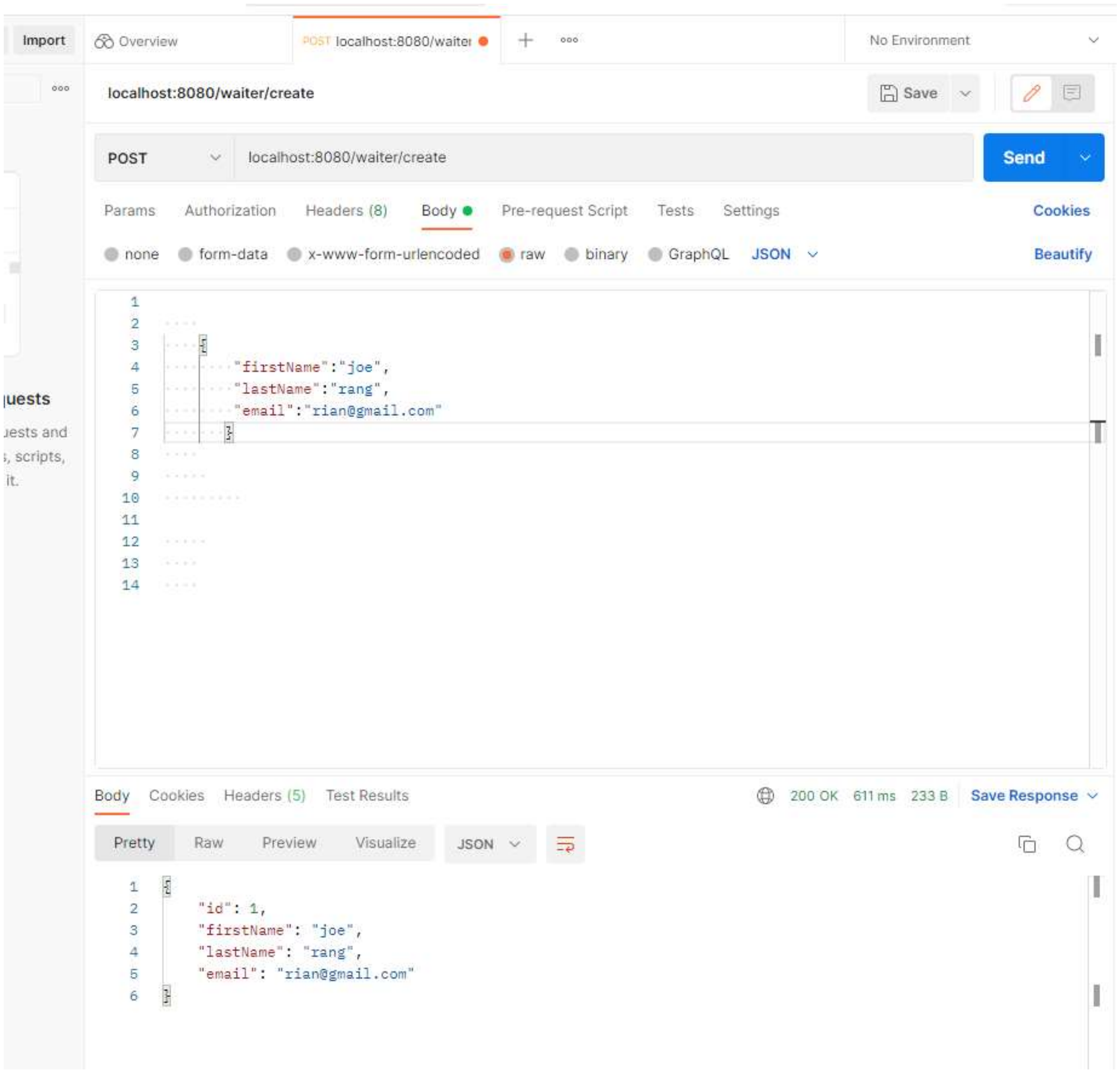
Use Postman and Tests

1.create 1



Import

Overview

POST localhost:8080/waiter

No Environment

localhost:8080/waiter/create

Save

uests

uests and

, scripts,

it.

POST localhost:8080/waiter/create

Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded [✓ raw] binary GraphQL JSON Beautify

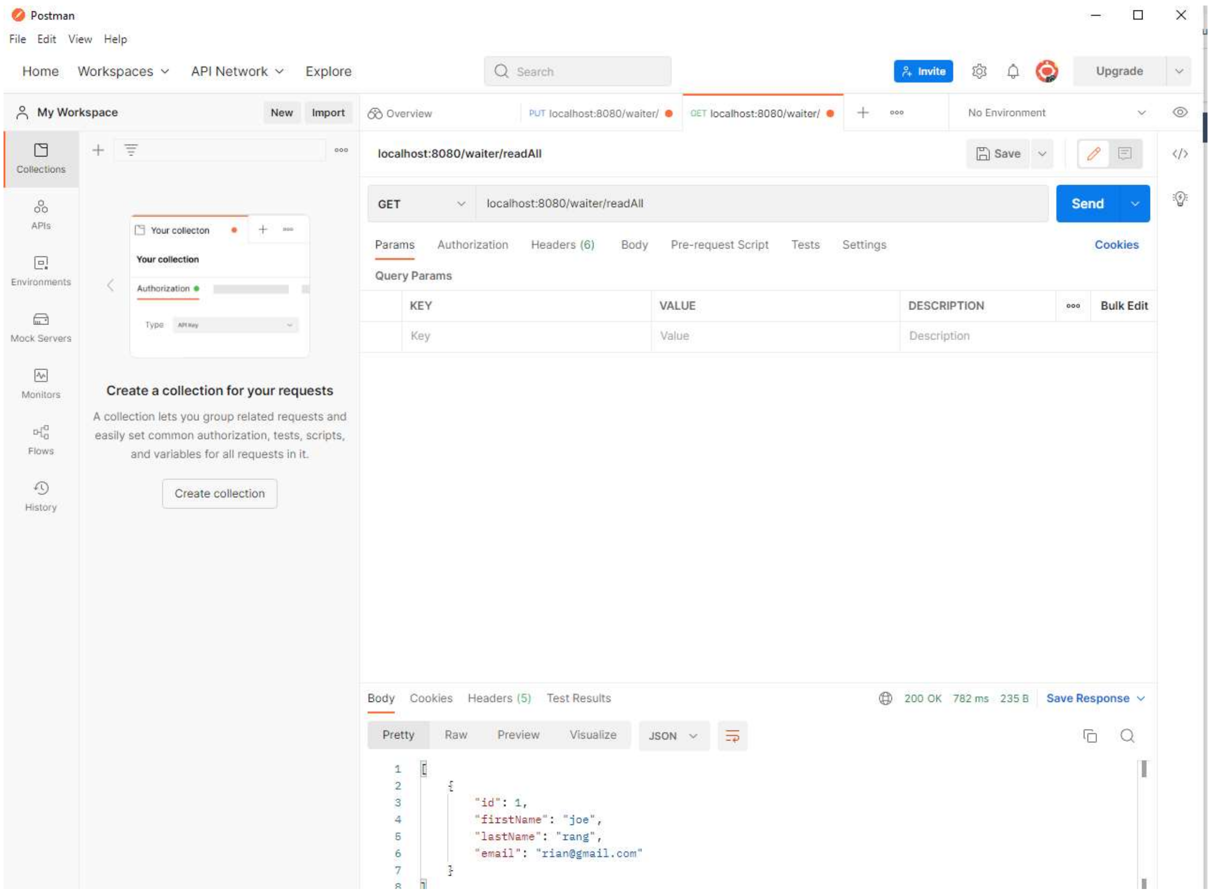
```
{
    "firstName":"joe",
    "lastName":"rang",
    "email":"rian@gmail.com"
}
```

