|  |  |  |
| --- | --- | --- |
| End Point: “http://localhost: {{Chosen-Port}} /shopping” | | |
| Description | Method + Path | Description |
| Fetch the products categories | GET  /product\_categories | Request: No value  Response: {    productCategories:  [{id: 1, name: “Beverages”} …...],    message: `<<{icon}>> Fetching All Product Categories Process Completed Successfully <<{icon}>>`  } |
| Fetch products by their specific category + need to mention the page you need and with that it goes to the DB | GET  /products\_by\_category | Request: { categoryId: “Beverages”, page: “1” }  Response: { message: `<<{icon}>> Get Products By Category ( ID No. 17 } ) Success <<{icon}>>`,      Products: [{id:1,      name: “Arak”,      category\_id: 17,      price: “45.00”,      image: “/ArakHaela.jpg”}………..]  } |
| Get product /s  By the value user has provided in the input from the front app | GET  /search\_products\_by\_free\_text | Request: {searchedProduct: “Cola”, page: “1” }  Response: {message: `<<{icon}>> Searched Products: “Cola” - Processed Successfully <<{icon}>>`,  Products: [{id:1,      name: “Cola”,      category\_id: 1,      price: “7.00”,      image: “/coca-cola-Big-unit.jpg”}………..]  or empty array if nothing found  } |
| Opens a new cart In the db | POST  /open\_new\_cart | Request: {userId: 4}  Response: {message: `<<{icon}>> Open New Cart For Client Process Completed Successfully - Cart No. 784 <<{icon}>>`,  recentlyOpenedCart: {id: 784,      customer\_id: 355932112,      created\_at: “2022-08-06 09:40:22”}  } |
| Removes items from user’s open cart | DELETE  /delete\_from\_open\_cart | Request (values as query strings): { currentProductLocationInTable: “45”,   cartId: “56,   emptyCart: “false” } |

|  |  |  |
| --- | --- | --- |
|  |  | Response: {  message: `<<{icon}>> Delete Items From Current Cart Process Completed Successfully <<{icon}>>`,  currentProductsInCart:[{    id: 6,    name: “Chocolate Elite Cow”,    category\_id: 8,    price: “4.30”,    image: “/Chocolate Elite Cow.jpg”,    quantity: 3,    total\_price: “12.90”,    cart\_id: 89,    specific\_row\_location\_in\_cart\_products\_table: 97}………..]  ,  currentTotalCartPrice: “780.59”  } |
| Update the user’s cart by adding new items | PUT  /update\_open\_cart | Request: {    productId: 4;    quantity: 1;    cartId: 79;  }  Response: {message: `<<{icon}>> Update Current Cart Process Completed Successfully <<{icons}>>`,  currentProductsInCart:[{    id: 6,    name: “Chocolate Elite Cow”,    category\_id: 8,    price: “4.30”,    image: “/Chocolate Elite Cow.jpg”,    quantity: 3,    total\_price: “12.90”,    cart\_id: 89,    specific\_row\_location\_in\_cart\_products\_table: 97}………..]  ,  currentTotalCartPrice: “780.59”  } |

|  |  |  |
| --- | --- | --- |
| Checks if dates are valid (available) for shipment and returns them as a list | GET  /check\_if\_dates\_ok\_for\_shipping | Request: No values  Response: {message: `<<{icon}>> Fetch Approved Dates For Shipping Process Completed Successfully <<{icon}>>`,  finalGoodDatesArray: [“2022-08-05”, …..]  } |

|  |  |  |
| --- | --- | --- |
| End Point: “http://localhost: {{Chosen-Port}} /admin” | | |
| Description | Method + Path | Description |
| Adding item to DB | POST  /add\_item | Request: {name: “Whiskey Johnny Walker Red”,    categoryId: 4,    price: “55.0”,    image: “/Whiskey Johnny Walker Red.jpg”  }  Response: {success\_message: `<<{icon}>> New Item: Whiskey Johnny Walker Red, Was Uploaded To DB Successfully <<{icon}>>`} |
| Fetch the specific item by his ID | GET  /get\_item\_by\_id | Request: { productId: “432” }  Response: { message: `<<{icon}>> Selected Product: ID No. 432 - Fetched Successfully <<{icon}>>`,  selectedProduct: { id: 432, name: “Goldstar Beer”,  category\_id: 1, price: “12.70”,  image: “/Goldstar Beer.jpg”  }  } |
| Updating the Item that is currently in DB | PUT  /update\_item | Request: {id: 68,    name: “Huggies Diapers”,    categoryId: 14,    price: “28.60”,    image: “/Huggies Diapers.jpg”  }  Response: {success\_message: `<<{icon}>> Update Item No. 68 Process Completed Successfully <<{icon}>>`  } |

