CMPE-281 Project API Design

Before Designing

http://ec2-52-11-87-42.us-west-2.compute.amazonaws.com/

Now the database web server are all running at this EC2, please feel free to use this URL to test first.

Recommend RESTful API Design Tutorial

http://www.restapitutorial.com/lessons/restquicktips.html
http://en.wikipedia.org/wiki/Representational_state_transfer#Example

https://devdactic.com/restful-api-user-authentication-2/

In the scenarios that needs user authentication, the backend will provide a token after user login, and in the future the front end need to send the token in the http headers authorization at this user's every request. The token should be included in the Authorization in the header.

Example

http://docs.timdorr.apiary.io

CODE	Description
200	OK - Request completed.
201	CREATED - New resource created.
204	NO CONTENT - Success but no content to return. Only the status code returns.
400	BAD REQUEST - The request could not be parsed.
401	UNAUTHORIZED - User is not logged in, cannot access resource.

403	FORBIDDEN - User is not authorized to perform the operation or access the resource.
404	NOT FOUND - Resource doesn't exist.
407	CONFLICT- Duplicate entries, such as creating a user that already exists.
500	INTERNAL SERVER ERROR - Server error.
Others	Can be found on http://www.restapitutorial.com/httpstatuscodes.html

Comment model (Yuan)

changed :

POST	/comment/addcomments/:order_id/:dish/:score/:description
Add a new comment	

Request

POST /comment/addcomments/BkR4rmEb/Rice/4/"Good"

Parameter	Description
Token	Token needed
order_id	The order id
dish	The dish that the user writes comment for
score	The score use give to the dish
description	The detailed comment user write

Response

Status	200
Content-Type	application/json

```
{
  "sucess": true,
  "msg": "Add comment successful",
  "data": {
    "__v": 0,
    "createdAt": "2016-05-02T00:56:04.153Z",
    "updatedAt": "2016-05-02T00:56:04.153Z",
    "user_name": "testuser",
    "order_id": "BkR4rmEb",
    "dish_name": "Rice",
    "score": 4,
    "comment_description": "\"Good\"",
    "_id": "5726a5a451a6e2e20aef3d48"
  }
}
```

Added: Find comment by order_id

GET	/comment/findbyorder/:order_id
Add a new comment	

Request

GET /comment/findbyorder/BkR4rmEb

Parameter	Description
id	order_id

Response

Status	200
Content-Type	application/json

```
{
  "sucess": true,
  "msg": "Add comment successful",
  "data": {
    "__v": 0,
    "createdAt": "2016-05-02T00:56:04.153Z",
    "updatedAt": "2016-05-02T00:56:04.153Z",
    "user_name": "testuser",
    "order_id": "BkR4rmEb",
    "dish_name": "Rice",
    "score": 4,
    "comment_description": "\"Good\"",
    "_id": "5726a5a451a6e2e20aef3d48"
  }
}
```

Add : Find order by order id:

GET	/order/getorders/:order_id

Add a new comment

Request

GET /order/getorders/BkR4rmEb

Parameter	Description
Token	Token needed for authorization
id	order_id

Response

Status	200
Content-Type	application/json

```
{
  "sucess": true,
  "data": [
    {
      "_id": "BkR4rmEb",
      "user": "testuser",
      "total_price": 22.5,
      "__v": 0,
      "card": [
        {
          "card_number": 1212334,
          "card_type": "credit"
        }
      ],
      "dishes": [
        {
          "name": "Rice",
          "description": "This is a traditional noodle",
          "price": 12,
          "amount": 1,
          "total": 12
        },
        {
          "name": "Ramen",
          "description": "This is a traditional noodle",
          "price": 10,
          "amount": 1,
          "total": 10.5
        }
      ],
      "delivery_address": [
        {
          "receiver": "Lucy",
          "phone": 23423425,
          "address": "44444 Road",
          "Zip_Code": 67890,
          "State": "CA"
        }
      ],
      "confirm_time": null,
      "order_time": "2016-05-02T00:54:13.541Z"
    }
  ]
}
```

GET	/comment//findbyuser/:user

Find all comments posted by a user

Request

GET /comment//findbyuser/Lucy

Parameter	Description
user	The current user who what to find all the comment he has written for