Body Cookies Headers (5) Test Results 200 OK 611 ms 233 B Save Response

Pretty Raw Preview Visualize JSON

```
{
    "id": 1,
    "firstName": "joe",
    "lastName": "rang",
    "email": "rian@gmail.com"
}
```

read all


Postman
File Edit View Help
Home Workspaces API Network Explore
Search
Invite
Upgrade
My Workspace
New Import
Overview
PUT localhost:8080/waiter/
GET localhost:8080/waiter/
No Environment
Collections
APIs
Environments
Mock Servers
Monitors
Flows
History
Create a collection for your requests
A collection lets you group related requests and easily set common authorization, tests, scripts, and variables for all requests in it.
Create collection
Save
Send
Params Authorization Headers (6) Body Pre-request Script Tests Settings
Cookies
Query Params
KEY VALUE DESCRIPTION Bulk Edit
Body Cookies Headers (5) Test Results
Save Response
Pretty Raw Preview Visualize JSON

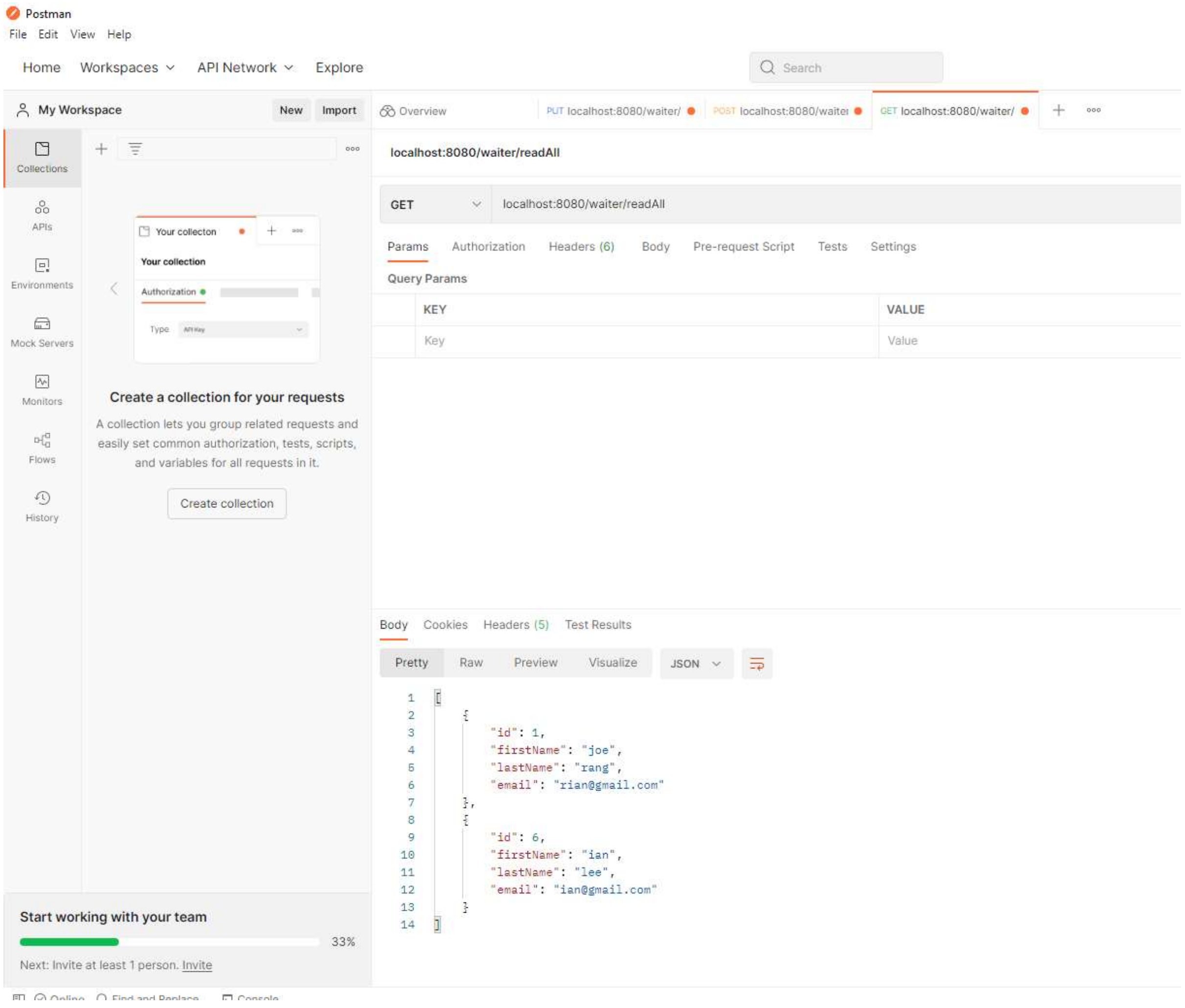
localhost:8080/waiter/readAll

GET localhost:8080/waiter/readAll

200 OK 782 ms 235 B

```
[
    {
        "id": 1,
        "firstName": "joe",
        "lastName": "rang",
        "email": "rian@gmail.com"
    }
]
```

