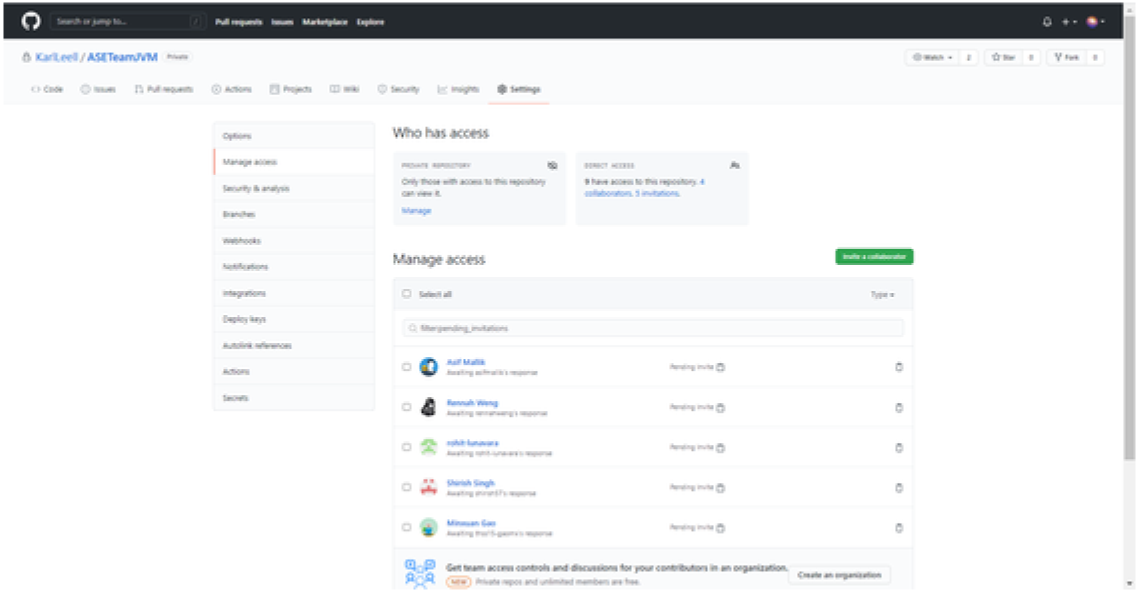
Assignment T3: First Iteration

Team JVM - Assignment T3

Part 1 Project Page

Link to GitHub Repo: https://github.com/KarlLeell/ASETeamJVM

The collaborator invitation has been sent to every IA while none accepted yet.



Who has access

PRIVATE REPOSITORY

Only those with access to this repository can view it.

Manage

DIRECT ACCESS

9 have access to this repository. 4 collaborators. 5 invitations.

Manage access

filter:pending_invitations

Asif Mallik — Pending invite

Rennah Weng — Pending invite

rohit lunavara — Pending invite

Shirish Singh — Pending invite

Minxuan Gao — Pending invite

Get team access controls and discussions for your contributors in an organization.

Private repos and unlimited members are free.

Part 2 User Story

1. As a user, I want to be able to log in to the app so I can access my history from different devices if I am logged in.
   My conditions of satisfaction are:
   a. No other user has the same username as I do.
   b. My account is protected by a password that only I know
   c. The website remembers that I have logged into the device for some period. During this period, I do not need to log in again if I visit the website again.
   d. If I do not visit the website for a long time (e.g. a day), the website should automatically log me out, so that other users that access the same device will not be able to use my account.
2. As a user, I want to record my daily diet in the app so that I can see my diet history and get some statistics about my past diet records.
   My conditions of satisfaction are:
   a. The app supports at least 5 types of food for me to choose from for each record
   b. I can modify or delete the created diet record in case I made a mistake
   c. The records are available on any device as long as I am logged in to the account.
   d. The app can provide me with accurate statistics on the food I eat
   e. I can see my history in a different time period (1 week, 1 month, 1 year)
3. As a user, I want to record my weight and on a daily basis so that I can track the changes in my weight throughout any period.
   My conditions of satisfaction are:
   a. I can record my weight in either pounds or kilograms.
   b. I can view my weight in either pounds or kilograms no matter which unit I used when entering it.
   c. I can modify or delete the created weight record in case I made a mistake
   d. The records are available on any device as long as I am logged in to the account.
4. As a user, I want to record my workout history so that I can keep track of the time and intensity of my past workouts.
   My conditions of satisfaction are:
   a. The app can recognize different types of workout from the records I input
   b. The app can provide me with statistics of my workout based on workout type and duration.
   c. I can modify or delete the created workout record in case I made a mistake

    d. I should receive a warning when I am trying to delete a record or modify sensitive attributes of the record like date

We remove the display and advice sections temporarily because we have not implemented the frontend and the advice generator. The display section will be added back and tested once we finish the frontend implementation. If there is time after all the core features are correctly implemented, we will implement the advice generator.

