CMPE-281 Project API Design

**Before Designing**

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

|  |  |
| --- | --- |
| POST | /comment/addcomments/:dish/:score/:description |
| Add a new comment | |
| |  | | --- | | Request | | POST /comment/addComments/Ramen/4/I like it!   |  |  | | --- | --- | | Parameter | Description | | Token | Token needed | | 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 | | {  "\_\_v": 0,  "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!",  "\_id": "5713256c603074ba091c4a05"  } | | | | |

|  |  |
| --- | --- |
| 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:

|  |  |
| --- | --- |
| 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 | | GET /dish/findbycatalog/:name   |  |  | | --- | --- | | Parameter | Description | | name | The catalog name | |      |  | | --- | | Response | | |  |  | | --- | --- | | Status | 200 | | Content-Type | application/json | | [  {  "\_id": "Ramen",  "dish\_price": 10,  "dish\_description": "Original ramen from Japan, made of pork",  "dish\_soldout": "No",  "\_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

|  |  |
| --- | --- |
| 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.eyJfaWQiOiJ0ZXN0dXNlciIsInBhc3N3b3JkIjoiJDJhJDEwJGkweC9jUk1sRGpIeFgxeFgvM1puMi43QWs0ZFZ4ZDVXdkpRZ29aUmp1ZzJwZEdlaW0ySXhtIiwiZW1haWwiOiJUZXN0QHFxLmNvbSIsIl9fdiI6MCwiZGVsaXZlcnlfYWRkcmVzcyI6W10sInBob25lIjpbeyJfaWQiOjEyMzQ1Njc4OTB9LHsiX2lkIjoyMjIyMn0seyJfaWQiOjMzMzMzfV19.ZCqDULLCe\_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 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": "testuser",  "email": "Test@qq.com",  "delivery\_address": [],  "phone": [  {  "\_id": 1234567890  }  ]  }  } | |  |  |  | | --- | --- | | Status | 403 | | Content-Type | application/json | | {  "message" : "Not Authorized"  "success" : false,  "status" : 403  } | | | | |

|  |  |
| --- | --- |
| POST | /addaddress/:state/:zipcode/:address |
| Add one delivery address | |
| |  | | --- | | Request | | POST /user/addaddress/CA/44444/Howold street   |  |  | | --- | --- | | 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": [  {  "address": "Howold street",  "state": "CA",  "zipcode": 44444,  "\_id": "5719a7164cf09c803238781a"  }  ]  } | |  |  |  | | --- | --- | | 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": [  {  "address": "Howold street",  "state": "CA",  "zipcode": 44444,  "\_id": "5719a7164cf09c803238781a"  }  ]  } | |  |  |  | | --- | --- | | Status | 403 | | Content-Type | application/json | | {  "message" : "Not Authorized"  "success" : false,  "status" : 403  } | | | | |

|  |  |
| --- | --- |
| POST | /user/removeaddress/:address |
| Remove one address | |
| |  | | --- | | Request | | POST /user/removeaddress/Howold street   |  |  | | --- | --- | | 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": []  } | |  |  |  | | --- | --- | | Status | 403 | | Content-Type | application/json | | {  "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:**

|  |  |
| --- | --- |
| 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 | | number | The number of dish user added | | Price | The price of this dish, if the number of dish is two, the price should calculated as price of two dishes | | total\_price | The 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": "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 |   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": "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  } | | | | |

|  |  |
| --- | --- |
| 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:

|  |  |
| --- | --- |
| 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  <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": [  {  "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  <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": "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  } | | | | |

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:

* 1. Menu Display
     1. Menu will display in catalog(starter, main course, noodle, rice, beverage, dessert, soup).
     2. Each dish should show picture, name, price, materia,comments
     3. 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.
     4. Main page should have special dishes for today.
     5. 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:

* + 1. 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.
    2. 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.
    3. 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.
    4. 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).