after creating 2 read all

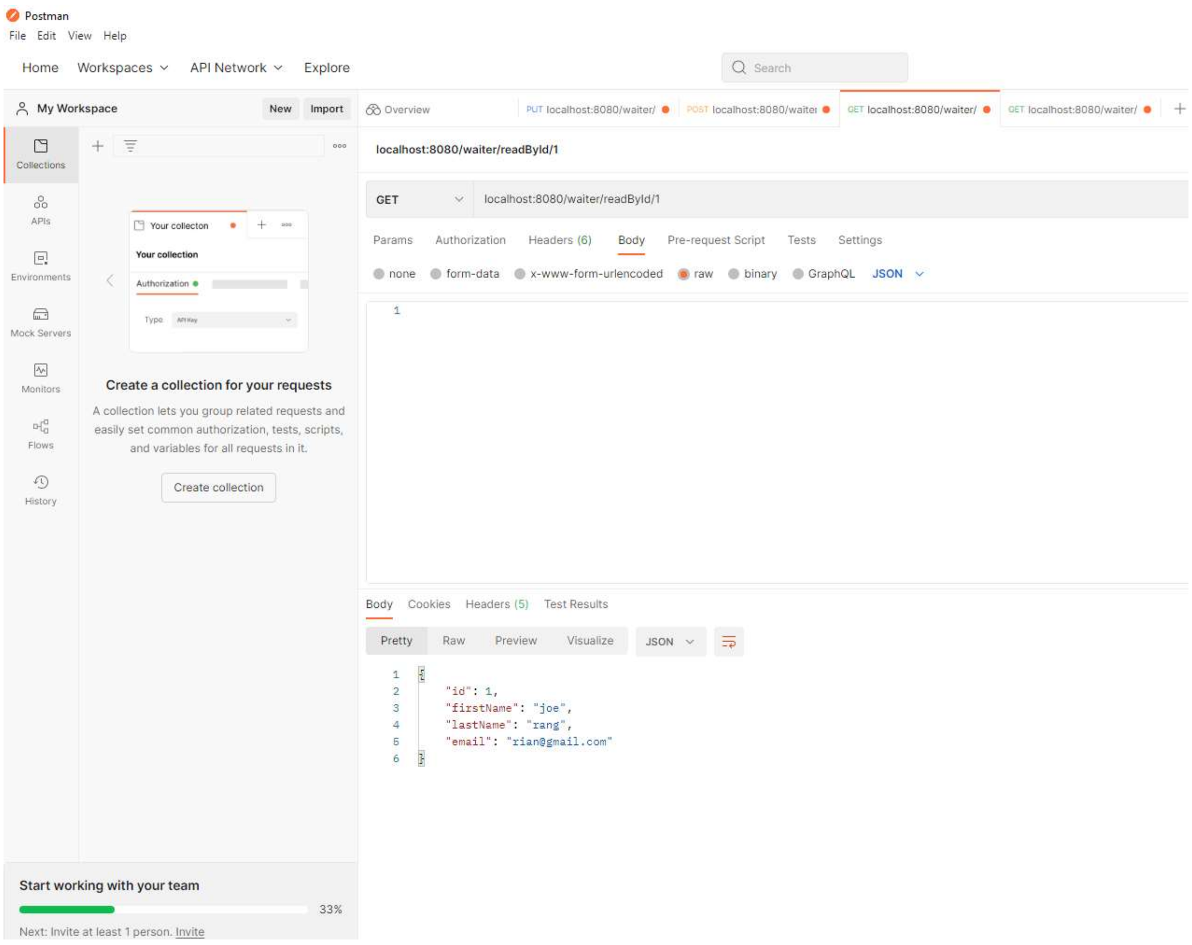

Postman
File Edit View Help
Home Workspaces API Network Explore
Search
My Workspace New Import
Overview
PUT localhost:8080/waiter/
POST localhost:8080/waiter
GET localhost:8080/waiter/
Collections
APIs
Environments
Mock Servers
Monitors
Flows
History
Create a collection for your requests
A collection lets you group related requests and easily set common authorization, tests, scripts, and variables for all requests in it.
Create collection
Start working with your team
33%
Next: Invite at least 1 person. Invite
Params Authorization Headers (6) Body Pre-request Script Tests Settings
Query Params
KEY VALUE
Body Cookies Headers (5) Test Results
Pretty Raw Preview Visualize JSON

localhost:8080/waiter/readAll

GET localhost:8080/waiter/readAll

```
[
    {
        "id": 1,
        "firstName": "joe",
        "lastName": "rang",
        "email": "rian@gmail.com"
    },
    {
        "id": 6,
        "firstName": "ian",
        "lastName": "lee",
        "email": "ian@gmail.com"
    }
]
```

