|  |  |  |
| --- | --- | --- |
| Request / Response  \* GET all responses are represented by an example of the first object in the array | Method + Path | Description |
| Request:  {      "firstName": "Ploni",      "lastName": "Almoni",      "idcard": "123456789",      "email": "ploni@gmail.com",      "password": "almoni123",      "city": "Tel-Aviv",      "street": "Lewinsky"  }  Response:  {      "firstName": "Ploni",      "lastName": "Almoni",      "idcard": "123456789",      "email": "ploni@gmail.com",      "city": "Tel-Aviv",      "street": "Lewinsky",      "uuid": "26fa9d0b-06fb-425c-b2b1-161340a4f080",      "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjp7ImZpcnN0TmFtZSI6IlBsb25pIiwibGFzdE5hbWUiOiJBbG1vbmkiLCJwYXNzd29yZCI6ImQ5MGE0OTVjN2UyYjAwYWQxZTAwOWE2NDcxZGU0Y2JjZGVmYTEyMWM1YmQ0NTRmYWE1ODAzZGUwOTk1ZGMyZGIzYzBmMGViMDFhNmY2NzllZGEwYWI4N2UwZWEyNGJmZTY3YTBmMjJjMWU4YzZmYWY4ZTJlNDFmMDgzMWRkM2VhIiwiaWRjYXJkIjoiMTIzNDU2Nzg5IiwiZW1haWwiOiJwbG9uaUBnbWFpbC5jb20iLCJjaXR5IjoiVGVsLUF2aXYiLCJzdHJlZXQiOiJMZXdpbnNreSIsInV1aWQiOiIyNmZhOWQwYi0wNmZiLTQyNWMtYjJiMS0xNjEzNDBhNGYwODAifSwiaWF0IjoxNjM0MzcxMTk3LCJleHAiOjE2MzQzNzI5OTd9.\_yxJ85lYTzjoykMJ5SqjhdwIbeGyaG364s7OFQvUuTE",      "isAdmin": **false**  } | POST  /auth/register | Register new user |
| Request:  {      "email": "yoav.lip@gmail.com",      "password": "walla123"  }  Response:  {      "uuid": "49c7ba65-6f8f-4a73-80b3-d3d7415f1fbf",      "firstName": "YOAV",      "lastName": "LIFSHITZ",      "password": "",      "idcard": "043051069",      "city": "HARISH",      "street": "Oren 23/2",      "email": "yoav.lip@gmail.com",      "isAdmin": 1,      "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..PaSy-YQBxRAYK9jrk7lQ\_H5rlqbtZBWICULa9w1PPx0"  } | POST  /auth/login | User login |
| Response:  [      {          "CategoryID": 1,          "CategoryName": "Beverages"      }  ] | GET  /categories | Get all categories |
| Request: 6  Response:  [      {          "ProductID": 9,          "ProductName": "Mishi Kobe Niku",          "UnitPrice": 97,          "CategoryID": 6,          "imageName": "9.jpg"      }  ] | GET  /categories/:id | Get all products of a category |
| Response:  [      {          "city": "Ashdod",          "population": 220174      }  ] | GET  /cities | Get (all) 10 most populated cities, sorted by ABC |
| Request: (uuid string)  Response:  [      {          "CartID": 176,          "UserUUID": "49c7ba65-6f8f-4a73-80b3-d3d7415f1fbf",          "Date": "2021-10-15T21:00:00.000Z"      }  ]  What to do with the redux?  Filter by uuid and then get max CartId | GET  /carts/user/:uuid | Get the last cart of a user |
| Request:  {      "UserUUID": "26fa9d0b-06fb-425c-b2b1-161340a4f080",      "Date": "2021-10-16"  }  Response:  {      "CartID": 177,      "UserUUID": "26fa9d0b-06fb-425c-b2b1-161340a4f080",      "Date": "2021-10-16"  } | POST  /carts | Add new cart |
| Request: (cart id number)  Response: 1 | DELETE  /carts/:id | Delete a cart |
| Request (cart id number)  Response: 1 | Get  /items/:cartId | Get all items of (an open) cart |
| Request:  {      "ProductID": 4,      "TotalPrice": 44,      "Quantity": 2,      "CartID": "177"  }  Response:  {      "CartID": "177",      "ProductID": 4,      "Quantity": 2,      "TotalPrice": 44,      "ItemID": 389  } | POST  /items | Add new item (to cart) |
