\*\*All requests may response a JSON contains hasError Key with appropriate message to show user.

Endpoints:

Endpoint: *http://localhost/store*

|  |  |  |
| --- | --- | --- |
| **Description** | **Method + Path** | **Request / Response** |
| Get All Food Categories | GET  /category | **Request: Object** {},  **Response:** Array [ {  \_Id: 5b8172f4be3ebd36884902fb,  name: Milk & Eggs  } ] |
| Get All Products By Category | GET  /category?id='\_id' | **Request: Object** {},  **Response:** Array [ {  \_Id: 5b86e7d6e0cb1a3e6c811f01,  Image: path,  name: Eggs,  price: 25.90,  category: {  \_Id: 5b8172f4be3ebd36884902fb,  name: Milk & Eggs  } } ] |
| Create New Category | POST  /category | **Request: Object** {  name: Milk & Eggs  },  **Response:** {  \_Id: 5b8172f4be3ebd36884902fb,  name: Milk & Eggs  } |
| Create New Product | POST  /product | **Request: FormData** {  name: Vodka,  price: 20.2,  file: fileReader.event.target.result,  category: category.\_id  },  **Response:** {  \_id: 5b9036342b95672dd0074efa,  name: Vodka,  price: 20.2,  file: fileReader.event.target.result,  category: category.\_id,  \_v:0  } |
| Update Product | PUT  /product | **Request: FormData** {  \_id: 5b9036342b95672dd0074efa,  name: Vodka,  price: 20.2,  file: fileReader.event.target.result,  category: category.\_id  },  **Response:** {  \_id: 5b9036342b95672dd0074efa,  name: Vodka,  price: 20.2,  file: fileReader.event.target.result,  category: category.\_id  } |
| Filter Products By name, no mention of categories. | GET  /product/filter?name= string | **Request: Object** {},  **Response:** Array [ {  \_Id: 5b86e7d6e0cb1a3e6c811f01,  Image: path,  name: Carrots,  price: 25.90,  category: {  \_Id: 5b8172f4be3ebd36884902fb,  name: Milk & Eggs  } } ] |
| Create Or Update user Cart,  \*depends if exists and \_id send in request, new is empty string; | POST  /cart | **Request: Object** {  \_id: 5b9036342b95672asd || '',  updatedAt: new Date(),  user: 313356546 // id  **cartProducts:** Array [ {  product: Object,  quantity: 2  **} ],**  **Response:** {  \_id: 5b9036342b95672asd,  updatedAt: date,  user: 313356546 // id  **cartProducts:** Array [ {  product: Fully Populated Object,  quantity: 2  **} ],** |
| Make New Order | POST  /order | **Request: Object** {  city: Tel Aviv,  address: Florentine 2 Apartment 1,  shippingDate: Date,  last4Digits: 4324,  total: 9000.0,  updatedAt: new Date(),  user: 313356546 // id  **cartProducts:** Array [ {  product: Object,  quantity: 2  **} ],**  **Response: Object Fully Populated Order** {} |
| Get All Orders Shipping Dates – only next to today. | GET  /order/shipping-dates | **Request:** {},  **Response: Array [**  **Array [yyyy/mm/dd] format**  **]** |

Endpoint: *http://localhost/users*

|  |  |  |
| --- | --- | --- |
| **Description** | **Method + Path** | **Request / Response** |
| Create New User | POST  /register | **Request: Object** {  \_id: 453687898,  address: Aro'e 109  city: Ramat Gan,  first: Moshe,  last: Azulay,  username: 'email address'  password: \*\*\*,  role: 'default is guest'  },  **Response:** Object {  **Logging in and, sending back full populated User Object && applying session cookie on user browser.**  **}** |
| Login To User  **\*in case one of the properties is not valid, returns an appropriate message** | POST  /login | **Request: Object** {  username: 'email address',  password: \*\*\*  },  **Response:** Object {  **Logging in and, sending back full populated User Object && applying session cookie on user browser.**  **}** |
| Logout From User  **\*removing session cookie** | GET  /logout | **Request:** {},  **Response:** Object {  message: 'Authentication closed'  } |
| Logout From User | GET  /checksession | **Request:** {},  **Response:**  **if (has\_session)**  Object {  **Logging in and, sending back full**  **populated User Object &&**  **applying session cookie on user**  **browser.**  **}**  **else**  **{ noUser: `s` }** |
| Check If user ID is already taken | GET  /find?id=454656342 | **Request:** {},  **Response:**  **if (exists)**  Object {  hasError: `ID is already taken.`  }  **else**  **null** |
| Check If user Email is already taken | GET  /find?email=  email@example.com | **Request:** {},  **Response:**  **if (exists)**  Object {  hasError: `Email is already taken.`  }  **else**  **null** |

Endpoint: *http://localhost/home*

|  |  |  |
| --- | --- | --- |
| **Description** | **Method + Path** | **Request / Response** |
| Gets homepage random product image path, orders submitted and products count. | GET  /init | **Request:** {},  **Response:** Object {  **chosenImage: src.path,**  **orders:22,**  **products:20**  **}** |