|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Method Name | HTTP Method | Parameters | Returns | Explanation |
| 1 | Login | POST | Login parameters - JSON: { "mail" : <string (size<50)>, "pass": <string (size<50)> } | User mail - JSON { "Mail": <string (size<50)> } ERROR: error reason<string> | Login cannot pass in GET respond with a message to the client |
| 2 | getTop5Products | GET | - | Returns 5 jason that SalesNumber is the maximum  musical instrument: {musical\_instrument[]} \*\*\* to see fields watch getPtoductDetails | Using GET because there is no information to deliver to the server  or any wish to change any information.  return the top "hot" 5 musical instrument . Product are related to the seller , not to the client. |
| 3 | registerUser | POST | User details - JSON :  { "mail": <string (size<50)>, "pass": <string (size<50)>, "fName": <string (size<50)>, "lName": <string (size<50)>, "phone": <string (size<50)>, "cellular": <string (size<50)>, "addr": <string (size<50)>, "city": "<string (size<50)>, "country":<string (size<50)>, "creditCardNum": <string (size<50)>, "isAdmin": <int>, "interest\_types": <string (size<50)>, "school": <string (size<50)>, "firstPet": <string (size<50)>, } \*\*\* interest\_types get all interest as long string sapereted with ',' . for example guitar,piano will be: "interest\_types": "guitar,piano" | Server response <string> | Register cannot pass in GET respond with a message to the client |
| ~~4~~ | ~~getCountries~~ (moved to client responsablity) | ~~GET~~ | ~~-~~ | ~~Returns all Countries(as string): Country[]~~ | ~~Using GET because there is no information to deliver to the server or any wish to change any information. The server answer return all the countries that the site supported~~ |
| 5 | verifyUserAndRestorePass | Post | Verify details - JSON : { "mail": <string (size<50)>, "school": <string (size<50)>, "firstPet": <string (size<50)> } | Password- JSON : {  "password": <string (size<50)> } | verify by 2 question to the user , Using POST because there is information to deliver to the server returning the user password if the information is verify,  returning empty string if the information is wrong. The information on the privates question will saved on the server - not in client |
| 6 | latestProducts | GET | - | Returns last 30 day upload musical instrument as JSON: {musical\_instrument[] (all fields)} \*\*\* to see fields watch getPtoductDetails | Using GET because there is no information to deliver to the server or any wish to change any information. returning all last uplouded product from last 30 days. Data on uplouds is on server |
| 7 | getAllProducts | GET | - | Returns all products (in list of categories) as JSON array: {musical\_instrument[] (all fields)} \*\*\* to see fields watch getPtoductDetails | Using GET because there is no information to deliver to the server or any wish to change any information. the product in relate with seller and not with client. returning all the products sorted by category. |
| 8 | getMatchingProducts | Post | userMail- JSON :  {  "mail": <string (size<50)>  } | returns array of recommended Product details - JSON : {musical\_instrument[] (all fields)} \*\*\* to see fields watch getPtoductDetails | Using POST to deliever the client mail so the server will  know which user to match products to. reffering to its interest types and products he bought in the past already. |
| 9 | getProductDetails | POST | profuctID - JSON : { "instrumentID": <string (size<50)>, } | Product details - JSON : { "Musical\_instrument": <string (size<50)>, "Manufactur": <string (size<50)>, "Year": <int >, "Description": <string (size<50)>, "PicturePath": <string (size<50)>, "PublishDate" : <datetime>  "Price": <int> , "StockAmount": <int>, "Delivery\_time": <int>, "SalesNumber": <int> } | Using POST to deliever the server the needed instrumentID the client wants to receive details at. give the user more details on the product  , like inventory information etc the infotmation is real time update and server request is requiered |
|  | getProductsInOneOrder | POST | one orderID- JSON : { "orderID": <int>, } | returns array of products ID and their amount bought int specific order - JSON: [{ "OrderID": <int> "ProductID": <string (size<50)> "Amount":<int> },...] | Using POST to deliever the server the desired orderID so  it will be able to deliever back the products bought in that order and their amount. |
| 10 | getOrderHistory | POST | userMail- JSON : { "mail": <string (size<50)> } | returns array of general order details made by the  user with the current email - JSON:  [{ "OrderID": <int>, "ClientMail": <string (size<50)>, "Time": " <dateTime>, "TotalPrice": <int>, "Details":<string (size<50)> " }...] | Using POST to deliever the mail so the client will be able to receive all the orders that client made and to access  the products bought if needed  (using the getProductsInOneOrder function) return all the orders of the user. Data is on server. the client side does not save orders. |