Response

Status	200
Content-Type	application/json

```
[
  {
    "_id": "5713256c603074ba091c4a05",
    "createdAt": "2016-04-17T05:55:56.243Z",
    "updatedAt": "2016-04-17T05:55:56.243Z",
    "user_name": "Lucy",
    "dish_name": {
      "_id": "Ramen",
      "dish_price": 10,
      "dish_description": "Original ramen from Japan,
made of pork",
      "_catalog": "Noodle",
      "dish_picture": "null"
    },
    "score": 4,
    "comment_description": "I like it!",
    "__v": 0
  },
  {
    "_id": "5713244b603074ba091c4a02",
    "createdAt": "2016-04-17T05:51:07.630Z",
    "updatedAt": "2016-04-17T05:51:07.630Z",
    "user_name": "Lucy",
    "dish_name": {
      "_id": "Ramen",
      "dish_price": 10,
      "dish_description": "Original ramen from Japan,
made of pork",
      "_catalog": "Noodle",
      "dish_picture": "null"
    },
    "score": 4,
    "comment_description": ""I like it!"",
    "__v": 0
  },
   {
    "_id": "57144a9726a01a9c0d3a6d44",
    "createdAt": "2016-04-18T02:46:47.376Z",
    "updatedAt": "2016-04-18T02:46:47.376Z",
    "user_name": "Lucy",
    "dish_name": {
      "_id": "Ramen",
      "dish_price": 10,
      "dish_description": "Original ramen from Japan,
made of pork",
      "dish_soldout": "No",
      "_catalog": "Noodle",
      "dish_picture": "null"
    },
    "score": 4,
    "comment_description": "I like it!",
    "__v": 0
  }
]
```

GET	/comment/findbydish/:dish

Find all comments for one dish

Request

GET /comment/findbydish/Ramen

Parameter	Description
dish	The dish name user want to find comment for

Response

Status	200
Content-Type	application/json

```
[
  {
    "_id": "5713256c603074ba091c4a05",
    "createdAt": "2016-04-17T05:55:56.243Z",
    "updatedAt": "2016-04-17T05:55:56.243Z",
    "user_name": "Lucy",
    "dish_name": "Ramen",
    "score": 4,
    "comment_description": "I like it!",
    "__v": 0
  },
  {
    "_id": "57132499603074ba091c4a04",
    "createdAt": "2016-04-17T05:52:25.797Z",
    "updatedAt": "2016-04-17T05:52:25.797Z",
    "user_name": "Dora Jia",
    "dish_name": "Ramen",
    "score": 6,
    "comment_description": "I like it!",
    "__v": 0
  },
  {
    "_id": "5713248b603074ba091c4a03",
    "createdAt": "2016-04-17T05:52:11.534Z",
    "updatedAt": "2016-04-17T05:52:11.534Z",
    "user_name": "Dora Jia",
    "dish_name": "Ramen",
    "score": 4,
    "comment_description": "'I like it!'",
    "__v": 0
  },
  {
    "_id": "5713244b603074ba091c4a02",
    "createdAt": "2016-04-17T05:51:07.630Z",
    "updatedAt": "2016-04-17T05:51:07.630Z",
    "user_name": "Lucy",
    "dish_name": "Ramen",
    "score": 4,
    "comment_description": "'I like it!'",
    "__v": 0
  },
  {
    "_id": "57144a9726a01a9c0d3a6d44",
    "createdAt": "2016-04-18T02:46:47.376Z",
    "updatedAt": "2016-04-18T02:46:47.376Z",
    "user_name": "Lucy",
    "dish_name": "Ramen",
    "score": 4,
    "comment_description": "I like it!",
    "__v": 0
  }
]
```

Catalog and dishes **(Yuan) (This part is for the main page, only for display)**

GET	/catalog/getallcatalog
Find all catalog information	

Request

GET /catalog/getallcatalog

Parameter	Description
no	no

Response

Status	200
Content-Type	application/json

```
[
  {
    "_id": "Noodle",
    "catalog_description": "Different noodle has different flavors",
    "_dish": [
      "Ramen"
    ]
  }
]
```

GET	/dish/findbycatalog/:name
Find all dishes under one catalog	

Request

Response

Status	200
Content-Type	application/json

```
[
 {
   "_id": "Ramen",
   "dish_price": 10,
   "dish_description": "Original ramen from Japan,
made of pork",
   "_catalog": "Noodle",
   "dish_picture": "null"
 }
]
```

GET	/dish/findbyname/:name

Find the dish info by using dish name

Request

GET /dish/findbyname/Ramen

Parameter	Description
name	The dish name

Response

Status	200
Content-Type	application/json

```
[
  {
    "_id": "Ramen",
    "dish_price": 10,
    "dish_description": "Original ramen from Japan, made of pork",
    "_catalog": "Noodle",
    "dish_picture": "null"
  }
]
```

User model (Yuan)

POST	/user/signup/:name/:password/:email/:phone

Sign up new account.