read one

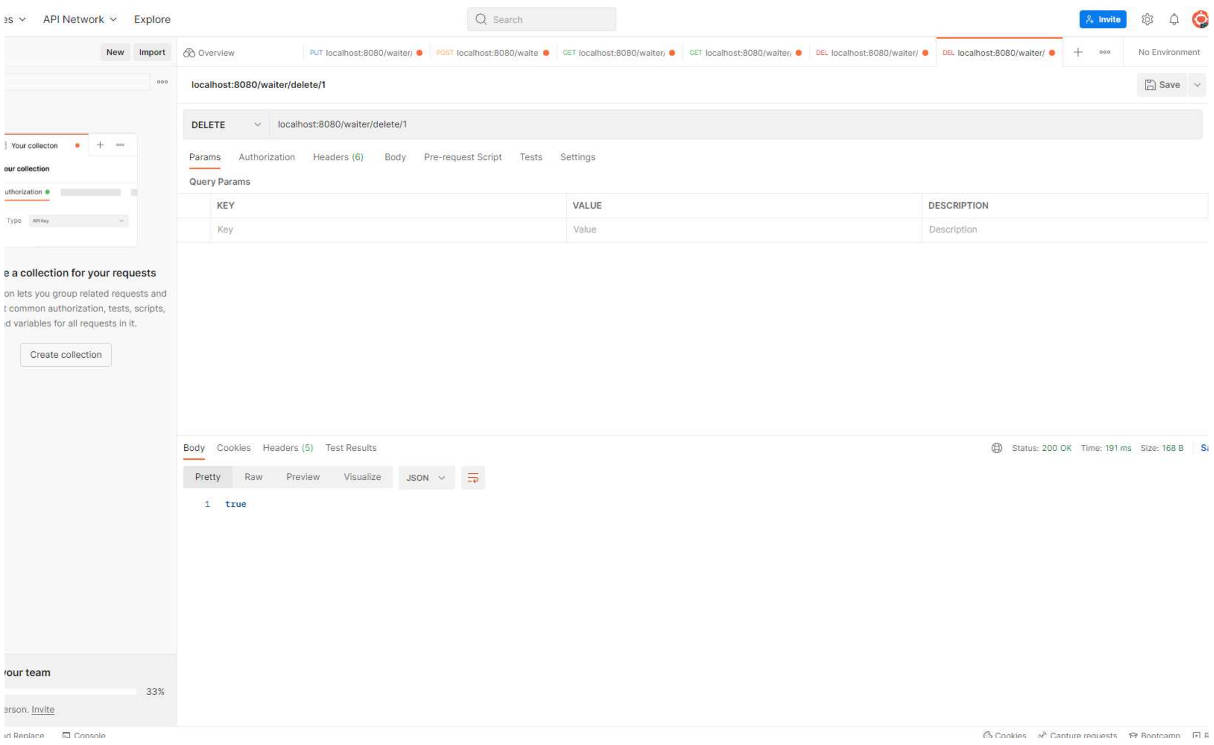

Postman
File Edit View Help
Home Workspaces API Network Explore
Search
My Workspace New Import
Collections
APIs
Environments
Mock Servers
Monitors
Flows
History
Create a collection for your requests
A collection lets you group related requests and easily set common authorization, tests, scripts, and variables for all requests in it.
Create collection
Start working with your team 33%
Next: Invite at least 1 person. Invite
Overview
PUT localhost:8080/waiter/
POST localhost:8080/waiter
GET localhost:8080/waiter/
GET localhost:8080/waiter/
Params Authorization Headers (6) Body Pre-request Script Tests Settings
none form-data x-www-form-urlencoded raw binary GraphQL JSON
Body Cookies Headers (5) Test Results
Pretty Raw Preview Visualize JSON

localhost:8080/waiter/readById/1

GET localhost:8080/waiter/readById/1

```
{
    "id": 1,
    "firstName": "joe",
    "lastName": "rang",
    "email": "rian@gmail.com"
}
```

delete

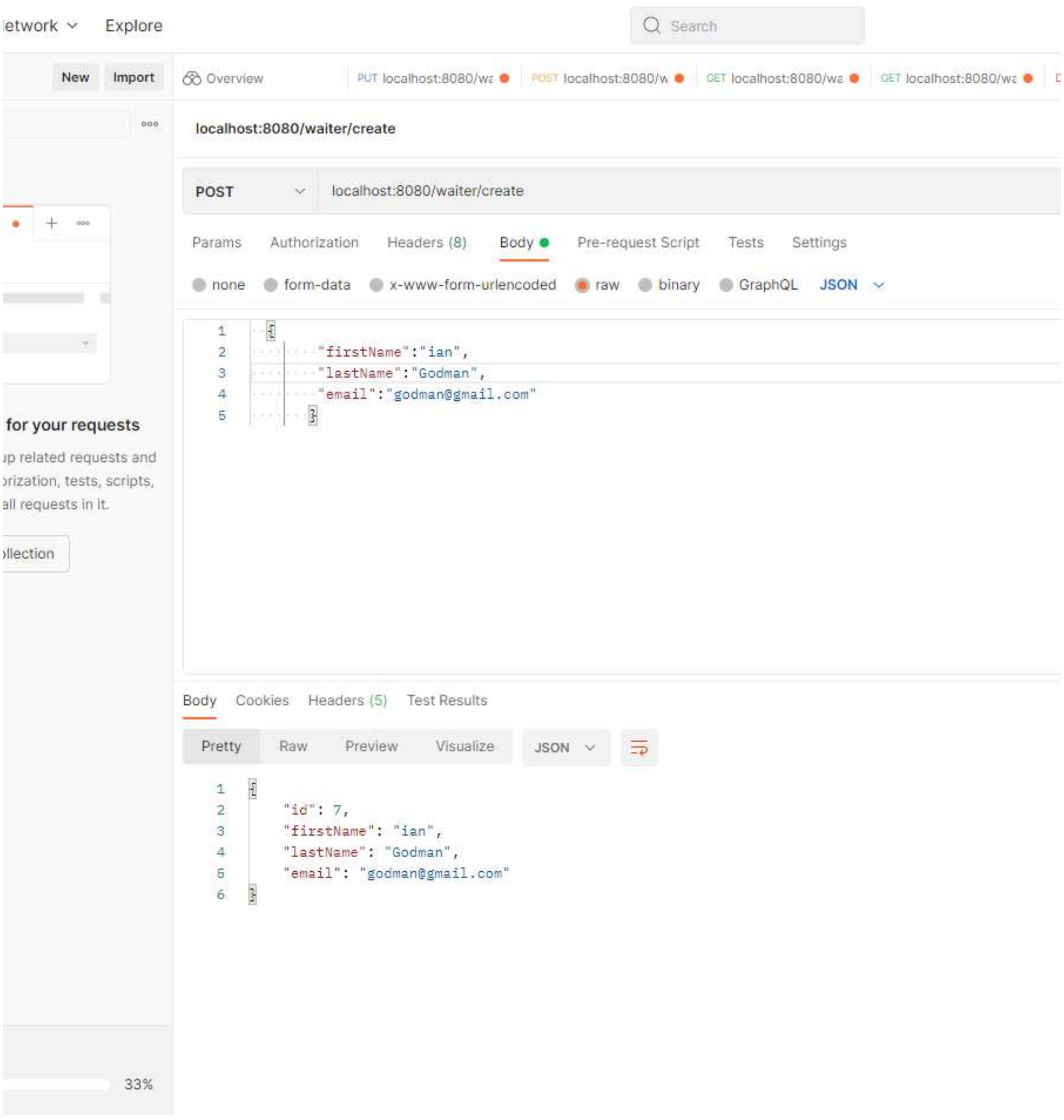

localhost:8080/waiter/delete/1

DELETE localhost:8080/waiter/delete/1

Status: 200 OK Time: 191 ms Size: 168 B

```
true
```

Create id 7



localhost:8080/waiter/create

POST localhost:8080/waiter/create

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded [✓ raw] binary GraphQL JSON