|  |  |  |
| --- | --- | --- |
| End Point: “http://localhost: {{Chosen-Port}} /orders” | | |
| Description | Method + Path | Description |
| Adding an order to the DB | POST  /upload\_new\_order | Request: { customerId: 45,      cartId: 566,      finalPrice: “1250.70”,      city: “Bnei Brak”,      street: “HaGolan”,      houseNumber: 43,      dateTime: “2022-08-06 09:40:22”,      last4DigitsCc: 4226  }  Response: { message: `<<{icon}>> Upload New Order Process Completed Successfully <<{icon}>>`,  newRecentUploadedOrder: {  id: 443,  customer\_id: 45,  cart\_id: 566,  final\_price: “786.50”;,  ship\_city: “Bnei Brak”,  ship\_street: “HaGolan”,  ship\_house\_number: 43,  ship\_date\_time: “2022-08-017 11:40:00”,  order\_submitted\_at: “2022-08-06 09:40:22”,  last\_4\_digits\_credit\_card: 4226 }  } |

|  |  |  |
| --- | --- | --- |
| End Point: “http://localhost: {{Chosen-Port}} /validate\_front\_token” | | |
| Description | Method + Path | Description |
| This path validates if the user’s token (from the front) is valid by checking if the sent token matches the one saved by the Jwt verify token function | GET  / | Request: { frontToken: (long alphanumeric token string) }  Response: {  message: "Ok",  isFrontTokenOk: true,  role: “customer”,  userName: “John Doe”  } |

|  |  |  |
| --- | --- | --- |
| End Point: “http://localhost: {{Chosen-Port}} /healthCheck” | | |
| Description | Method + Path | Description |
| Gives indication if the server is alive and running | GET  / | Request: No value  Response: `<<{icon}>> Server Is Running <<{icon}>>` |

|  |  |  |
| --- | --- | --- |
| End Point: “http://localhost: {{Chosen-Port}} /onInitWebPage” | | |
| Description | Method + Path | Description |
| Fetch the total number of orders in the DB | GET  /orders | Request: No value  Response: {  totalNumberOfOrders: 743,  message: `<<{icon}>> Fetching All Orders Process Completed Successfully <<{icon}>>`  } |
| Fetch the total number of products in the DB | GET  /products | Request: No value  Response: {  totalNumberOfProducts: 648,  message: `<<{icon}>> Fetching All Products Process Completed Successfully <<{icon}>>`  } |

|  |  |  |
| --- | --- | --- |
| End Point: “http://localhost: {{Chosen-Port}} /auth” | | |
| Description | Method + Path | Description |
| This path is in charge of all the login process | POST  /login | Request: {    userName: “xzc@v.coo”,    password: “Df3333,    verifiedToken: true;  }  Response: (3 options depends on request parameters)  First option – new customer: {      userId: 67,      userName: “xzc@v.coo”,      userResidence: {  city: “Bnei Brak”,  street: “HaGolan”,  house\_number:43 },      message: `<<{icon}>> Login Success  <<{icon}>>`,      isLoggedIn: true,      token: (alphanumeric string token),      role: “customer”,  isNewCustomer: true  }  Second option – user has open cart in DB: {      userId: 67,      userName: “xzc@v.coo”,      userResidence: {  city: “Bnei Brak”,  street: “HaGolan”,  house\_number:43 },      message: `<<{icon}>> Login Success  <<{icon}>>`,      isLoggedIn: true,      token: (alphanumeric string token),      role: “customer”,  cartCreatedAt: “2022-09-06 09:40:22”,    currentTotalPrice: “1032.74”,    currentProductsInCart: [{      id: 32,      name: “Sprite 1.5L”,      category\_id: 1,      price: “4.50”,      image: “/Sprite 1.5L.jpg”,      quantity: 2,      total\_price: “9.00”,      cart\_id: 183,      specific\_row\_location\_in\_cart\_products\_table: 57}……..],    openCartExist: true,    currentCartId: 166  } |

|  |  |  |
| --- | --- | --- |
|  |  | third option – user doesn’t have open cart in DB: {      userId: 67,      userName: “xzc@v.coo”,      userResidence: {  city: “Bnei Brak”,  street: “HaGolan”,  house\_number:43 },      message: `<<{icon}>> Login Success  <<{icon}>>`,      isLoggedIn: true,      token: (alphanumeric string token),      role: “customer”,  customerLastOrder: {      id: 74,      customer\_id: 54,      cart\_id: 877,      final\_price: “1544.21”,      ship\_city: “Bnei Brak”,      ship\_street: “HaGolan”,      ship\_house\_number: 43,      ship\_date\_time: “2022-09-25 10:05:22”,      order\_submitted\_at: “2022-09-06 09:40:22”,      last\_4\_digits\_credit\_card: 5521      },    openCartNotExist: true  } |
| Responsible for the first stage of the registration process | POST  /registerLevelOne | Request: {id: 3}  Response: {  message: `<<{icon}>> Submitted Customer ID Was Not Found In The System <<{icon}>>`  } |
| Responsible for the second stage of the registration process | POST  /registerLevelTwo | Request: {  id: 422506331,  userName: “eliBin22@gmail.com”,  password: “Er3351”,  confirmPassword: “Er3351”,  city: “Haifa”, street: “Ahuza”, houseNumber: 12,  firstName: “Eli”, lastName: “Bin” }  Response: {  message: `<<{icon}>> New User: eliBin22@gmail.com Registered successfully <<{icon}>>`  } |