Request

POST /user/signup/test4/test/Test@Dora/12345

Parameter	Description
name	The account name of user.
password	The password of user.
email	The email address of user
phone	The phone number of user

Response

Status	200
Content-Type	application/json

```
{
  "success": true,
  "msg": "Successful created new user."
}
```

Status	403
Content-Type	application/json

```
{
        "message" : "Not Authorized"
        "success" : false,
        "status" : 403
}
```

POST	/user/login/:name/:password

User login to the system

Request

POST /user/login/testuser/testpassword

Parameter	Description
name	The account name of user.
password	The password of user.

Response

Status	200
Content-Type	application/json

```
{
  "success": true,
  "token": "JWT
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJfaWQiOi
J0ZXN0dXNlciIsInBhc3N3b3JkIjoiJDJhJDEwJGkweC9
jUk1sRGpIeFgxeFgvM1puMi43QWs0ZFZ4ZDVXdkpR
Z29aUmp1ZzJwZEdlaW0ySXhtIiwiZW1haWwiOiJUZX
N0QHFxLmNvbSIsIl9fdiI6MCwiZGVsaXZlcnlfYWRkcm
VzcyI6W10sInBob25lIjpbeyJfaWQiOjEyMzQ1Njc4OTB
9LHsiX2lkIjoyMjIyMn0seyJfaWQiOjMzMzMzfV19.ZCq
DULLCe_2kj3ZJrjX1zo92qa7sAromiehryDTEwvc"
}
```

Status	403
Content-Type	application/json

```
{
        "message" : "Not Authorized"
        "success" : false,
        "status" : 403
}
```

POST	/user/updatepassword/:oldpassword/:newpassword

Update user password

Request

POST /user/updatepassword/testpassword/testpasswordupdate

Parameter	Description
name	The account name of user.
password	The password of user.

Token needed

https://devdactic.com/restful-api-user-authentication-2/

In the scenarios that needs user authentication, the backend will provide a token after user login, and in the future the front end need to send the token as http headers at this user’s every request. The token should be included in the Authorization in the header.

Response

Status	200
Content-Type	application/json

```
{
  "success": true,
  "msg": "Update successful"
}
```

Status	403
Content-Type	application/json

```
{
        "message" : "Not Authorized"
        "success" : false,
        "status" : 403
}
```

GET	/user/userprofile

Get user profile information

Request

GET /user/userprofile

Parameter	Description
Token	The web token used to authenticate user

Token needed

https://devdactic.com/restful-api-user-authentication-2/

In the scenarios that needs user authentication, the

Response

Status	200
Content-Type	application/json

```
{
  "sucess": true,
  "data": {
    "_id": "testuser",
    "email": "Test@qq.com",
    "delivery_address": [],
    "phone": [
      {
        "_id": 1234567890
      }
    ]
  }
}
```

Status	403
Content-Type	application/json

```
{
                "message" : "Not Authorized"
                "success" : false,
                "status" : 403
}
```

POST	user/addaddress

Add one delivery address

Request

POST /user/addaddress

Parameter	Description

Response

Status	200
Content-Type	application/json

```
{
  "sucess": true,
  "delivery_address": [
    {
      "isdefault": false,
      "phone": 4081234567,
      "receiver": "Judy Kass",
      "address": "Howold street",
      "state": "CA",
      "zipcode": 44444,
      "_id": "5725bb92baf9a11a04c4d5d0"
    }
  ]
}
```

Status	403
Content-Type	application/json

```
{
                "message" : "Not Authorized"
                "success" : false,
        "status" : 403
}
```

GET	/user/getaddress

Get one user’s all delivery address

Request

GET /user/getaddress

Parameter	Description
Token	The web token that returned from backend after user login, it is used to authenticate user

Token needed

https://devdactic.com/restful-api-user-authentication-2/

In the scenarios that needs user authentication, the backend will provide a token after user login, and in the future the front end need to send the token as http headers at this user’s every request.

