# Resource overview

/orders

/orders/{orderID}

/orders/{orderID}/checkout

/orders/{orderID}/items/{orderItemsID}

/restaurants

/restaurants/{restID}

/restaurants/{restID}/menus

/restaurants/{restID}/menus/{menuID}

/restaurants/{restID}/menus/{menuID}/items

/restaurants/{restID}/menus/{menuID}/items/{itemID}

/users/registration

/restaurants

|  |  |
| --- | --- |
| **Action** | **Description** |
| GET /restaurants | Get a list of restaurants |
| POST /restaurants | Add a new restaurant |
| GET /restaurants/{restID} | Get a single restaurants with {restID} |
| PUT /restaurants/{restID} | Update a single restaurant with {restID} |
| DELETE /restaurants/{restID} | Delete a single restaurant with {restID} |
| GET /restaurants/{restID}/menus | Get a list of menus from the restaurant with {restID} |
| POST /restaurants/{restID}/menus | Add a new menu to the restaurant with {restID} |
| GET /restaurants/{restID}/menus/{menuID} | Get a single menu with {menuID} |
| PUT /restaurants/{restID}/menus/{menuID} | Update a single menu with {menuID} |
| DELETE /restaurants/{restID}/menus/{menuID} | Delete a single menu with {menuID} |
| GET /restaurants/{restID}/menus/{menuID}/items | Get a list of items from the menu with {menuID} |
| POST /restaurants/{restID}/menus/{menuID}/items | Add a new item to the menu with {menuID} |
| GET /restaurants/{restID}/menus/{menuID}/items/{itemID} | Get a single item with {itemID} |
| PUT /restaurants/{restID}/menus/{menuID}/items/{itemID} | Update a single item with {itemID} |
| DELETE /restaurants/{restID}/menus/{menuID}/items/{itemID} | Delete a single item with {itemID} |

/orders

|  |  |
| --- | --- |
| **Action** | **Description** |
| GET /orders | Get a list of orders |
| PUT /orders | Add a new order if no non-placed order exists for the user (currently hard-coded)   * Respond with orderID |
| GET /orders/{orderID} | Get an order with {orderID} |
| PUT /orders/{orderID} | Update an order with {orderID} |
| DELETE /orders/{orderID} | Delete an order with {orderID} |
| GET /orders/{orderID}/checkout | Get a list of items from the order with {orderID} |
| PUT /orders/{orderID}/checkout | Place the order with {orderID} |

*Data Model:*

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Description** | **Type/Format** | **Required** |
| userId | User to which the order belongs | Integer | Y |
| lastUpdated | Last updated date | Date | Y |
| Placed | Whether the order is placed or not | Integer | Y |

*JSON Form:*

{

orderId:

userId:

lastUpdated:

placed:

}

*PUT Response:*

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Description** | **Type/Format** | **Required** |
| orderID | Generated by the API during PUT. | Integer |  |

/orders/{orderID}/items/{itemID}

|  |  |
| --- | --- |
| **Action** | **Description** |
| GET /orders/{orderID}/items/{itemID} | Get orderItemID, quantity |
| PUT /orders/{orderID}/items/{itemID} | Create or update an orderItem   * Response body returned on **200 OK**. See below. |
| DELETE /orders/{orderID}/items/{itemID} | Delete an orderItem   * Response body returned on **200 OK**. See below. |

*Data Model:*

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Description** | **Type/Format** | **Required** |
| quantity | Quantity of the item | Integer | Y |

*JSON Form:*

{

quantity:

}

*PUT Response:*

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Description** | **Type/Format** | **Required** |
| OrderItemID | Generated by the API during PUT. | Integer |  |

*Error Messages:*

|  |  |
| --- | --- |
| **Code** | **Message** |
| 9000 | Order with OrderID {id} Does Not Exist |
| 9001 | Item with ItemID {id} Does Not Exist |
| 9002 | Failed to add order item to shopping cart |
| 9003 | Expected integer “quantity” of items in order as JSON input |
| 9004 | OrderItemID for OrderID {id} and ItemID Does Not Exist |
| 9005 | Failed to delete order item from shopping cart |

/users/registration

|  |  |
| --- | --- |
| **Action** | **Description** |
| GET /users/registration | Get user registration page |
| POST /users/registration | Create a new user   * Response body returned on **200 OK**. See below. |

*Data Model:*

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Description** | **Type/Format** | **Required** |
| userID | System-defined unique Customer identifier | Integer | Y |
| name | Full name | String, up to 45 characters | Y |
| phone | Example: “1234567890” | String, up to 20 numeric characters | Y |
| password | Hash using SHA-512 is stored, never full password | String, up to 45 characters | Y |
| email | Email, which is also username for login | String, e-mail format, up to 45 characters | Y |

*JSON Form:*

{

name

phone

email

password

}

*POST Response:*

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Description** | **Type/Format** | **Required** |
| userID | Generated by the API during POST. | String |  |

*Error Messages:*

|  |  |
| --- | --- |
| **Code** | **Text** |
| 9000 | User email already exists |
| 9002 | Failed to add user |
| 9003 | Name/email/password/phone was not received |