| Request: (item id number)  Response: 1 | Delete  /items/:id | Delete an item (from cart) |
| Request: (cart id number)  Response: 1 | DELETE  /items/all/:cartId | Delete all items from a cart |
| Request: (uuid string)  Response:  [      {          "OrderID": 113,          "UserUUID": "49c7ba65-6f8f-4a73-80b3-d3d7415f1fbf",          "CartID": 173,          "TotalPrice": 44,          "City": "Bené Beraq",          "Street": "Harish",          "Delivery": "2021-10-24T21:00:00.000Z",          "Date": "2021-10-15T21:00:00.000Z",          "CreditCard": "5555"      }  ] | GET  /orders/user/:uuid | Get the last order of a user |
| Response:  [      {          "count": (number)      }  ] | GET  /orders/count | Count all the orders in database |
| Response:  [      {          "Delivery": "2021-07-31T21:00:00.000Z",          "count": 3      }  ] | GET  /orders/count/dates | Get all delivery dates who are already fully booked |
| Request:  {      "UserUUID": "26fa9d0b-06fb-425c-b2b1-161340a4f080",      "Date": "2021-10-16",      "Delivery": "2021-10-23",      "City": "Haifa",      "Street": "Herzl",      "CreditCard": "5555",      "TotalPrice": 44,      "CartID": "177"  }  Response:  {      "OrderID": 116,      "CartID": "177",      "UserUUID": "26fa9d0b-06fb-425c-b2b1-161340a4f080",      "TotalPrice": 44,      "Date": "2021-10-16",      "City": "Haifa",      "Street": "Herzl",      "Delivery": "2021-10-23",      "CreditCard": "5555"  } | POST  /orders | Add an order |
| Request: Order(number).pdf  Response: (downloads file to computer) | GET  /orders/pdf/:name | Get an order pdf file from server |
| Response:  [      {          "ProductID": 1,          "ProductName": "Chai",          "UnitPrice": 18,          "CategoryID": 1,          "imageName": "1.jpg"      },  ] | GET  /products | Get all products |
| Request: un  Response:  [      {          "ProductID": 4,          "ProductName": "Chef Anton's Cajun Seasoning",          "UnitPrice": 22,          "CategoryID": 2,          "imageName": "4.jpg"      },  ] | GET  /products/search/:query | Search all products by query |
| Request:  {      "ProductName": "Bamba",      "UnitPrice": 3.5,      "CategoryID": 8  }  + image/jpeg file  Response:  {      "ProductID": 81,      "ProductName": "Bamba",      "UnitPrice": 3.5,      "CategoryID": 8  } | POST  /products | Add a product (admin only) |
| Request:  {      "ProductID": 81,      "ProductName": "Bamba",      "UnitPrice": 4,      "CategoryID": 8  }  + image/jpeg file  Response:  {      "ProductID": 81,      "ProductName": "Bamba",      "UnitPrice": 4,      "CategoryID": 8  } | PUT  /products/:id | Update a product (admin only) |
| Request: 1  Response: {      "ProductID": 1,      "ProductName": "Chai",      "UnitPrice": 18,      "CategoryID": 1,      "imageName": "1.jpg"  } | GET  products/images/:name | Get images of a product |
| Response: [  {          "idcard": "123456789",          "email": "ploni@gmail.com"      },  ] | GET  /users | Get IDs and emails of all users |
|  | No path. Server makes the call it from pdf-helper.js in order to issue the invoice/receipt after posting a new order | Get specific user's details |