Response

Status	200
Content-Type	application/json

```
{
  "sucess": true,
  "delivery_address": [
    {
      "isdefault": false,
      "phone": 4081234567,
      "receiver": "Judy Kass",
      "address": "Howold street",
      "state": "CA",
      "zipcode": 44444,
      "_id": "5725bb92baf9a11a04c4d5d0"
    },
    {
      "isdefault": true,
      "phone": 2342345566,
      "receiver": "Ledy",
      "address": "Helloworld street",
      "state": "CA",
      "zipcode": 34354,
      "_id": "5725bc1abaf9a11a04c4d5d1"
    }
  ]
}
```

Status	403
Content-Type	application/json

```
{
        "message" : "Not Authorized"
        "success" : false,
        "status" : 403
}
```

POST	/user/removeaddress/:id
Remove one address	

Request

POST /user/removeaddress/5725bb92baf9a11a04c4d5d0

Parameter	Description
Token	The web token that returned from backend after user login, it is used to authenticate user
id	The _id field in address

Token needed

https://devdactic.com/restful-api-user-authentication-2/

In the scenarios that needs user authentication, the backend will provide a token after user login, and in the

Response

Status	200
Content-Type	application/json

```
{
  "sucess": true,
  "delivery_address": [
    {
      "isdefault": true,
      "phone": 2342345566,
      "receiver": "Ledy",
      "address": "Helloworld street",
      "state": "CA",
      "zipcode": 34354,
      "_id": "5725bc1abaf9a11a04c4d5d1"
    }
  ]
}
```

Status	403
Content-Type	application/json

```
{
                "message" : "Not Authorized"
                "success" : false,
                "status" : 403
}
```

POST	/user/editaddress/:id

Remove one address

Request

POST
/user/removeaddress/5725bb92baf9a11a04c4d5d0

Response

Status	200
Content-Type	application/json

```
{
  "sucess": true,
  "delivery_address": [
    {
      "isdefault": false,
      "phone": 2342345566,
      "receiver": "Ledytest",
      "address": "Helloworld streettest",
      "state": "CA",
      "zipcode": 34354,
      "_id": "5725ca20cf65820b07cccca6"
    }
  ]
}
```

Status	403
Conte	application/json

nt-
Type

```
{
        "message" : "Not Authorized"
        "success" : false,
        "status" : 403
}
```

POST	/user/removephone/:phone

Remove one phone number

Request

POST /user/removephone/123456789

Parameter	Description
Token	The web token that returned from backend after user login, it is used to authenticate user

Token needed

https://devdactic.com/restful-api-user-authentication-2/

In the scenarios that needs user authentication, the backend will provide a token after user login, and in the future the front end need to send the token as http headers at this user's every request.

Response

Status	200
Content-Type	application/json

```
{
  "sucess": true,
  "phone": [
    {
      "_id": 1234567890
    },
    {
      "_id": 22222
    },
    {
      "_id": 33333
    }
  ]
}
```

Status	403
Content-Type	application/json

```
{
        "message" : "Not Authorized"
        "success" : false,
        "status" : 403
}
```

POST	/user/addphone/:phone

Add one phone number

Request

POST /user/addphone/123456789

Parameter	Description
Token	The web token that returned from backend after user login, it is used to authenticate user

Token needed

https://devdactic.com/restful-api-user-authentication-2/

In the scenarios that needs user authentication, the backend will provide a token after user login, and in the future the front end need to send the token as http headers at this user’s every request.