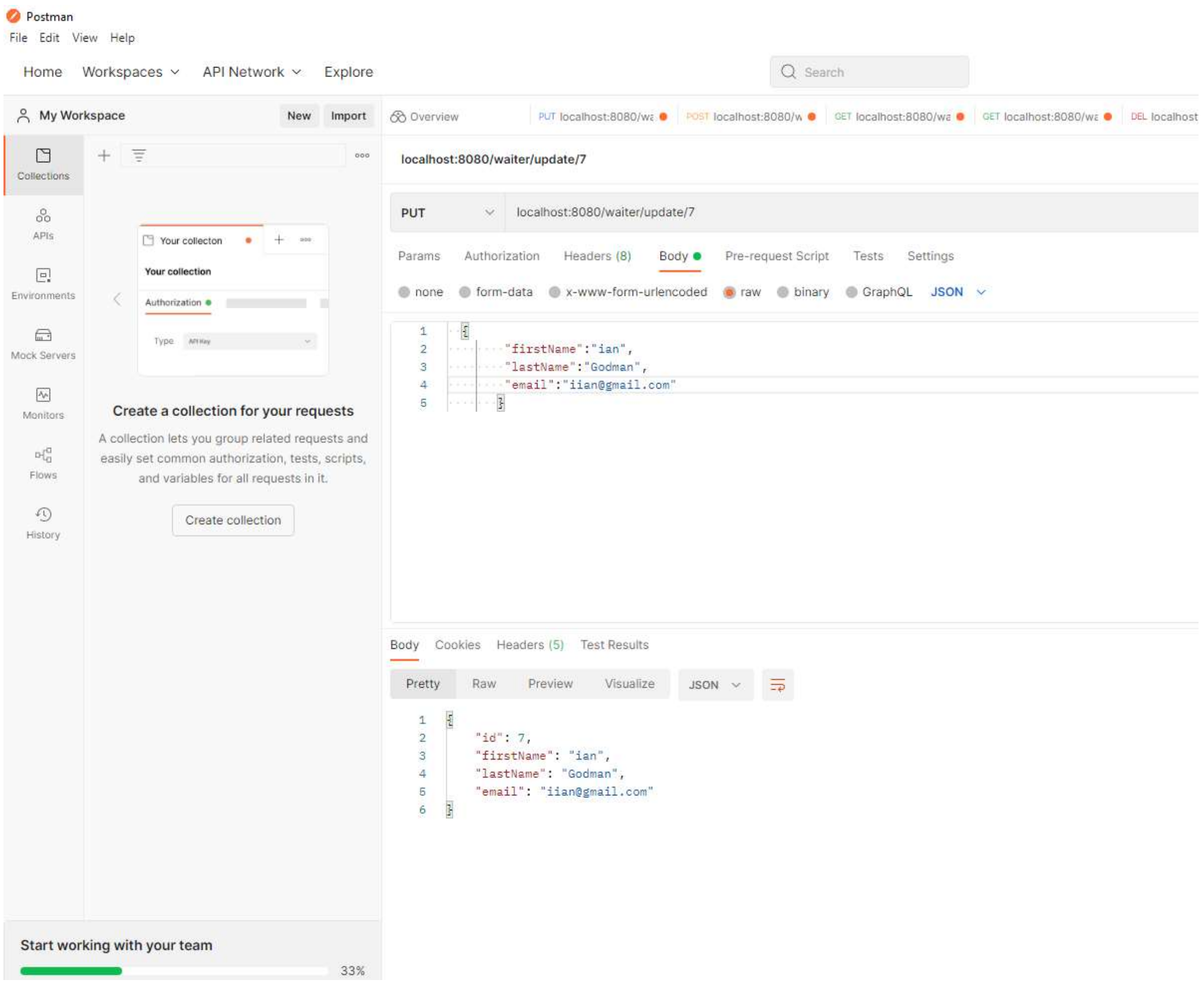
```
{
    "firstName":"ian",
    "lastName":"Godman",
    "email":"godman@gmail.com"
  }
```

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON

```
{
    "id": 7,
    "firstName": "ian",
    "lastName": "Godman",
    "email": "godman@gmail.com"
}
```