User Story Testing Plan and Results

We plan to use both programmatic and manual testing to ensure the correct implementation of our system. The detailed test inputs and outputs are shown in the appendix at the end of this file

1. User register/login
    a. Register performs as expected with both unregistered and registered username. When an unregistered username is passed to the register API then a new user is added to the database; when a registered username is passed in then an error message is returned.
    b. Login performs as expected with both correct and incorrect username-password pair. When a correct pair is passed to the login API then an auth token is generated, recorded in the database, and returned to the user; when the pair is incorrect then an error message is returned
    c. **TODO:** Currently, User authentication does not work properly with other services. This means that a user does not need to have an auth token to get services. We will fix this before the next iteration.
2. Diet services
    a. ADD diet records performs as expected with both valid and invalid input (invalid input contains unrecognized weight units such as “kilogram”). When the input is valid, the record is successfully written to the database; when the input is not valid then the record is not written.
    b. GET diet records performs as expected with both valid and invalid selections of time length (invalid time length contain unrecognized units such as “quarter” and illegal length such as “-3“). The returns for valid input and the error messages for invalid input accord with expectation.
    c. EDIT diet records performs as expected with both valid and invalid input (invalid input contains unrecognized diet history ID). When the input is valid, the record is successfully updated in the database; when the input is not valid then the record is not updated.
    d. DELETE diet records performs as expected with both valid and invalid input (invalid input contains unrecognized diet history ID). When the input is valid, the record is successfully deleted in the database; when the input is not valid then the record is not deleted.
3. Weight services
    a. ADD weight records performs as expected with both valid and invalid input (invalid input contains unrecognized weight units such as “kilogram”). When the input is valid, the record is successfully written to the database; when the input is not valid then the record is not written.
    b. GET weight records performs as expected with both valid and invalid selections of time length (invalid time length contain unrecognized units such as “quarter”). The returns for valid input and the error messages for invalid input accord with expectation.
    c. EDIT weight records performs as expected with both valid and invalid input (invalid input contains unrecognized weight history ID). When the input is valid, the record is successfully updated in the database; when the input is not valid then the record is not updated.
    Bug found and fixed: Weight ID is stored as Integer in DB while parameter passed to DB calls have String.
    d. DELETE weight records performs as expected with both valid and invalid input (invalid input contains unrecognized weight history ID). When the input is valid, the record is successfully deleted in the database; when the input is not valid then the record is not deleted.
    Bug found and fixed: Weight ID is stored as Integer in DB while parameter passed to DB calls have String.
4. Exercise services
    a. ADD exercise records performs as expected with valid and some input (unrecognized exercise type and user). When the input is valid, the record is successfully written to the database; when the input is not valid then the record is not written.
    Bug found and fixed: Non-positive duration should also be invalid input.
    b. GET exercise records performs as expected with both valid and invalid selections of time length (invalid time length contain unrecognized units such as “quarter”). The returns for valid input and the error messages for invalid input accord with expectation.
    c. EDIT exercise records performs as expected with valid and some invalid input (invalid input contains unrecognized exercise history ID, invalid exercise type, and invalid user). When the input is valid, the record is successfully updated in the database; when the input is not valid then the record is not updated.
    Bug found and fixed: Non-positive duration should also be invalid input.
    d. DELETE exercise records performs as expected with both valid and invalid input (invalid input contains unrecognized exercise history ID). When the input is valid, the record is successfully deleted in the database; when the input is not valid then the record is not deleted.
5. **General Bug found**: For some exceptions, the error message does not show up correctly. Will fix soon.

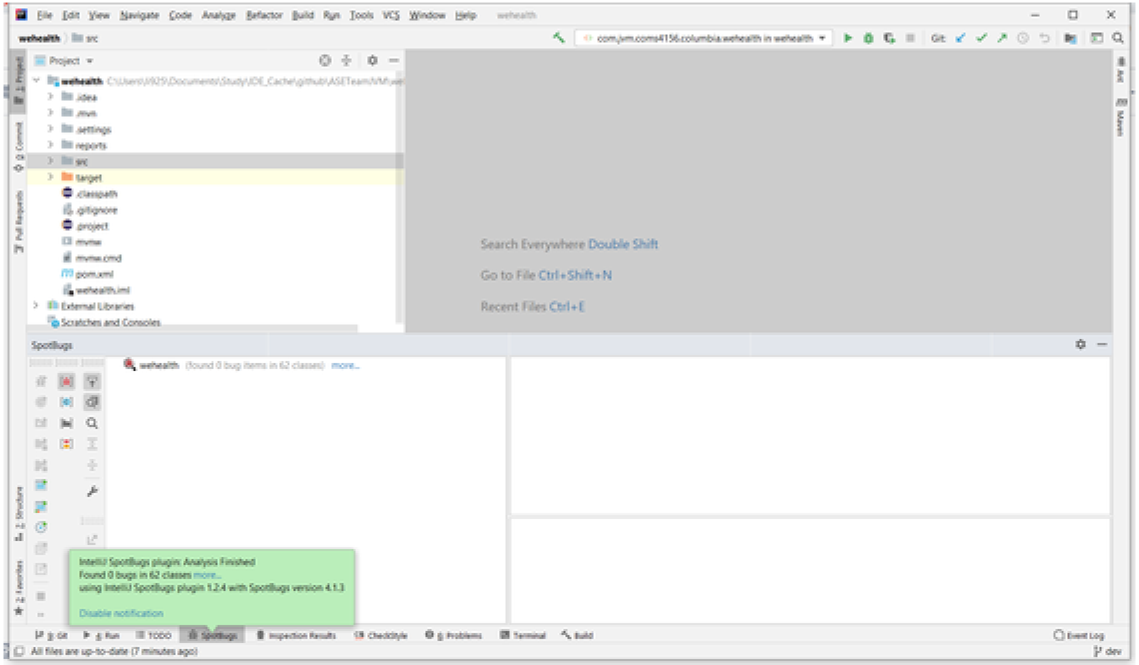
Part 3

Link to unit tests: https://github.com/KarlLeell/ASETeamJVM/tree/dev/wehealth/src/test/java/com/jvm/coms4156/columbia/wehealth

Link to Maven file: https://github.com/KarlLeell/ASETeamJVM/blob/dev/wehealth/pom.xml

Part 4

Link to latest SpotBug report: https://github.com/KarlLeell/ASETeamJVM/tree/dev/wehealth/reports (after the fixes, there is no bug, and the plugin does not allow exporting the empty report, so we only include the scanned files and a screenshot here)



wehealth (found 0 bug items in 62 classes) more...

