בתור המתכנת אני אמור לנסות לנחש מה רצה הלקוח
* לא כל דרישה שהציב הלקוח היא אפשרית מבחינה טכנית
* ישנן דרישות שמשפיעות אחת על השנייה ולכן לפעמים צריך לחזור אחורה ולעדכן פונקציות ישנות.

הנחות :

* המערכת שלנו מתעסקת במוצר " חולצות "- קטגוריות הם : צבע, שם מותג , גודל וכו'.
* נניח שאת עניין הצגת הלינקים הוא רק בצד client , ואין צורך של התערבות בצד שרת (כמו שמירת תכני קבצי האתרים שלנו בפועל ) . יש אישור מארז.
* נניח כי בכל item יש שדה category , אשר לפיו ניתן לבצע סינון בצד הלקוח בלבד, ללא התערבות צד השרת .
* נניח כי מאשר והשרת מבחינתנו מחזיק את הפרטים על המוצרים לפי קטגוריות , אז הוא גם מחזיק את כמות המלאי המעודכנת שיש מאותו פריט . והוא השרת דואג לעדכן זאת בעת ביצועי קניות אצלו, או ספירות מלאי באופן עצמאי של השרת . (ולכן כצד לקוח אנו יותר סומכים על המידע המעודכן מהשרת, אשר מתעדכן בצד לקוח בחלק משימוש הפקודות הנ"ל- ולרוב נניח כי צד שרת יותר מעודכן בנתוניו מאשר הצד לקוח כי הוא בפועל הראשון שמתעדכן , וכעבור זמן מה בשימוש הפקודות הנ"ל של הAPI אז גם צד לקוח מתעדכן (אבל בזמן אחרי עדכון השרת). .
* נניח שהשרת stateless, אך הוא שומר על כל פרטי המוצרים המעודכנים, עגלת הקניות ,ושמירת פרטי ההזמנות שבוצעו במערכת פר משתמש .
* מניחים תמיד כי את רוב בדיקות הקלט ותקינותיו הן יהיו על הצד לקוח !!! (והשרת מסתמך על כך ברוב הפונקציות שלו ) .
* נניח כי : הדרישה, "יש להראות תמיד את אייקון העגלה למשתמש, אך רק במקרה ולקוח כבר מחובר למערכת" – מבוצעת על פי צד הלקוח בלבד כחלק מהעובדה שהוחזר true בעת ביצוע פקודה id=1 , ואין קשר נוסף לשרת .
* אם יש חסרים במלאי זה בסדר ,מכיוון שהמערכת תדע להשלים את זה ידנית במקרה הצורך דרך מערכת חיצונית אחרת .
* המחיר הקבוע הוא שקל.

דוגמא לjson של מוצר ,יכול להיות שצריך להוסיף דברים בשביל הclient בשלב הבא,למשל תמונה .