Update id 7

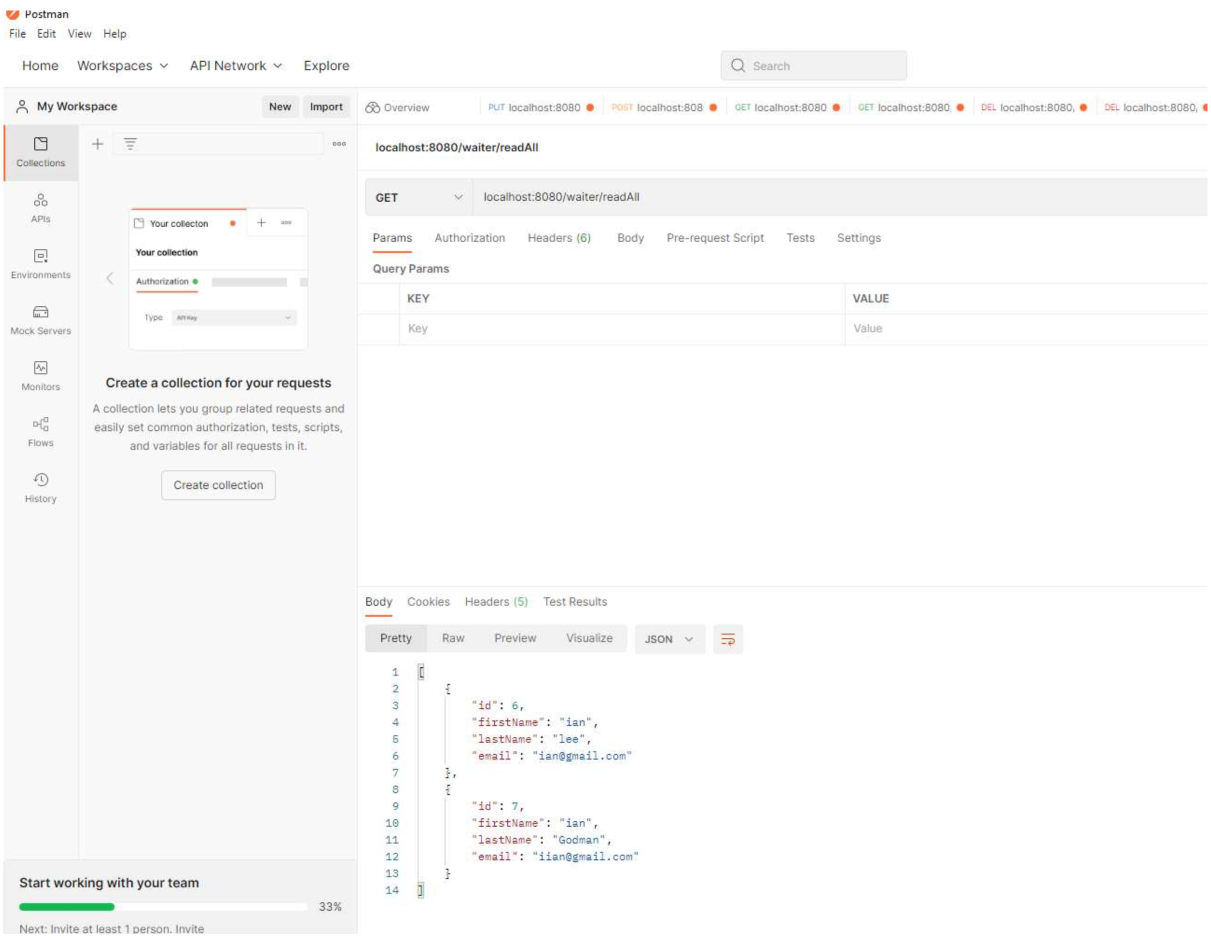

PUT localhost:8080/waiter/update/7

```
{
    "firstName":"ian",
    "lastName":"Godman",
    "email":"iian@gmail.com"
}
```

```
{
    "id": 7,
    "firstName": "ian",
    "lastName": "Godman",
    "email": "iian@gmail.com"
}
```

Then read all, changed new:

After update the email have changed.

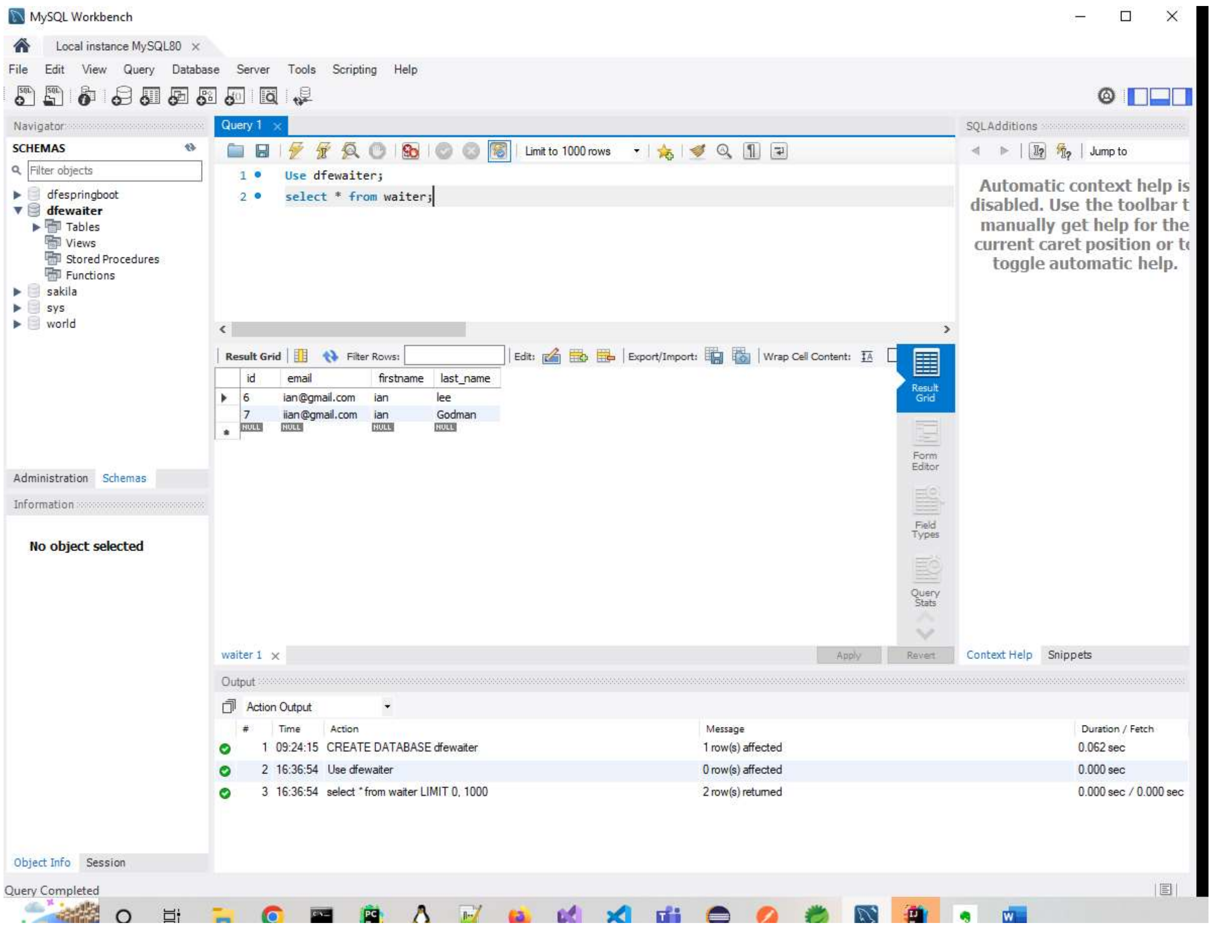

Postman
File Edit View Help
Home Workspaces API Network Explore
Search
My Workspace New Import
Overview
PUT localhost:8080
POST localhost:808
GET localhost:8080
GET localhost:8080
DEL localhost:8080,
DEL localhost:8080,
Collections
APIs
Environments
Mock Servers
Monitors
Flows
History
Your collecton
Your collection
Authorization
Type API Key
Create a collection for your requests
A collection lets you group related requests and easily set common authorization, tests, scripts, and variables for all requests in it.
Create collection
Start working with your team
33%
Next: Invite at least 1 person. Invite
Params Authorization Headers (6) Body Pre-request Script Tests Settings
Query Params
KEY
VALUE
Key
Value
Body Cookies Headers (5) Test Results
Pretty Raw Preview Visualize JSON

localhost:8080/waiter/readAll

GET localhost:8080/waiter/readAll

```
[
    {
        "id": 6,
        "firstName": "ian",
        "lastName": "lee",
        "email": "ian@gmail.com"
    },
    {
        "id": 7,
        "firstName": "ian",
        "lastName": "Godman",
        "email": "iian@gmail.com"
    }
]
```