IntelliJ SpotBugs plugin: Analysis Finished
Found 0 bugs in 62 classes more...
using IntelliJ SpotBugs plugin 1.2.4 with SpotBugs version 4.1.3

Disable notification

All files are up-to-date (7 minutes ago)

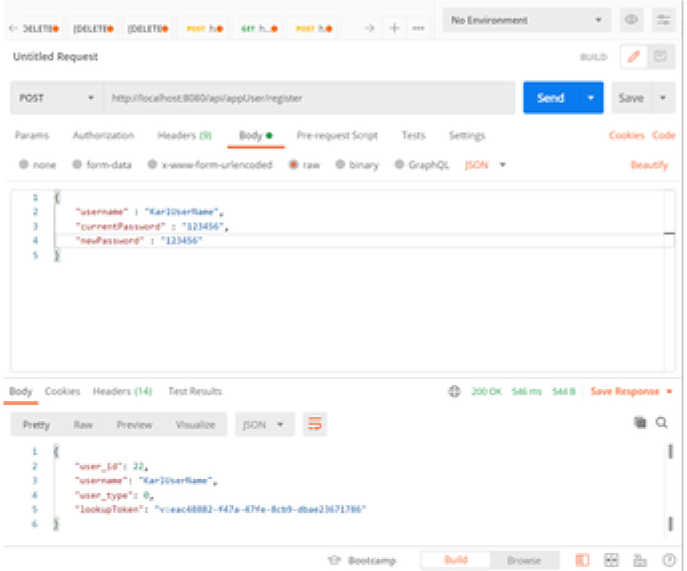
Link to earlier SpotBug report: https://github.com/KarlLeell/ASETeamJVM/tree/dev/wehealth/reports/SpotbugReportOld

Link to latest StyleCheck report: https://github.com/KarlLeell/ASETeamJVM/tree/dev/wehealth/reports/InspectionReportLatest

Link to earlier StyleCheck report: https://github.com/KarlLeell/ASETeamJVM/tree/dev/wehealth/reports/InspectionReportOld

Appendix A: Testing Results Snapshot

Register/Log in

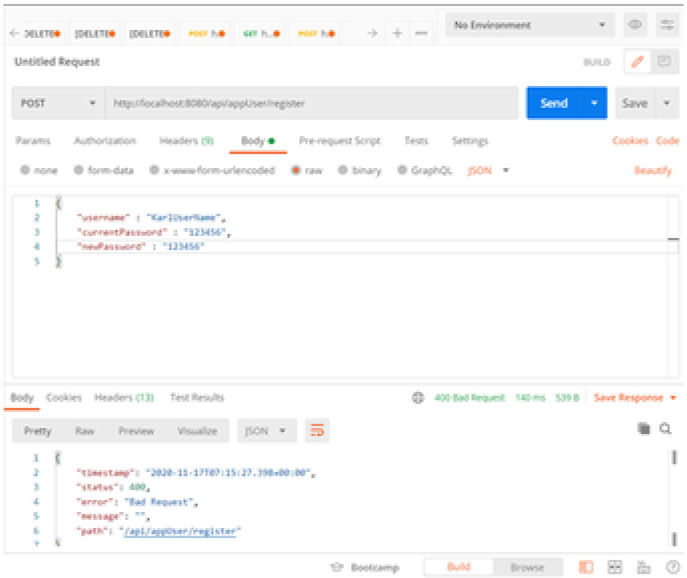


POST http://localhost:8080/api/appUser/register

```
{
    "username" : "KarlUserName",
    "currentPassword" : "123456",
    "newPassword" : "123456"
}
```

200 OK 546 ms 544 B

```
{
    "user_id": 22,
    "username": "KarlUserName",
    "user_type": 0,
    "lookupToken": "v:eac48882-f47a-47fe-8cb9-dbae23671786"
}
```

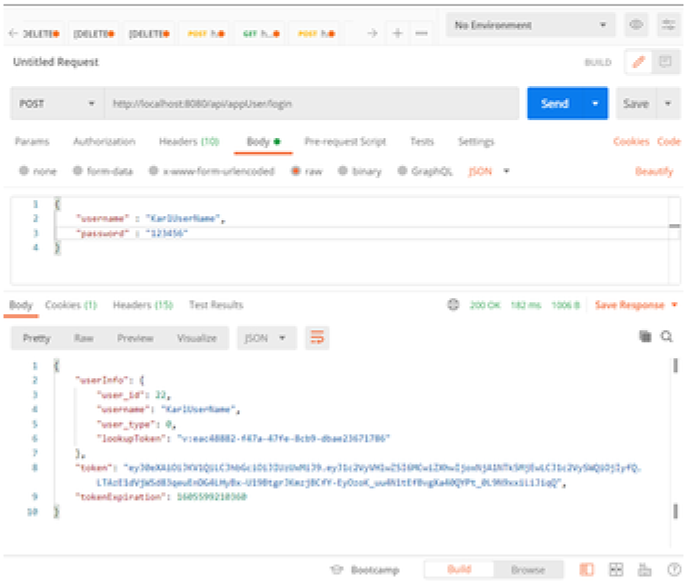


POST http://localhost:8080/api/appUser/register

```
{
    "username" : "Karl1UserName",
    "currentPassword" : "123456",
    "newPassword" : "123456"
}
```

400 Bad Request 140 ms 539 B

```
{
    "timestamp": "2020-11-17T07:15:27.398+00:00",
    "status": 400,
    "error": "Bad Request",
    "message": "",
    "path": "/api/appUser/register"
}
```