Response

Status 200

Content-Type	application/json

```
{
  "sucess": true,
  "phone": [
    {
      "_id": 1234567890
    },
    {
      "_id": 22222
    },
    {
      "_id": 33333
    },
    {
      "_id": 123456789
    }
  ]
}
```

Status	403
Content-Type	application/json

```
{
                "message" : "Not Authorized"
                "success" : false,
                "status" : 403
}
```

Shopping cart model (Yuan)

POST	/cart/additem/:name/:price/:number/:total_price

Add items to shopping cart

Request

POST /cart/additem/Ramen/10/1/10

Parameter	Description
Token	The web token that returned from backend after user login, it is used to authenticate user
name	The dish name that added into shopping cart

Response

Status	200
Content-Type	application/json

```
{
  "success": true,
  "msg": "Add item in shopping cart successful",
  "data": {
    "_id": "cart",
    "__v": 0,
    "total_price": 10,
    "status": "active",
    "dish": [
      {
        "dish_price": 10,
        "_id": "Ramen",
        "dish_number": 1
      }
    ]
  }
}
```

Status	403
Content-Type	application/json

```
{
                "message" : "Not Authorized"
                "success" : false,
                "status" : 403
}
```

POST	/cart/changenumber/:name/:newprice/:newnumber/:newtotal_price
Change the number of items in shopping cart	

Request

POST /cart/changenumber/Ramen/20/2/30

Parameter	Description
Token	The web token that returned from backend after user login, it is used to authenticate user
name	The dish user wants to change the number of
newnumber	The new number of the dish, it should be greater than 0, otherwise, please use delete item.
newprice	The new price of this item
newtotal_price	The new total_price of the items in shopping cart

Token needed

https://devdactic.com/restful-api-user-authentication-2/

In the scenarios that needs user authentication, the backend will provide a token after user login, and in the future the front end need to send the token as http headers at this user’s every request.

Response

Status	200
Content-Type	application/json

```
{
  "success": true,
  "msg": "change the number of item in shopping cart
successful",
  "data": {
    "_id": "cart",
    "__v": 0,
    "total_price": 30,
    "status": "active",
    "dish": [
      {
        "dish_price": 20,
        "_id": "Ramen",
        "dish_number": 2
      },
      {
        "dish_price": 10,
        "_id": "Rice",
        "dish_number": 1
      }
    ]
  }
}
```

Status	403
Content-Type	application/json

```
{
        "message" : "Not Authorized"
        "success" : false,
        "status" : 403
}
```

POST	cart/removeitem/:name/:newtotal_price

Remove items from shopping cart

Request

POST /cart/removeitem/Rice/20

Parameter	Description
Token	The web token that returned from backend after user login, it is used to authenticate user
Name	The dish user wants to remove
newtotal_price	New total price of the items in shopping cart

Response

Status	200
Content-Type	application/json

```
{
  "success": true,
  "msg": "Remove items from shopping cart
successful",
  "data": {
    "_id": "cart",
    "__v": 0,
    "total_price": 20,
    "status": "active",
    "dish": [
      {
        "dish_price": 20,
        "_id": "Ramen",
        "dish_number": 2
      }
    ]
  }
}
```

Status	403
Content-Type	application/json

```
{
                "message" : "Not Authorized"
                "success" : false,
                "status" : 403
}
```

GET	/cart/placeorder/:restaurant/:street/:city/:state/:zip/:phone

Place order

Request

GET /cart/placeorder/testRestaurant/811 North/San Jose/CA/95138/12345

Parameter	Description
Token	The web token that returned from backend after user login, it is used to authenticate user
restaurant	The restaurant name user placed order at
street	The user’s delivery address
city	The user’s delivery address
state	The user’s delivery address
zip	The user’s delivery address
phone	The user’s phone

Token needed

https://devdactic.com/restful-api-user-authentication-2/

In the scenarios that needs user authentication, the backend will provide a token after user login, and in the future the front end need to send the token as http headers at this user’s every request.

Response

Status	200
Content-Type	application/json

```
{
  "success": true,
  "msg": "place order successful",
  "data": {
    "__v": 0,
    "_id": "HJrYjkhl",
    "user": "cart",
    "restaurant_name": "testRestaurant",
    "total_price": 20,
    "dishes": [
      [
        {
          "dish_price": 20,
          "_id": "Ramen",
          "dish_number": 2
        }
      ]
    ],
    "confirm_time": null,
    "order_time": "2016-04-25T19:09:48.578Z"
  }
}
```

Status	403
Content-Type	application/json

```
{
        "message" : "Not Authorized"
        "success" : false,
        "status" : 403
}
```

New Place Order Function:

POST	/cart/placeorder
After select items in shopping cart, place order	

Request

POST /cart/placeorder

Token needed

Please send the following format **application/json file,** since the address, card and orderItems doesn't have defined schema, so please pass anything you want, but there are several fields better not change, it is in **bold**. And it is related to the user's cart, before test, please make sure the cart has these items, otherwise, there will be error.