| 11 | buyProductsInCart | POST | general details about the cart,  and array of products ID and amount - JSON: { "mail": <string (size<50)>, "totalPrice": <int>, "details": <string (size<50)>, "products": [{"instrumentID": <string (size<50)> , "amount": <int> }, ...] } | Server response <string> | Using Post to deliever the needed information to the server about the creation of orders. it create new order record with the general details and a record for each product bought and the orderID. |
| 12 | checkInInventory | POST | array of products ID and requested amount [ {"instrumentID": <string (size<50)>, "amount": <int> } ... ] | Boolean: true- if all products available false- if check failed  in case only part of the products available, return array of available instrumentID - JSON: [ { "Musical\_instrument": }... ] | Using POST to deliever the needed information to the server before creation of orders and inventory changings  there is a need to check if the desired products and  their amount is available to the client to buy,  as respond the server answers in true if the product available, false if the check  failed or sends the available products if only part of them ar available. |
| 13 | approveBuying | POST | array of products ID and requested amount [ {"instrumentID": <string (size<50)> "amount": <int> }, ... ] | Boolean: true- if update succeed false- if update failed | Using POST to deliever the product and amount that the client finished "buying", it updates the server with the current inventory after the sale, and updates the number of sales for the product. |
| בונוס | | | | | |
| 14 | getAllClients | GET | - | returns an array of all current clients in the system: [ { "mail": <string (size<50)>,  "pass": <string (size<50)>,  "fName": <string (size<50)>,  "lName": <string (size<50)>,  "phone": <string (size<50)>,  "cellular": <string (size<50)>,  "addr": <string (size<50)>,  "city": "<string (size<50)>,  "country":<string (size<50)>,  "creditCardNum": <string (size<50)>,  "isAdmin": <int>,  "interest\_types": <string (size<50)>,  "school": <string (size<50)>,  "firstPet": <string (size<50)>,  }, ...] | Using GET because there is no information to deliver  to the server or any wish to change any information. The client list is in the server DB |
| 15 | ~~LoginStaff~~ (same as login, clients responsability) | ~~POST~~ | ~~Username, Password~~ | ~~Bool~~ | ~~Login cannot pass in GET True= success False=fail~~ |
| 16 | getOrdersReport | GET | - | returns array of general order details made by all user with the current email - JSON:  [{ "OrderID": <int>, "ClientMail": <string (size<50)>, "Time": " <dateTime>, "TotalPrice": <int>, "Details":<string (size<50)> " }...] | Using GET because there is no information to deliver to the server or any wish to change any information. order is in relate with different users, the data isn't on client. |
| 17 | addProduct | POST | product Details { "instrumentID": string (size<50), "manufactur": int , "year": int , "description": string (size<50), "picturePath": string (size<50), "price": int , "amount": int, "categoryName": string (size<50) } | Bool | adding cannot pass in GET. using post because we need to update th e server on the product. True= added False=fail to add |
| 18 | deleteProduct | DELETE | productID- { instrumentID: string (size<50) } | Bool | Using DELETE because there is a data to remove from the server. |
| 19 | addUser | POST | \*\*\* to see fields watch registerUser | Server response <string> | adding user is the same as register it cannot pass in GET respond with a message to the client |
| 20 | deleteUser | DELETE | UserID- { mail: string (size<50) } | Bool | Using DELETE because there is a data to remove from the server. |
| 21 | getInventory | GET | - | Returns all productID and stock amount- [ { Musical\_instrument: <string> , StockAmount : <int>}, { Musical\_instrument: <string> , StockAmount : <int>}, { Musical\_instrument: <string> , StockAmount : <int>}, . . .  ] | Using GET because there is no information to deliver to the server or any wish to change any information. The function return the number of product for eache product  - in relate with the seller and not with the client |
| פונקציונליות שנוספה | | | | | |
| **22** | **getProductByCategory** | **Post** | **Category- { Category: string (size<50) }** | **product[]** | **על מנת להציג את המוצרים לפי קטגוריה, הוספנו פונקציה ששולפת את המידע מה-DB. בטבלת המוצרים שלנו לא החזקנו את הקטגוריה של המוצר ועל כן החזרת json של מוצר לא תיתן לנו את מידע זה, יש לנו טבלה נוספת עבור המידע של קטגוריה של כל מוצר.** |

Alon Mor 200828812

Liron Seliktar 308552900