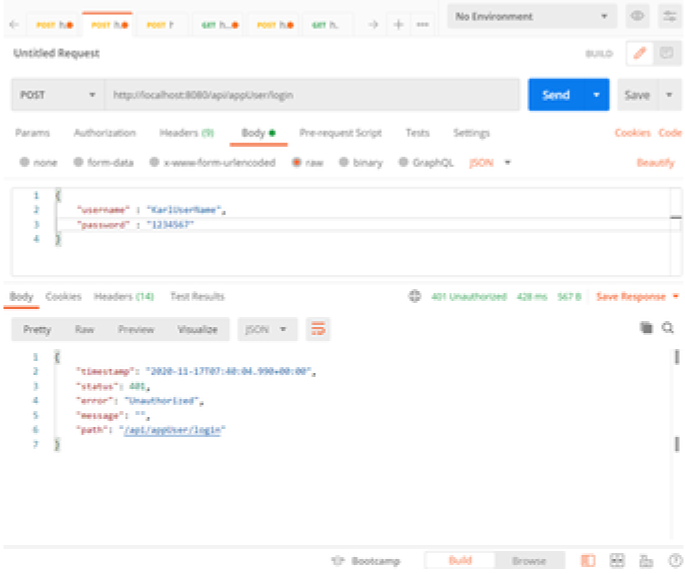


POST http://localhost:8080/api/appUser/login

```
{
    "username" : "Karl1UserName",
    "password" : "123456"
}
```

200 OK 182 ms 1006 B

```
{
    "userInfo": {
        "user_id": 22,
        "username": "Karl1UserName",
        "user_type": 0,
        "lookupToken": "v:eac48882-f47a-47fe-8cb9-dbae23671786"
    },
    "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzUxMiJ9.eyJ1c2VyVHlwZSI6MCwiZXhwIjoxNjA1NTk5MjEwLCJ1c2VyNWQ0OjIyfQ.
        LTAcE1dVjW5dB3qeuEnDG4LHyBx-U19BtgrJKmzjBCfY-EyOzoK_uu4N1tEfBvgKa40QYPt_0L9R9xxiiJ1qQ",
    "tokenExpiration": 1605599210360
}
```



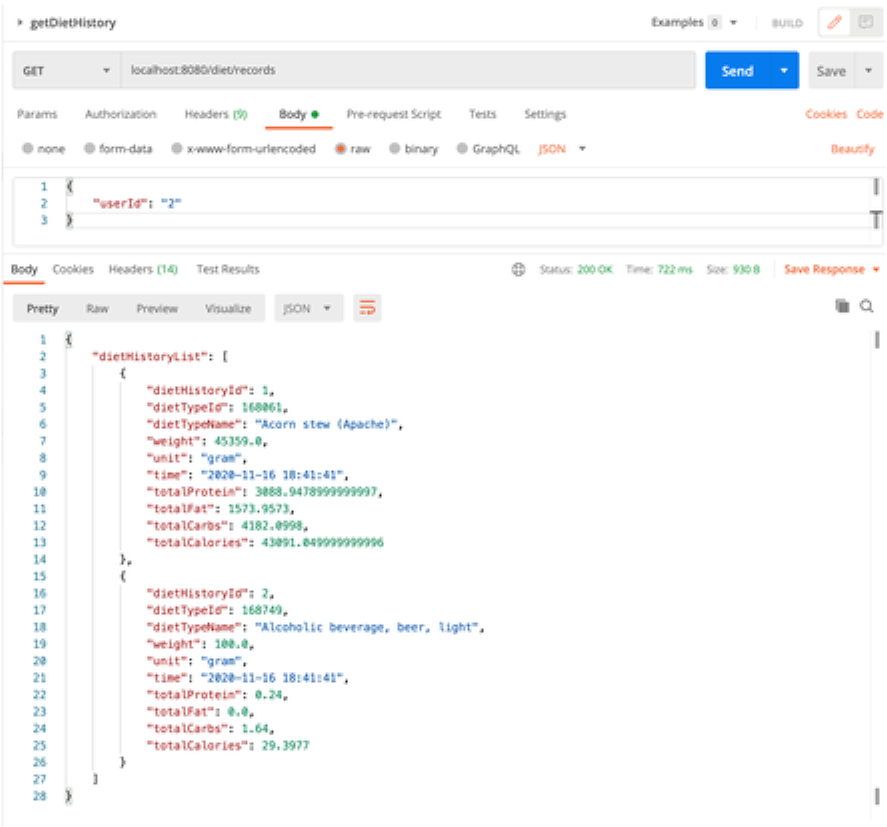
POST http://localhost:8080/api/appUser/login

```
{
    "username" : "Karl1UserName",
    "password" : "1234567"
}
```

401 Unauthorized 428 ms 567 B

```
{
    "timestamp": "2020-11-17T07:40:04.990+00:00",
    "status": 401,
    "error": "Unauthorized",
    "message": "",
    "path": "/api/appUser/login"
}
```

