# Super Slice Online Ordering System API Docs

## User Services

These are all the services available on the customer page. Employees can also use these services, so this was left as user services.

### Log-in

Queries a user based on user and password. These are passed as parameters.

If there is a match for username & password, it will return the user object.

Otherwise, this will return null and should be handled in the front end.

|  |
| --- |
| **Request method**: GET  **URI**: /query/user?user=*jojo*&password=*password* |
| **Successful Login Return**  {  "userId": 36,  "username": "jojo",  "password": "cpassword",  "userRole": {  "roleId": 2,  "roleTitle": "Customer"  },  "tickets": []  } |
| **Unsuccessful Login Return**  null |

### Register

Submits a username & password to attempt a registration. These should be passed in the body.

If registration is successful, will return the newly created user object.

Otherwise, will return a new user with id of 0 and null values for all fields.

|  |
| --- |
| **Request method**: POST  **URI**: /register |
| **BODY FORMAT JSON**  {"username":"jojo","password":"cpassword"} |
| **Successful Register Return**  {  "userId": 37,  "username": "jojo",  "password": "cpassword",  "userRole": {  "roleId": 2,  "roleTitle": "Customer"  },  "tickets": []  } |
| **Unsuccessful Register Return**  {  "userId": 0,  "username": null,  "password": null,  "userRole": null,  "tickets": []  } |

## 

### Make Order

Submits a special OrderForm object to create a ticket & all pizzas within.

OrderForm format was created to simplify the front-end work for placing an order.

OrderForm object format shown below and should be placed in the body as JSON.

See “Index” for a list of possible “type”, size, and “toppingNames.”

(Note, toppingNames should only populate for CustomPizza type, else leave it empty)

|  |
| --- |
| **Request Method**: POST  **URI**: /makeOrder |
| **BODY FORMAT JSON**  {  "username": "jojo",  "note": "Punch door down on arrival",  "pizzaForms": [  { "type": "VeggiePizza", "size": "Large", "cost":18.00, "toppingNames": []},  { "type": "CustomPizza", "size": "Small", "cost":10.00 ,"toppingNames":["Mushrooms","Onions","Pepperoni"]}  ]  } |
| **Ticket Return**  {  "ticketId": 35,  "placementTime": "15:18:48",  "status": "Pending",  "note": "Punch door down on arrival",  "pizzas": [],  "user": {  "userId": 36,  "username": "jojo",  "password": "cpassword",  "userRole": {  "roleId": 2,  "roleTitle": "Customer"  }  }  } |

## Employee Services

These are used in the employee page exclusively. Employee login is handled by user login.

### Get Ticket By ID

Pass in a ticket ID as a path variable. Returns ticket with filled out pizza objects.

|  |
| --- |
| **Request Method:** GET  **URI**:/ticket/*35* |
| **Successful Ticket Return**  {  "ticketId": 36,  "placementTime": "15:38:31",  "status": "Pending",  "note": "Punch door down on arrival please.",  "pizzas": [  {  "pizzaId": 87,  "toppings": [  {  "toppingId": 30,  "toppingName": "Small",  "cost": 8.0  },  {  "toppingId": 11,  "toppingName": "Onions",  "cost": 1.0  },  {  "toppingId": 10,  "toppingName": "Mushrooms",  "cost": 1.0  }  ]  }  ],  "user": {  "userId": 36,  "username": "jojo",  "password": "cpassword",  "userRole": {  "roleId": 2,  "roleTitle": "Customer"  }  }  } |
| **Unsuccessful Ticket Return**  404 |

### Get Ticket By Status

Pass in one of a number of status options as parameter: Pending, Incomplete, Complete.

Returns a list of all tickets following status request, along with ALL FULL PIZZA objects.

|  |
| --- |
| **Request Method:** GET  **URI:** /ticket?status=*Incomplete* |
| **Successful Ticket Returns (Pizzas truncated)**  [  {  "ticketId": 11,  "placementTime": "19:49:50",  "status": "OutForDelivery",  "note": "I would like you to ring the middle bell.",  "pizzas": [..],  "user": {  "userId": 15,  "username": "schaud24",  "password": "cpassword",  "userRole": {  "roleId": 2,  "roleTitle": "Customer"  }  }  },  {  "ticketId": 20,  "placementTime": "21:48:24",  "status": "OutForDelivery",  "note": "I would like you to ring the middle bell.",  "pizzas": [..]  "user": {  "userId": 9,  "username": "naibing03",  "password": "cpassword",  "userRole": {  "roleId": 2,  "roleTitle": "Customer"  }  }  }  ] |
| **Sample embedded Pizza object (Toppings truncated)**  "pizzas": [  {  "pizzaId": 85,  "toppings": [..]  },  {  "pizzaId": 86,  "toppings": [..]  }  ] |

### Update Ticket

Pass in the ticket ID and the new status as parameters to update the ticket status.

