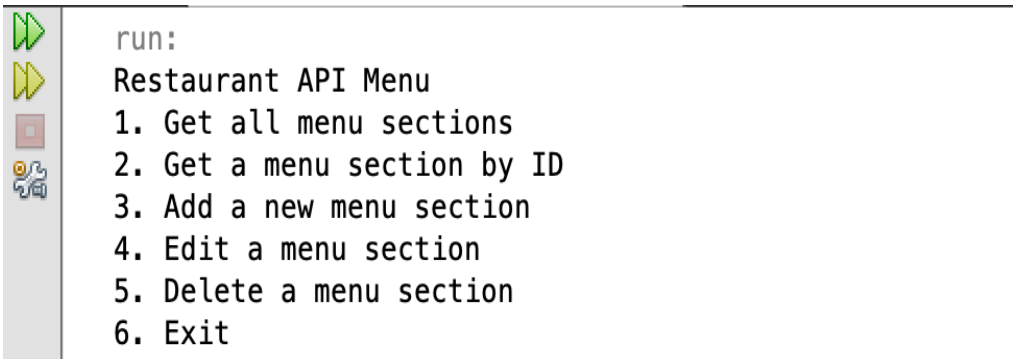
This is the simple restaurant api

There are 2 folders, one for client, one for server. I already deploy the server to heroku, so when running the project, we don't need to run the server locally, only need to download the client side files.

- If you have NetBeans, please clone the RestaurantApiClient folder, zip it, open NetBeans, Click File -> import Project -> From zip, and choose the zip file. Run this project to start the client side.
- If you have eclipse, please create a new java project in eclipse, go to RestaurantApiClient folder -> src -> RestaurantApiClient.java, download RestaurantApiClient.java and paste it in your project, run the java file to start the client side; Also, please go to RestaurantApiClient folder -> lib -> java-json.jar, download the jar and import it into your project; If you are using default package in the java project, please delete “package restaurantapiclient;” in line 2 in RestaurantApiClient.java.

After running the client in NetBeans, it will show the menu like this:



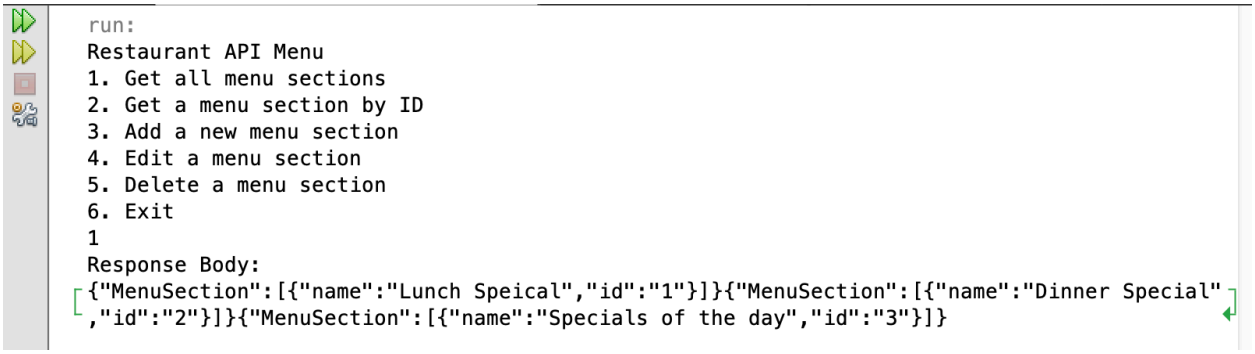
1) Get a menu section by id

Type 2, then type the id you want to search. It will show the response from server.

```
Restaurant API Menu
1. Get all menu sections
2. Get a menu section by ID
3. Add a new menu section
4. Edit a menu section
5. Delete a menu section
6. Exit
2
Please insert the ID
1
Response Body:
{"MenuSection":[{"name":"Lunch Speical","id":"1"}]}
```

2) Get all menu sections

Type 1, and it will show the response from the server that contains all the records stored in the database.