My SQL

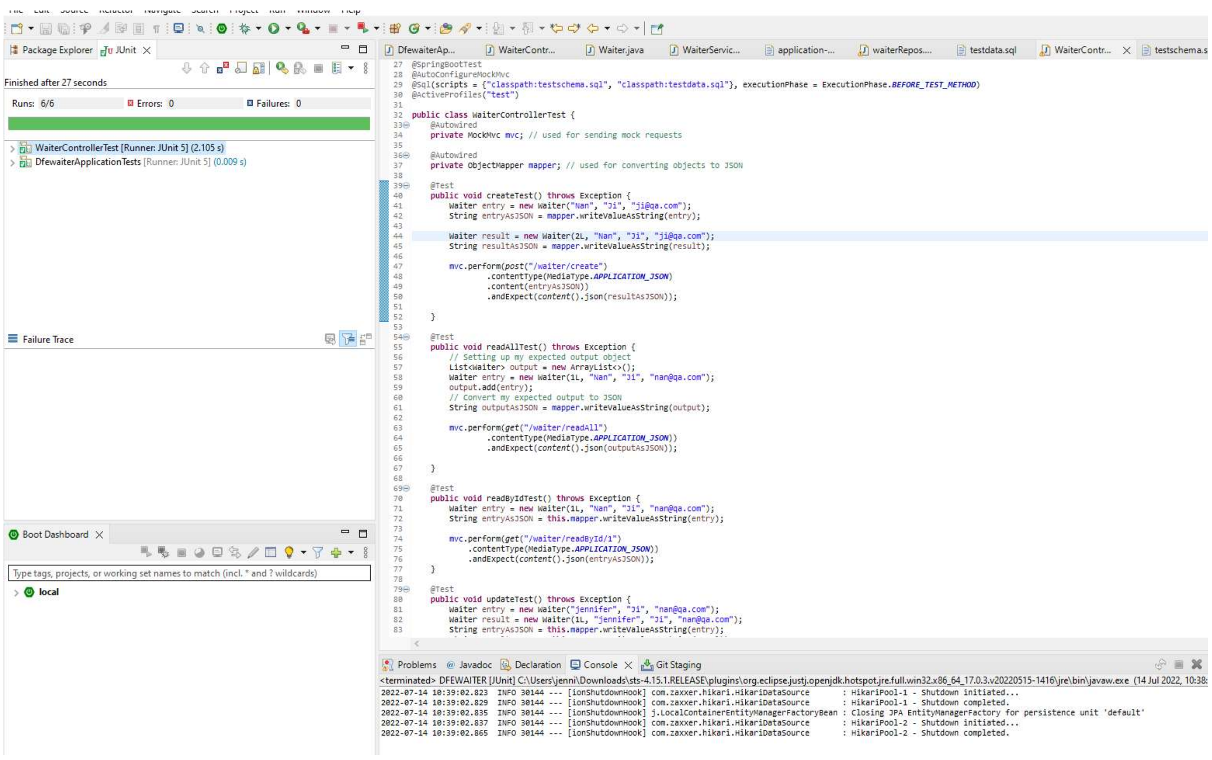


```
Use dfewaiter;
select * from waiter;
```

id	email	firstname	last_name
6	ian@gmail.com	ian	lee
7	iian@gmail.com	ian	Godman
NULL	NULL	NULL	NULL

```
#  Time      Action                              Message             Duration / Fetch
1  09:24:15  CREATE DATABASE dfewaiter           1 row(s) affected   0.062 sec
2  16:36:54  Use dfewaiter                       0 row(s) affected   0.000 sec
3  16:36:54  select * from waiter LIMIT 0, 1000  2 row(s) returned   0.000 sec / 0.000 sec
```

After run Test (run JUnit test, and screenshot)