{
"**address**":{
  "address" : "44444 Road",
  "zip_Code": 67890,
  "state" :"CA"
},

"**phone**":12345567,

Response

Status	200
Content-Type	application/json

```
{
  "success": true,
  "msg": "place order successful",
  "data": {
    "__v": 0,
    "_id": "HJfK4cxb",
    "user": "carttest",
    "total_price": 20,
    "card": [
      {
        "card_type": "credit",
        "card_number": 1212334
      },
      {
        "card_type": "credit",
        "card_number": 123324325
      }
    ],
    "dishes": [
      {
        "total": 10,
        "amount": 1,
        "price": 10,
        "description": "This is a traditional noodle",
        "name": "Rice"
      },
      {
        "total": 10,
        "amount": 1,
        "price": 10,
        "description": "This is a traditional noodle",
        "name": "Ramen"
      }
    ],
    "delivery_address": [
      {
        "State": "CA",
        "Zip_Code": 67890,
        "address": "44444 Road"
      }
    ],
    "confirm_time": null,
    "order_time": "2016-04-29T07:59:53.780Z"
  }
}
```

Status	403
Content-Type	application/json

```
{
        "message" : "Not Authorized"
        "success" : false,
        "status" : 403
}
```

GET	/cart/getitems
Get all items from shopping cart	

Request

GET /cart/getitems

Parameter	Description
Token	The web token that returned from backend after user login, it is used to authenticate user

Token needed

https://devdactic.com/restful-api-user-authentication-2/

In the scenarios that needs user authentication, the backend will provide a token after user login, and in the future the front end need to send the token as http headers at this user’s every request.

Response

Status	200
Content-Type	application/json

```
{
  "sucess": true,
  "data": {
    "_id": "cart",
    "__v": 0,
    "total_price": 0,
    "status": "Empty",
    "dish": []
  }
}
```

Status	403
Content-Type	application/json

```
{
                "message" : "Not Authorized"
                "success" : false,
                "status" : 403
}
```

Order Model **(Yuan)**

GET	/order/getorders
Get all orders for the user	

Request

GET /order/getorders

Parameter	Description
Token	The web token that returned from backend after user login, it is used to authenticate user

Token needed

Response

Status	200
Content-Type	application/json

```
{
  "sucess": true,
  "data": [
    {
      "order_id": "Skou7k2x",
      "user": "cart",
      "restaurant_name": "testRestaurant",
      "total_price": 30,
      "__v": 0,
      "dishes": [
        [
          {
            "dish_number": 2,
            "_id": "Ramen",
            "dish_price": 20
          },
          {
            "dish_number": 1,
            "_id": "Rice",
            "dish_price": 10
          }
        ]
      ],
      "confirm_time": null,
      "order_time": "2016-04-25T18:35:30.882Z"
    },
    {
      "_id": "HkuHrJ3l",
      "user": "cart",
      "restaurant_name": "testRestaurant",
      "total_price": 30,
      "__v": 0,
      "dishes": [
        [
          {
            "dish_number": 2,
            "_id": "Ramen",
            "dish_price": 20
          },
          {
            "dish_number": 1,
            "_id": "Rice",
            "dish_price": 10
          }
        ]
      ],
      "confirm_time": null,
      "order_time": "2016-04-25T18:43:12.136Z"
    },
    {
      "_id": "HJrYjkhl",
      "user": "cart",
      "restaurant_name": "testRestaurant",
      "total_price": 20,
      "__v": 0,
      "dishes": [
        [
          {
            "dish_number": 2,
            "_id": "Ramen",
            "dish_price": 20
          }
        ]
      ],
      "confirm_time": null,
      "order_time": "2016-04-25T19:09:48.578Z"
    }
  ]
}
```

Status	403
Content-Type	application/json

```
{
        "message" : "Not Authorized"
        "success" : false,
        "status" : 403
}
```

POST	/order/confirm/:id

Confirm order

Request

POST /order/confirm/HJrYjkhl

Parameter	Description
Token	The web token that returned from backend after user login, it is used to authenticate user
id	The order id