Diet

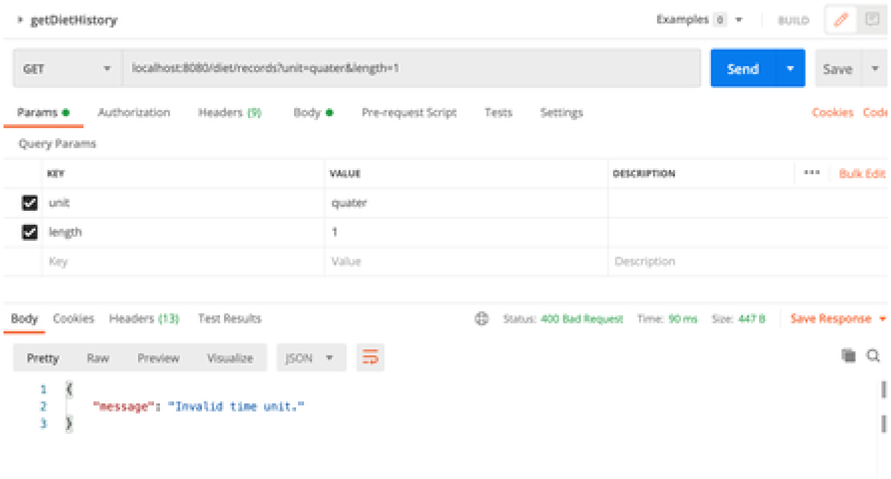


GET localhost:8080/diet/records

```
{
    "userId": "2"
}
```

Status: 200 OK Time: 722 ms Size: 930 B

```
{
    "dietHistoryList": [
        {
            "dietHistoryId": 1,
            "dietTypeId": 168061,
            "dietTypeName": "Acorn stew (Apache)",
            "weight": 45359.0,
            "unit": "gram",
            "time": "2020-11-16 18:41:41",
            "totalProtein": 3088.9478999999997,
            "totalFat": 1573.9573,
            "totalCarbs": 4182.0998,
            "totalCalories": 43091.049999999996
        },
        {
            "dietHistoryId": 2,
            "dietTypeId": 168749,
            "dietTypeName": "Alcoholic beverage, beer, light",
            "weight": 100.0,
            "unit": "gram",
            "time": "2020-11-16 18:41:41",
            "totalProtein": 0.24,
            "totalFat": 0.0,
            "totalCarbs": 1.64,
            "totalCalories": 29.3977
        }
    ]
}
```



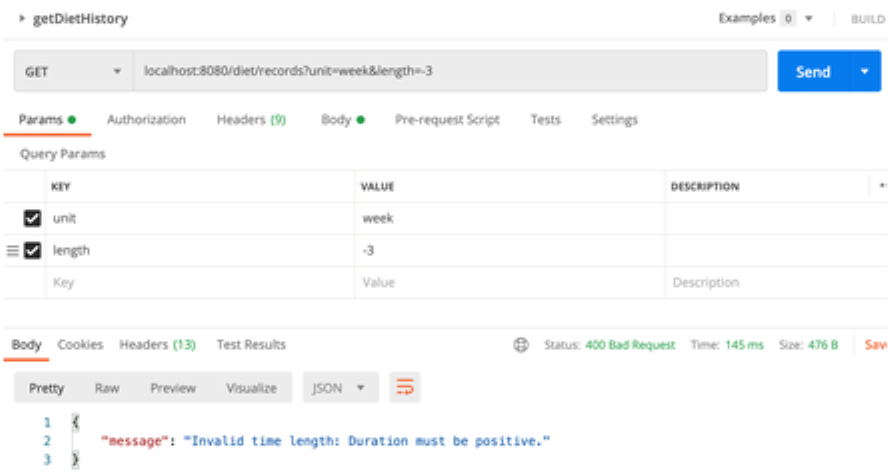
GET localhost:8080/diet/records?unit=quater&length=1

Query Params

	KEY	VALUE	DESCRIPTION
☑	unit	quater	
☑	length	1	
	Key	Value	Description

Status: 400 Bad Request Time: 90 ms Size: 447 B

```
{
    "message": "Invalid time unit."
}
```



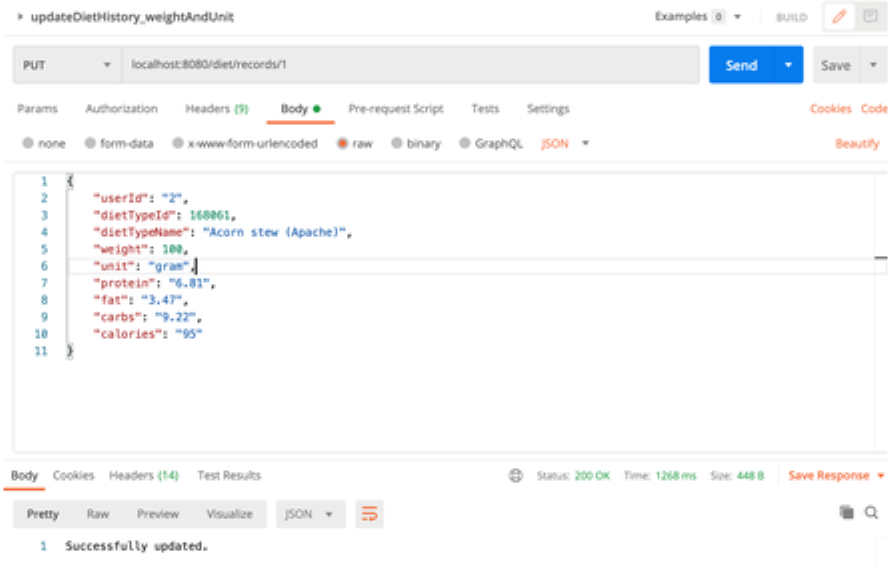
GET localhost:8080/diet/records?unit=week&length=-3

Query Params

	KEY	VALUE	DESCRIPTION
☑	unit	week	
☑	length	-3	
	Key	Value	Description

Status: 400 Bad Request Time: 145 ms Size: 476 B

```
{
    "message": "Invalid time length: Duration must be positive."
}
```



