

Note: Setup Database Connection:

- Use mysql 5.5 database, Ubuntu Desktop 14.04
- DB name: restaurant
- User/Password: ampos/ampos.

I. Menu Item Rest APIs

1. Get all menu items with pagination support.

- + URL: `http://<host>:<port>/menu?page=0&size=5` GET
- + Content-type: application/json
- + Param: page is the pageID and size is the size of the page
- + Output:

```
{
  "content": [
    {
      "id": 5,
      "title": "Item 5",
      "description": "Item 5",
      "imageUrl": "1564996774468uploads",
      "price": 60,
      "details": "Hot",
      "active": true
    },
    {
      "id": 6,
      "title": "Item 6",
      "description": "B dry",
      "imageUrl": "1564996797369uploads",
      "price": 60,
      "details": "Hot",
      "active": true
    }
  ],
  "last": true,
  "totalElements": 2,
  "totalPages": 1,
  "size": 5,
  "number": 0,
  "sort": null,
  "first": true,
  "numberOfElements": 2
}
```

2. Get a menu item by ID

- + URL: http://<host>:<port>/menu/<id> GET
- + Content-type: application/json
- + Output:

```
{
  "id": 5,
  "title": "Item 5",
  "description": "Item 5",
  "imageUrl": "1564996774468uploads",
  "price": 60,
  "details": "Thai",
  "active": true
}
```

3. Create a menu item

- + URL: http://<host>:<port>/menu POST
- + Content-type: multipart/form-data
 - Param **'file'**: path of image file
 - Param **'item'**:

```
{
  "title": "Pizza",
  "description": "Pizza",
  "price": 60,
  "details": "Italian"
}
```

4. Edit a menu item

- + URL: http://<host>:<port>/menu PUT
- + Content-type: multipart/form-data
- + Input params:
 - Param **'file'**: path of image file
 - Param **'item'**:

```
{
  "id": 5,
  "title": "Beer",
  "description": "alcohol",
  "imageUrl": "1564996774468uploads",
  "price": 60,
  "details": "can"
}
```

Note: if an item is not existed, it will return an error message

5. Delete a menu item

URL: http://<host>:<port>/menu/{id} DELETE

Note:

- To handle the deleted menu item but it has existed in the bill order, the operation marked as deactivated so that the bill order won't miss menu info
- In case, the menu item has no reference to the bill order, the item will be deleted completely
- if deleting an item is not existed, it will return an error message

6. Search menu items by keyword

URL: `http://<host>:<port>/menu/search?keyword=<keyword>` GET

Param: keyword (search by keyword and return the list that matches title, description, additional details)

II. Bill Order Item Rest APIs

1. Get all order items

+ URL: `http://<host>:<port>/billOrder` GET

+ Content-type: application/json

+ Output:

```
[
  {
    "id": 2,
    "billId": 2,
    "title": "A 1 6",
    "quantity": 6,
    "orderedTime": "2019-08-05 16:44:44.0",
    "prices": 360
  }
]
```

2. Get order items by bill id.

+ URL: `http://<host>:<port>/billOrder/{id}` GET

+ Content-type: application/json

+ Output:

```
[
  {
    "id": 2,
    "billId": 2,
    "title": "Kimchi",
    "quantity": 6,
    "orderedTime": "2019/08/05 04:44:44 PM",
    "prices": 360
  }
]
```

3. Create a bill order item

- + URL: `http://<host>:<port>/billOrder` POST
- + Content-type: `application/json`
- + Input:

```
{
  "menus" : [{
    "title": "Kimchi",
    "quantity": 6
  }]
}
```

Note: if a title is not existed, it will return an error message

4. Update operation can be “add” or “remove” menu items and quantities depends on the “title” attribute

- + URL: `http://<host>:<port>/billOrder` PUT
- + Content-type: `application/json`
- + Input body for editing menu item(s) for the update operation:

```
[
  {
    "id": 2,
    "billId": 2,
    "title": "Edit title",
    "quantity": 10
  }
]
```
- + Input for removing menu item(s) for the delete operation:

```
[
  { //For remove items, if title is empty or null, mark its as remove
    "id": 3,
    "billId": 2
  }
]
```

5. Check bill order

- + URL: `http://<host>:<port>/billOrder/check<id>` GET
Id: bill number id
- + Content-type: `application/json`
- + Output:

```
{
  "orderItems": [
    {
      "title": "Kimchi",
      "quantity": 10,
      "price": 60,
      "totalPrices": 600,
    }
  ]
}
```

```
        "id": 2
      }
    ],
    "totalPirces": 600
  }
}
```

6. Total price of items in the bill orders group by menu item

- + URL: `http://<host>:<port>/billOrder/totalBills` GET
- + Content-type: `application/json`
- + Output:

```
{
  "orderItems": [
    {
      "title": "Kimchi",
      "quantity": 10,
      "price": 60,
      "totalPrices": 600,
      "id": 2
    }
  ],
  "totalPirces": 600
}
```