Token needed

Response

Status	200
Content-Type	application/json

```
{
  "success": true,
  "msg": "Confirm Order succeed!",
  "data": {
    "_id": "HJrYjkhl",
    "user": "cart",
    "restaurant_name": "testRestaurant",
    "total_price": 20,
    "__v": 0,
    "dishes": [
      [
        {
          "dish_number": 2,
          "_id": "Ramen",
          "dish_price": 20
        }
      ]
    ],
    "confirm_time": "2016-04-26T23:40:52.995Z",
    "order_time": "2016-04-25T19:09:48.578Z"
  }
}
```

Status	403
Content-Type	application/json

```
{
        "message" : "Not Authorized"
        "success" : false,
        "status" : 403
}
```

Special dishes show (Xing):

GET	/special

Get all special dishes info

Request
GET /special No parameters needed .

Response

Status	200
Content-Type	application/json

```
[
  {
    "_id": 2,
    "special_dish": {
      "_id": "Ramen",
      "dish_price": 10,
      "dish_description": "Original ramen from Japan, made of pork",
      "_catalog": "Noodle",
      "dish_picture": "null"
    },
    "__v": 0
  },
  {
    "_id": 1,
    "special_dish": {
      "_id": "pork and chives dumpling",
      "dish_price": 8.5,
      "dish_description": "Most traditional dumpling with pork and chives",
      "_catalog": "Dumpling",
      "dish_picture": "null"
    }
  }
]
```

Recommendation model (Xing):

GET	/recommendation/:order

Get the recommendation list according to user's order

Request

GET /recommendation/Ramen,pork and chives dumpling

Parameter	Description
order	The user’s order list which separate the dish names with “,”

Response

Status	200
Content-Type	application/json

```
[
  {
    "_id": "egg drop soup",
    "dish_price": 2,
    "dish_description": "wispy beaten eggs in boiled chicken broth",
    "_catalog": "Soup",
    "dish_picture": "null"
  },
  {
    "_id": "cocacola",
    "dish_price": 1.5,
    "dish_description": "",
    "_catalog": "Beverage",
    "dish_picture": "null"
  }
]
```

Inventory Model (Xing):

GET	/inventory/:branchId

Get the dishes list as well as their sold_out status according to branch Id(zip code)

Request

GET /inventory/95054

Parameter	Description
brachId	The branch restaurant id, Currently, it is the zip code.

Response

Status	200
Content-Type	application/json

```
[
  {
    "dish_name": {
      "_id": "cocacola",
      "dish_price": 1.5,
      "dish_description": "",
      "_catalog": "Beverage",
      "dish_picture": "null"
    },
    "dish_soldout": false
  },
  {
    "dish_name": {
      "_id": "Ramen",
      "dish_price": 10,
      "dish_description": "Original ramen from Japan, made of pork",
      "dish_soldout": "No",
      "_catalog": "Noodle",
      "dish_picture": "null"
    },
    "dish_soldout": true
  },
  {
    "dish_name": {
      "_id": "egg drop soup",
      "dish_price": 2,
      "dish_description": "wispy beaten eggs in boiled chicken broth",
      "_catalog": "Soup",
      "dish_picture": "null"
    },
    "dish_soldout": false
  },
  {
    "dish_name": {
      "_id": "pork and chives dumpling",
      "dish_price": 8.5,
      "dish_description": "Most traditional dumpling with
pork and chives",
      "_catalog": "Dumpling",
      "dish_picture": "null"
    },
    "dish_soldout": false
  }
]
```

Paycard Model(Xing)

GET	/paycard/
GET all cards info by user_name(_id in user collection)	

Request

GET /paycard/

No parameters needed

Token needed

https://devdactic.com/restful-api-user-authentication-2/

In the scenarios that needs user authentication, the backend will provide a token after user login, and in the future the front end need to send the token as http headers at this user’s every request.