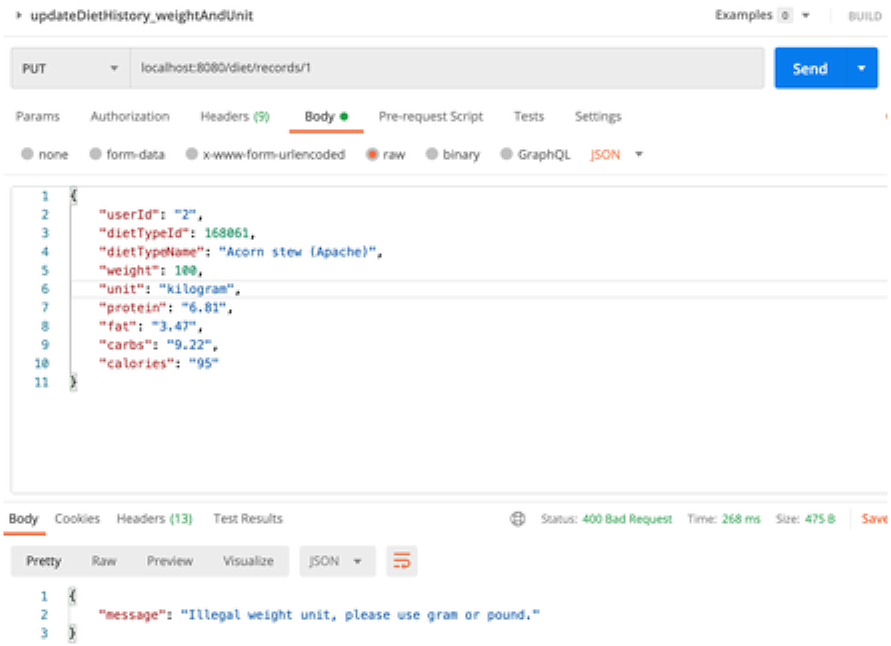
PUT localhost:8080/diet/records/1

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
{
    "userId": "2",
    "dietTypeId": 168061,
    "dietTypeName": "Acorn stew (Apache)",
    "weight": 100,
    "unit": "gram",
    "protein": "6.81",
    "fat": "3.47",
    "carbs": "9.22",
    "calories": "95"
}
```

Status: 200 OK Time: 1268 ms Size: 448 B

```
Successfully updated.
```



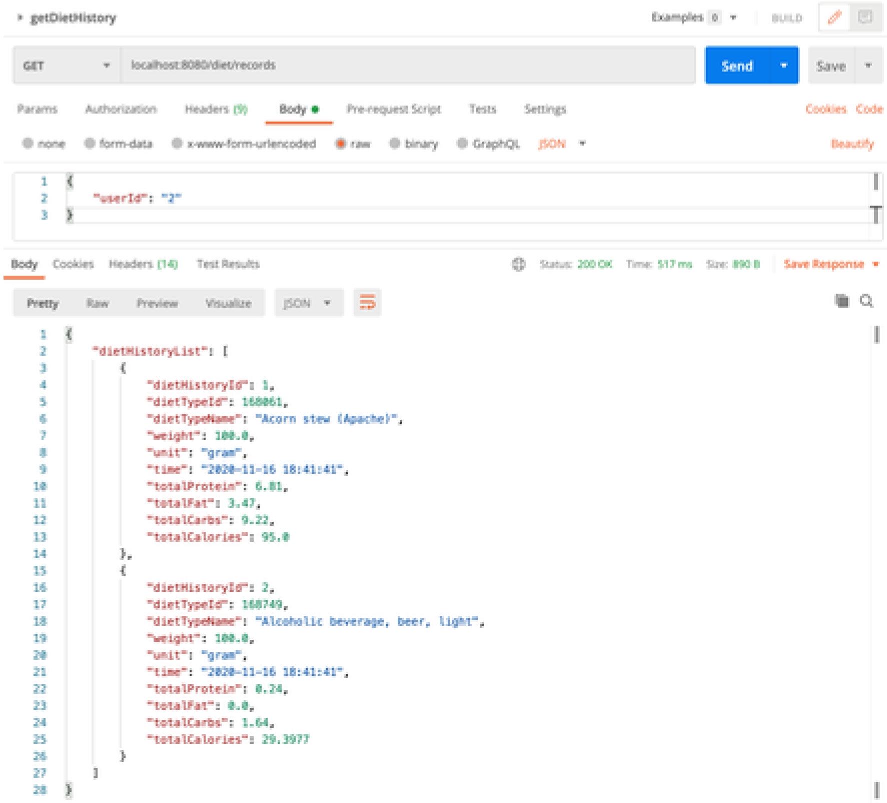
PUT localhost:8080/diet/records/1

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
{
    "userId": "2",
    "dietTypeId": 168061,
    "dietTypeName": "Acorn stew (Apache)",
    "weight": 100,
    "unit": "kilogram",
    "protein": "6.81",
    "fat": "3.47",
    "carbs": "9.22",
    "calories": "95"
}
```

Status: 400 Bad Request Time: 268 ms Size: 475 B

```
{
    "message": "Illegal weight unit, please use gram or pound."
}
```



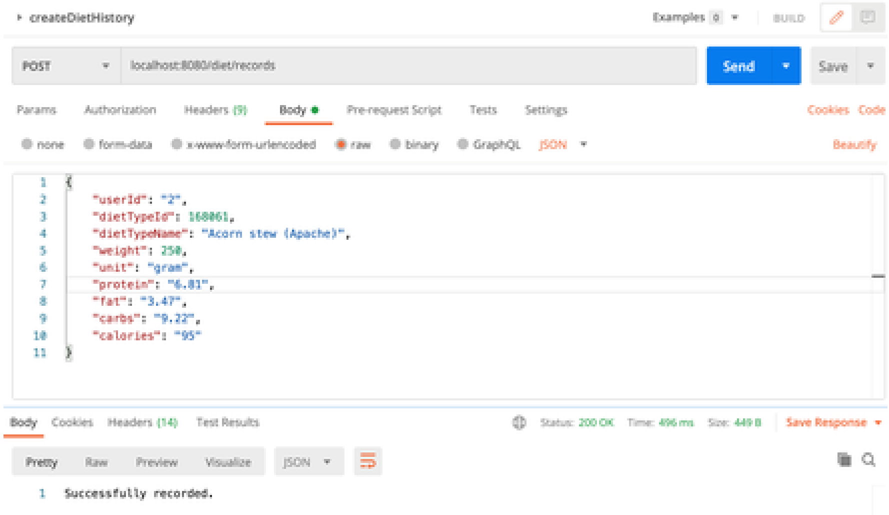
GET localhost:8080/diet/records

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
{
    "userId": "2"
}
```

Status: 200 OK Time: 517 ms Size: 890 B

