Ela Greenberg

Homework

Hamburger API docs

Part 1

Example request:

Accept: application/json

Content-Type: application/json

POST/orders

curl -H "Content-Type: application/json" -X POST -d'{

   "mealType":"lunch",

   "mealCat":{

   "main":"burgerMeal",

   "burger":{

      "pattyType":"veggie",

      "pattyQty":2,

      "pattyWeightG":300,

      "pattyCook":"medium",

      "bunType":"regular",

      "condiment1":"ketchupandmustard",

      "condiment2":"guacamole",

      "topping1":"lettuceandtomatoes",

      "topping2":"none",

      "topping3":"onions",

      "topping4":"jalapenos"

   },

   "sides":{

      "side1":{

         "type":"potatochips",

         "size":"large"

      },

      "side2":{

         "type":"coleslaw",

         "size":"small"

      }

   },

   "drink":{

      "type":"7-Up",

      "size":"small",

      "ice":"yes"

   }

Example response

200 OK

Example request:

Accept: application/json

Content-Type: application/json

POST/orders

curl -H "Content-Type: application/json" -X POST -d'{

   "mealType":"lunch",

   "mealCat":{

   "main":"burgerMeal",

   "burger":{

      "pattyType":"veggie",

      "pattyQty":3,

      "pattyWeightG":300,

      "pattyCook":"medium",

      "bunType":"regular",

      "condiment1":"ketchupandmustard",

      "condiment2":"guacamole",

      "topping1":"lettuceandtomatoes",

      "topping2":"none",

      "topping3":"onions",

      "topping4":"jalapenos"

   },

   "sides":{

      "side1":{

         "type":"potatochips",

         "size":"large"

      },

      "side2":{

         "type":"coleslaw",

         "size":"small"

      }

   },

   "drink":{

      "type":"7-Up",

      "size":"small",

      "ice":"yes"

Example response

400 Bad request.

Part 2

|  |  |
| --- | --- |
| Method | syntax |
| GET | GET/products/Name |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Attributes | Data Types | Default value | Mandatory/Optional | Description |
| GET/TableNo | TableNo | int | 1 | Mandatory | This specifies that it is a sit-in order. Table values can be from 1 to 10. |
| GET/table No | TableNo | int | 99 | Mandatory | This specifies that it is a takeout order. |
| Getbill | OrderNum | string |  | Mandatory | This specifies the order number. It’s a sequential number that regenerates everyday and begins with 1. |
| Getbill | Timestamp | string |  | Mandatory | This is a unique identifier generated by the CPU and reflects when the order is received. |
| Getbill | Item1 | int | 1 | Mandatory | This specifies the number of items ordered. The default is 1. |
| Getbill | ItemOrdered | string | burgermeal | Mandatory | This specifies the item ordered. |
| Getbill | Cost | int | 10.99 | Mandatory | This specifies the cost. |
| Getbill | ItemOrdered | string | Salad | Mandatory | This specifies the item ordered. |
| Getbill | Cost | int | 9.50 |  | This specifies the cost of the salad. |

Create an order

|  |  |
| --- | --- |
| Method | syntax |
| POST | Post/meal/lunch/burger/etc. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Attributes | Data Types | Default value | Mandatory/Optional | Description |
| Meal Type | lunch | string | lunch | Mandatory | This Specifies type of meal. This can be ordered between 12:00 and 2:00 |
| Main | mealCat | string | burgerMeal | Mandatory | Specifies category of meal. The choice is burger meal, chicken meal, or vegetarian special. |
| Burger | pattyType | string | beef | Mandatory | Specifies the type of burger. Can be beef or veggie. |
| Burger | pattyQty | int | 1 | Mandatory | Number of patties per order.  Acceptable values are 1 to 2. |
| Burger | pattyWeight | int | 300 | Mandatory | Specifies weight of burger. Choice is 300 grams or 400 grams. |
| Burger | pattyCook | string | MR | Mandatory | Specifies how long to cook the burger. Can be medium-rare, medium, or well done. |
| Burger | bunType | string | wholeWheat | Mandatory | The type of bun. Can be regular or whole wheat. |
| Burger | condiment1 | string | ketchup | Optional | Specifies condiment. Choices are ketchup or mustard or ketchup and mustard or none. |
| Burger | condiment2 | string | secretsauce | Optional | Specifies condiment. Choices are special sauce, salsa, or guacamole or none. |
| Burger | topping1 | string | lettuce | Optional | Specifies topping. Choices are lettuce or tomato or lettuce and tomato or none. |
| Burger | topping2 | string | pickles | Optional | Specifies topping. Choice is pickles or none. |
| Burger | topping 3 | string | none | Optional | Specifies topping. Choice is onions or none. |
| Burger | topping 4 | string | none | Optional | Specifies topping. Choice is jalapenos or none. |
| Sides | Side1 | string | frenchfries | Mandatory | Specifies side 1. Choice is Frenchfries or potato chips or none. |
| Sides | Size | string | large | Mandatory | Specifies size of side. Choice is large or none. |
| Sides | Side2 | string | None | Mandatory | Specifies side 2. Choice is salad or coleslaw or none. |
| Sides | Size | string | large | Mandatory | Specifies size of side. Choice is small or none. |
| Drink | Type | string | Coke | Mandatory | Specifies type of drink. Choice is Coke or 7-Up or Fanta or none. |
| Drink | Size | string | Large | Mandatory | Specifies size of drink. Choice is small or large or none. |
| Drink | Ice | string | Yes | Optional | Specifies ice in drink. |

3. Possible response codes

|  |  |
| --- | --- |
| Status | Code Description |
| 200 - OK | Your meal was successfully ordered. |
| 211 – Not available | A menu item is currently unavailable. |
| 400 - Bad Request | Your order was unacceptable. Missing paraments or invalid parameters. Example: a request for 3 hamburger patties, when the limit is 2. |
| 402 - Request Failed | Your order was not received by the kitchen. Try again. |
| 429 - Too Many Requests | The kitchen cannot handle all these requests. |

4. Overview of the API

Readme

Hamburger API is for ordering a hamburger at a restaurant.

Types of orders:

Orders made for takeout.

Purpose

* The cook needs to be able to view all the orders placed in the order of arrival.
* The customer needs to be able to place an order
* The customer needs to be able to know the order status