Response

Status	200
Content-Type	application/json

```
[
  {
    "_id": "572396f606484fcb75d90fa4",
    "user_name": "Dora",
    "card_type": "Debit",
    "card_holder": "YuanYuan Jia",
    "card_number": 329883942322,
    "expire_date": "09/2019",
    "bank_name": "Citi"
  },
  {
    "_id": "572396fd06484fcb75d90fa5",
    "user_name": "Dora",
    "card_type": "credit",
    "card_holder": "YuanYuan Jia",
    "card_number": 365678654492,
    "expire_date": "01/2020",
    "bank_name": "BOA"
  }
]
```

Status	403
Content-Type	application/json

```
{
        "message" : "Not Authorized"
        "success" : false,
        "status" : 403
}
```

POST	/paycard/:type/:holder/:cardNo/:expireDate/:bankName

Create a card file for the specific user

Notice:

1. If the card has no expire date, that field should use a space instead.
2. For the expire date string “MM/YYYY”, use %2F to represent “/”.

Request

POST /paycard/debit/Sun/888888666666/05%2F2011/BOA

Parameter	Description
type	The card type: debit or credit
holder	The card holder’s name
cardNo	Card Number
expire	This card’s expire date

Response

Status	200
Content-Type	application/json

```
{
  "__v": 0,
  "user_name": "Java",
  "card_type": "debit",
  "card_holder": "Sun",
  "card_number": 888888666666,
  "expire_date": "05/2011",
  "bank_name": "BOA",
  "_id": "572435f52dc30c441adaed74"
}
```

PUT	/:cardNo/:type/:holder/:newcardNo/:expireDate/:bankName

Create a card file for the specific user

Notice:

1. _id and cardNo is used to find the record in the table.
2. All card's fields need to set no matter it is changed or not.
3. If the card has no expire date, that field should use a space instead.
4. For the expire date string "MM/YYYY", use %2F to represent "/".

Request

Response

Status	200
Content-Type	application/json

```
{
  "_id": "572435f52dc30c441adaed74",
  "user_name": "Java",
  "card_type": "debit",
  "card_holder": "Sun",
  "card_number": 123456781234,
  "expire_date": "04/2019",
  "bank_name": "BOA",
  "__v": 0
}
```

DELETE	/paycard/:cardNo
DELETE the specific cards info by user_name(_id in user table) and card number	

Request
DELETE /paycard/123456781234

Parameter	Description
cardNo	Card number

Token needed

https://devdactic.com/restful-api-user-authentication-2/

In the scenarios that needs user authentication, the backend will provide a token after user login, and in the future the front end need to send the token as http headers at this user’s every request.

Response

Status	200
Content-Type	application/json

```
{
  "message": "Success. Item deleted."
}
```

User module

1. user need account name, password, phone number, email address to sign up
2. user need account name and password to login
3. Get user information(profile, delivery address, order history)
4. Edit user information(profile, delivery address)
5. Confirm delivery in order history
6. Comment dishes in order history after confirm
7. Logout

Menu module

1. Get menu list by catalog(starter, main course, noodle, rice, beverage, dessert, soup)
2. Get a certain dish info by dish id(picture, name, price, materia,comments)
3. Order Menu
4. Get Special dishes for today
5. Get comment of a dish by dish id

Ref requirements:

a. Menu Display
    i. Menu will display in catalog(starter, main course, noodle, rice, beverage, dessert, soup).
    ii. Each dish should show picture, name, price, materia,comments
    iii. The menu will display in the main page for use scan, but it has order button, once user click on the order button, the user will need to login first and select the location(zipcode), then the user will go to an order page.
    iv. Main page should have special dishes for today.
    v. If we have comments, then in the order page, every dish needs to have a comment button or total score (stars or something) and once click on the comment button, the comment page for the dishes will show up and display all the comments made by users for this dishes.

Shopping Cart Module

1. Add dish into shopping cart by dish information
2. Get items in shopping cart
3. Delete items in shopping cart by dish_name
4. Edit shopping by dish_infor
5. Get recommended dishes by dish id
6. Check out preparation(enter the address and contact information)
7. Check out (credit card, cash)

.

Ref requirements:

i. In order page, every dish has “Add to shopping cart” button, once user click on add to shopping cart, the dish will be saved in the shopping cart. After user has selected all the items, he goes to the shopping cart to check his dishes.
ii. In shopping cart page, the user will see all the dishes he selected and the total price, also the recommended dishes. User can select the recommended dishes into the order or delete the unwanted items from the shopping cart.
iii. After user has confirmed that he wants to check out, he will click on check out and user will be asked to enter the address and contact information, after user finish it, user will go to the checkout page.
iv. In the checkout page, the order details(orders, delivery address, contact number) will show up and user can select different ways to check out(credit card, cash).