```
{
    "dietHistoryList": [
        {
            "dietHistoryId": 1,
            "dietTypeId": 168061,
            "dietTypeName": "Acorn stew (Apache)",
            "weight": 100.0,
            "unit": "gram",
            "time": "2020-11-16 18:41:41",
            "totalProtein": 6.81,
            "totalFat": 3.47,
            "totalCarbs": 9.22,
            "totalCalories": 95.0
        },
        {
            "dietHistoryId": 2,
            "dietTypeId": 168749,
            "dietTypeName": "Alcoholic beverage, beer, light",
            "weight": 100.0,
            "unit": "gram",
            "time": "2020-11-16 18:41:41",
            "totalProtein": 0.24,
            "totalFat": 0.0,
            "totalCarbs": 1.64,
            "totalCalories": 29.3977
        }
    ]
}
```



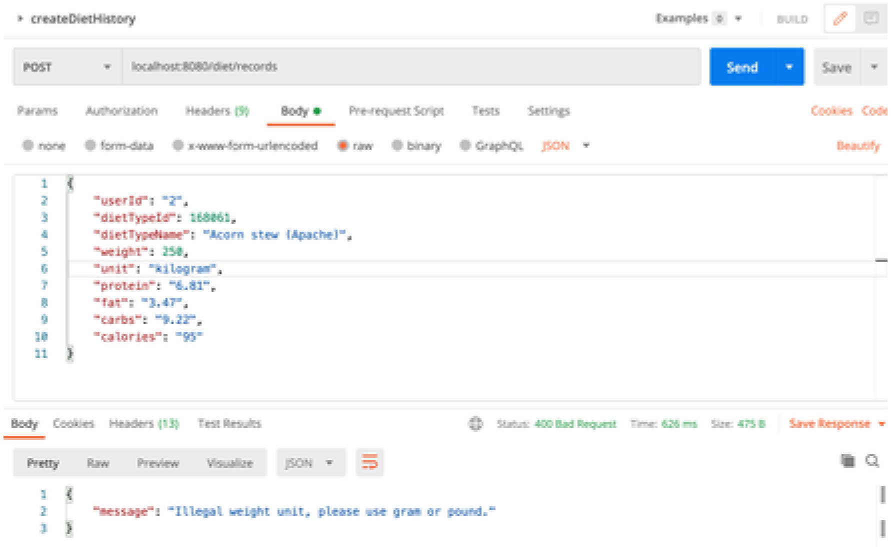
POST localhost:8080/diet/records

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
{
    "userId": "2",
    "dietTypeId": 168061,
    "dietTypeName": "Acorn stew (Apache)",
    "weight": 250,
    "unit": "gram",
    "protein": "6.81",
    "fat": "3.47",
    "carbs": "9.22",
    "calories": "95"
}
```

Status: 200 OK Time: 496 ms Size: 449 B

```
Successfully recorded.
```



createDietHistory

POST localhost:8080/diet/records

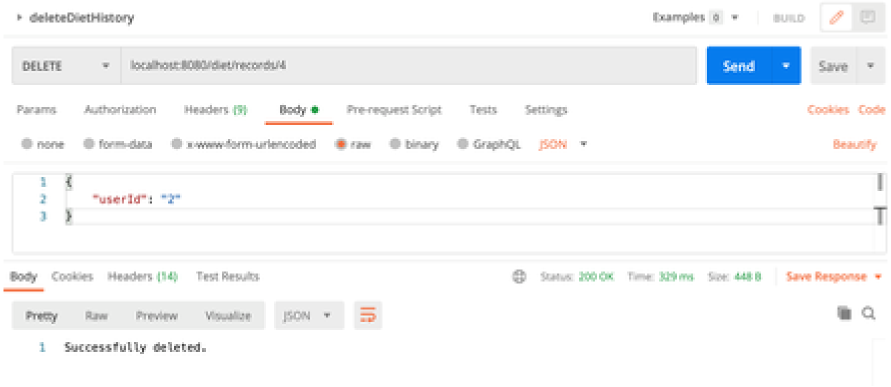
Body: ○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
{
    "userId": "2",
    "dietTypeId": 168061,
    "dietTypeName": "Acorn stew (Apache)",
    "weight": 250,
    "unit": "kilogram",
    "protein": "6.81",
    "fat": "3.47",
    "carbs": "9.22",
    "calories": "95"
}
```

Status: 400 Bad Request Time: 626 ms Size: 475 B

JSON