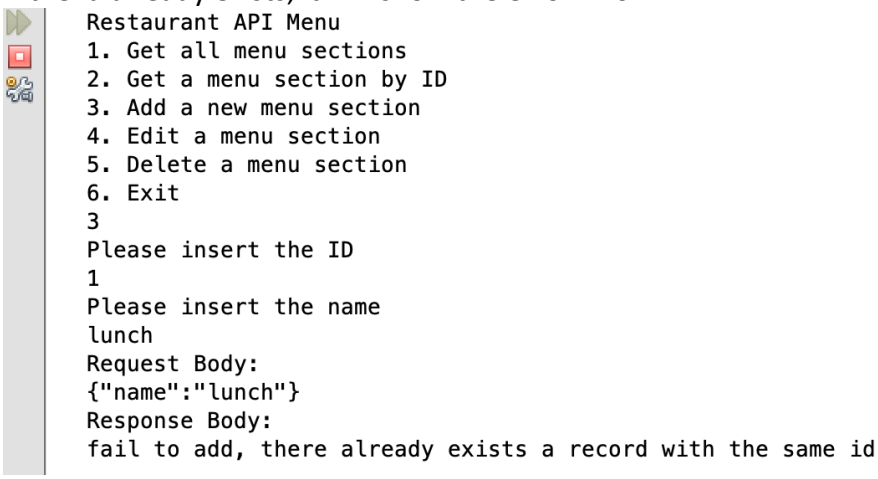
```
run:
Restaurant API Menu
1. Get all menu sections
2. Get a menu section by ID
3. Add a new menu section
4. Edit a menu section
5. Delete a menu section
6. Exit
1
Response Body:
{"MenuSection":[{"name":"Lunch Speical","id":"1"}]}{"MenuSection":[{"name":"Dinner Special"
,"id":"2"}]}{"MenuSection":[{"name":"Specials of the day","id":"3"}]}
```

3) Add a new menu section

Type 3, input id and name, and it will show the response from the server.

```
Restaurant API Menu
1. Get all menu sections
2. Get a menu section by ID
3. Add a new menu section
4. Edit a menu section
5. Delete a menu section
6. Exit
3
Please insert the ID
4
Please insert the name
lunch
Request Body:
{"name":"lunch"}
Response Body:
{"success":true,"MenuSection":[{"name":"lunch","id":"4"}]}
```

If the id already exists, it will show the error info.



```
Restaurant API Menu
1. Get all menu sections
2. Get a menu section by ID
3. Add a new menu section
4. Edit a menu section
5. Delete a menu section
6. Exit
3
Please insert the ID
1
Please insert the name
lunch
Request Body:
{"name":"lunch"}
Response Body:
fail to add, there already exists a record with the same id
```

4) Edit a menu section

Type 4, input id and the new name, and it will show the response from the server.

```
Restaurant API Menu
1. Get all menu sections
2. Get a menu section by ID
3. Add a new menu section
4. Edit a menu section
5. Delete a menu section
6. Exit
4
Please insert the ID
2
Please insert the name
lunch specials
Request Body:
{"name":"lunch specials"}
Response Body:
{"success":true,"MenuSection":[{"name":"lunch specials","id":"2"}]}
```

If the id doesn't exist, it will show the error info.

```
Restaurant API Menu
1. Get all menu sections
2. Get a menu section by ID
3. Add a new menu section
4. Edit a menu section
5. Delete a menu section
6. Exit
4
Please insert the ID
4
Please insert the name
4
Request Body:
{"name":"4"}
Response Body:
fail to edit, there isn't a record with the id
```

5) Delete a menu section

Type 5, input id and the new name, and it will show the response from the server.

```
Restaurant API Menu
1. Get all menu sections
2. Get a menu section by ID
3. Add a new menu section
4. Edit a menu section
5. Delete a menu section
6. Exit
5
Please insert the ID
4
{"success":true}
```