The return will be the ticket item with pizza objects filled out.

|  |
| --- |
| **Request Method**: PUT  **URI:** /update?id=*36*&status=*Cooking* |
| **Successful Ticket Return**  {  "ticketId": 36,  "placementTime": "15:38:31",  "status": "Cooking",  "note": "Punch door down on arrival please.",  "pizzas": [  {  "pizzaId": 87,  "toppings": [  {  "toppingId": 10,  "toppingName": "Mushrooms",  "cost": 1.0  },  {  "toppingId": 11,  "toppingName": "Onions",  "cost": 1.0  },  {  "toppingId": 30,  "toppingName": "Small",  "cost": 8.0  }  ]  }  ],  "user": {  "userId": 36,  "username": "jojo",  "password": "cpassword",  "userRole": {  "roleId": 2,  "roleTitle": "Customer"  }  }  } |
| **Unsuccessful Ticket Return**  505 |

## Pizza Services

This is a service for the web page to access, getting general information that isn’t necessarily tied to a user or employee.

### Get Meat/Veggie Toppings & Sizes

Returns a list format of all toppings, which includes id, name, cost, and a list of all pizza IDs with said topping. Sizes get stored as a topping at the moment and are handled the same way.

|  |
| --- |
| **Request Method:** GET  **URI:** /getMeatToppings OR /getVeggieToppings OR /getSizes |
| **Successful Toppings Return (Only first two on list, meat)**  [  {  "toppingId": 22,  "toppingName": "Sausage",  "cost": 2.0,  "pizzas": [  {  "pizzaId": 82  },  {  "pizzaId": 14  },  {  "pizzaId": 48  }  ]  },  {  "toppingId": 23,  "toppingName": "Bacon",  "cost": 2.5,  "pizzas": [  {  "pizzaId": 82  },  {  "pizzaId": 14  },  {  "pizzaId": 18  }  ]  }  ] |

### Get Pizza By Type

Gets a blank pizza template for a pizza of a certain type. Used to get a list of toppings for each pizza type. As such, pass pizza names as a parameter of “toppings.”

See index for a list of all possible pizza types to pass.

|  |
| --- |
| **Request Method:** GET  **URI:** /pizza?toppings=*WhitePizza* |
| **Successful Toppings Return**  {  "pizzaId": 0,  "ticket": {  "ticketId": 0,  "placementTime": null,  "status": null,  "note": null,  "pizzas": [],  "user": null  },  "toppings": [  null,  {  "toppingId": 19,  "toppingName": "Roasted Cauliflower",  "cost": 1.0  },  {  "toppingId": 37,  "toppingName": "White Sauce",  "cost": 1.0  },  {  "toppingId": 35,  "toppingName": "Parmesan Cheese",  "cost": 1.0  },  {  "toppingId": 36,  "toppingName": "Ricotta Cheese",  "cost": 1.0  }  ]  } |

## Index

### Sizes

|  |  |  |
| --- | --- | --- |
| Small | Medium | Large |

### Pizza Types

|  |  |  |  |
| --- | --- | --- | --- |
| PlainPizza | WhitePizza | VeggiePizza | MeatLoversPizza |
| HawaiianPizza | MediterraneanPizza | FourCheesPizza | AlfredoPizza |
| ItalianPizza | SupremePizza | PepperoniPizza | MushroomPizza |
| ChickenPizza | AnchoviePizza | SpinachPizza | BaconPizza |
| CauliflowerPizza | SausagePizza | EggplantPizza | CustomPizza |

### Toppings

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Mushrooms | Onions | Green Peppers | Red Peppers | Black Olives |
| Jalapeños | Tomatoes | Broccoli | Roasted Eggplant | Roasted Cauliflower |
| Extra Cheese | Pineapples | Sausage | Bacon | Chicken |
| Meatballs | Proscuitto | Anchovies | Ham | Pepperoni |
| Spinach | Feta Cheese | Parmesan Cheese | Ricotta Cheese | White Sauce |