Finished after 27 seconds

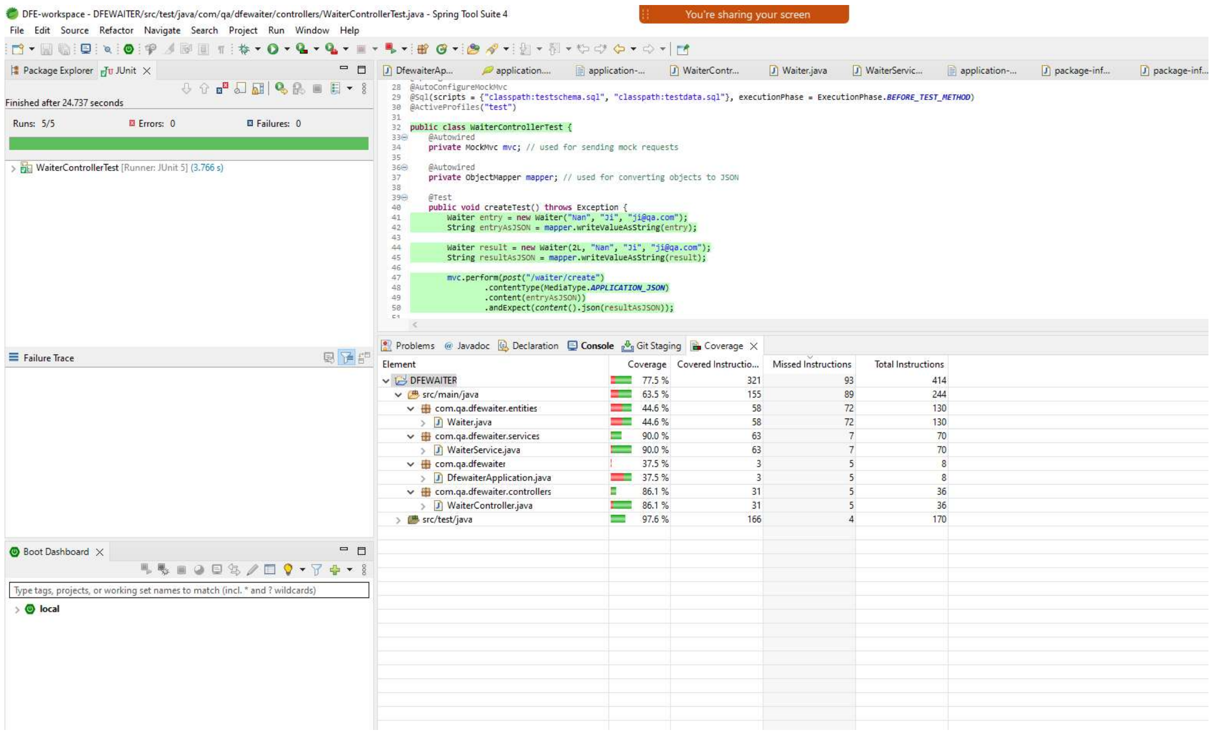
Runs: 6/6 Errors: 0 Failures: 0

WaiterControllerTest [Runner: JUnit 5] (2.105 s)
DfewaiterApplicationTests [Runner: JUnit 5] (0.009 s)

```java
@SpringBootTest
@AutoConfigureMockMvc
@Sql(scripts = {"classpath:testschema.sql", "classpath:testdata.sql"}, executionPhase = ExecutionPhase.BEFORE_TEST_METHOD)
@ActiveProfiles("test")

public class WaiterControllerTest {
    @Autowired
    private MockMvc mvc; // used for sending mock requests

    @Autowired
    private ObjectMapper mapper; // used for converting objects to JSON

    @Test
    public void createTest() throws Exception {
        Waiter entry = new Waiter("Nan", "Ji", "ji@qa.com");
        String entryAsJSON = mapper.writeValueAsString(entry);

        Waiter result = new Waiter(2L, "Nan", "Ji", "ji@qa.com");
        String resultAsJSON = mapper.writeValueAsString(result);

        mvc.perform(post("/waiter/create")
                .contentType(MediaType.APPLICATION_JSON)
                .content(entryAsJSON))
                .andExpect(content().json(resultAsJSON));

    }

    @Test
    public void readAllTest() throws Exception {
        // Setting up my expected output object
        List<Waiter> output = new ArrayList<>();
        Waiter entry = new Waiter(1L, "Nan", "Ji", "nan@qa.com");
        output.add(entry);
        // Convert my expected output to JSON
        String outputAsJSON = mapper.writeValueAsString(output);

        mvc.perform(get("/waiter/readAll")
                .contentType(MediaType.APPLICATION_JSON))
                .andExpect(content().json(outputAsJSON));

    }

    @Test
    public void readByIdTest() throws Exception {
        Waiter entry = new Waiter(1L, "Nan", "Ji", "nan@qa.com");
        String entryAsJSON = this.mapper.writeValueAsString(entry);

        mvc.perform(get("/waiter/readById/1")
            .contentType(MediaType.APPLICATION_JSON))
            .andExpect(content().json(entryAsJSON));
    }

    @Test
    public void updateTest() throws Exception {
        Waiter entry = new Waiter("jennifer", "Ji", "nan@qa.com");
        Waiter result = new Waiter(1L, "jennifer", "Ji", "nan@qa.com");
        String entryAsJSON = this.mapper.writeValueAsString(entry);
```

```
<terminated> DFEWAITER [JUnit] C:\Users\jenni\Downloads\sts-4.15.1.RELEASE\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.3.v20220515-1416\jre\bin\javaw.exe (14 Jul 2022, 10:38:
2022-07-14 10:39:02.823  INFO 30144 --- [ionShutdownHook] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Shutdown initiated...
2022-07-14 10:39:02.829  INFO 30144 --- [ionShutdownHook] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Shutdown completed.
2022-07-14 10:39:02.835  INFO 30144 --- [ionShutdownHook] j.LocalContainerEntityManagerFactoryBean : Closing JPA EntityManagerFactory for persistence unit 'default'
2022-07-14 10:39:02.837  INFO 30144 --- [ionShutdownHook] com.zaxxer.hikari.HikariDataSource       : HikariPool-2 - Shutdown initiated...
2022-07-14 10:39:02.865  INFO 30144 --- [ionShutdownHook] com.zaxxer.hikari.HikariDataSource       : HikariPool-2 - Shutdown completed.
```

Another test



DFE-workspace - DFEWAITER/src/test/java/com/qa/dfewaiter/controllers/WaiterControllerTest.java - Spring Tool Suite 4

You're sharing your screen

Finished after 24.737 seconds

Runs: 5/5 Errors: 0 Failures: 0

WaiterControllerTest [Runner: JUnit 5] (3.766 s)

```java
@AutoConfigureMockMvc
@Sql(scripts = {"classpath:testschema.sql", "classpath:testdata.sql"}, executionPhase = ExecutionPhase.BEFORE_TEST_METHOD)
@ActiveProfiles("test")

public class WaiterControllerTest {
    @Autowired
    private MockMvc mvc; // used for sending mock requests

    @Autowired
    private ObjectMapper mapper; // used for converting objects to JSON

    @Test
    public void createTest() throws Exception {
        Waiter entry = new Waiter("Nan", "Ji", "ji@qa.com");
        String entryAsJSON = mapper.writeValueAsString(entry);

        Waiter result = new Waiter(2L, "Nan", "Ji", "ji@qa.com");
        String resultAsJSON = mapper.writeValueAsString(result);

        mvc.perform(post("/waiter/create")
                .contentType(MediaType.APPLICATION_JSON)
                .content(entryAsJSON))
                .andExpect(content().json(resultAsJSON));
```

Element	Coverage	Covered Instructio...	Missed Instructions	Total Instructions
DFEWAITER	77.5 %	321	93	414
src/main/java	63.5 %	155	89	244
com.qa.dfewaiter.entities	44.6 %	58	72	130
Waiter.java	44.6 %	58	72	130
com.qa.dfewaiter.services	90.0 %	63	7	70
WaiterService.java	90.0 %	63	7	70
com.qa.dfewaiter	37.5 %	3	5	8
DfewaiterApplication.java	37.5 %	3	5	8
com.qa.dfewaiter.controllers	86.1 %	31	5	36
WaiterController.java	86.1 %	31	5	36
src/test/java	97.6 %	166	4	170