```
{
    "message": "Illegal weight unit, please use gram or pound."
}
```



deleteDietHistory

DELETE localhost:8080/diet/records/4

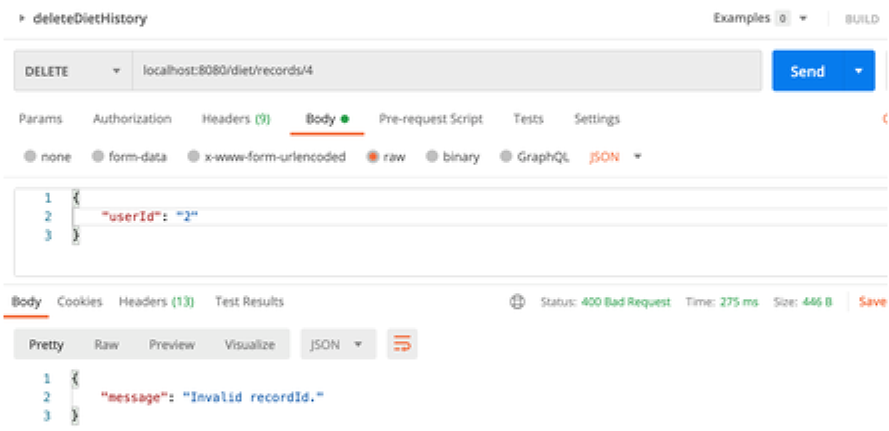
Body: ○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
{
    "userId": "2"
}
```

Status: 200 OK Time: 329 ms Size: 448 B

JSON

```
Successfully deleted.
```



deleteDietHistory

DELETE localhost:8080/diet/records/4

Body: ○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

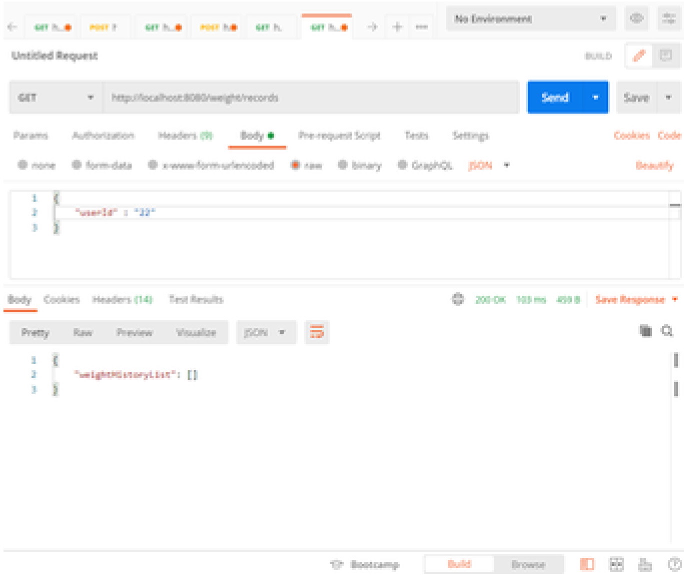
```
{
    "userId": "2"
}
```

Status: 400 Bad Request Time: 275 ms Size: 446 B

JSON

```
{
    "message": "Invalid recordId."
}
```

Weight



Untitled Request

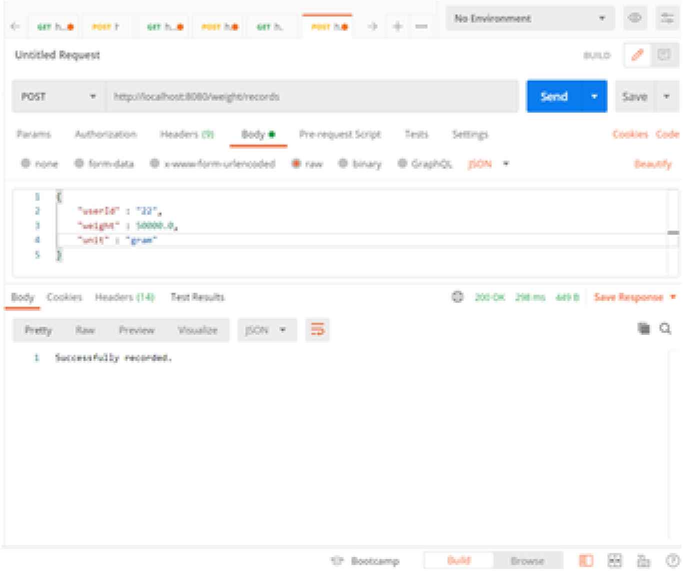
GET http://localhost:8080/weight/records

Body: raw JSON

```
{
    "userId" : "22"
}
```

Body 200 OK 103 ms 459 B

```
{
    "weightHistoryList": []
}
```



Untitled Request

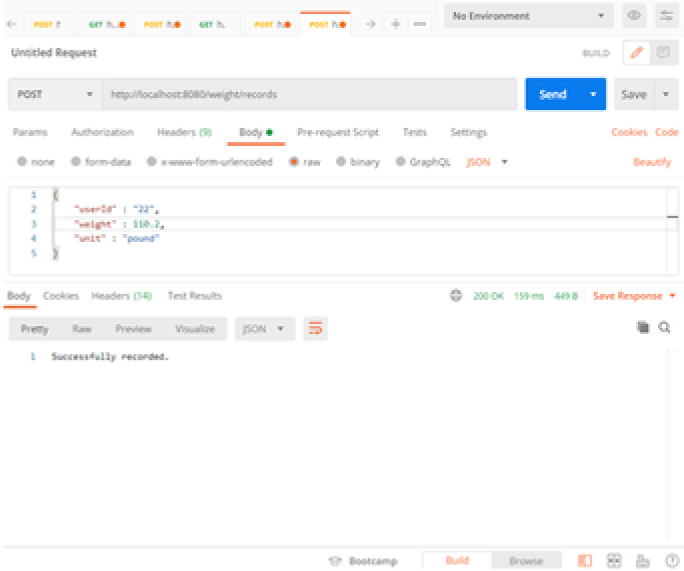
POST http://localhost:8080/weight/records

Body: raw JSON

```
{
    "userId" : "22",
    "weight" : 50000.0,
    "unit" : "gram"
}
```

Body 200 OK 298 ms 449 B

```
Successfully recorded.
```



Untitled Request

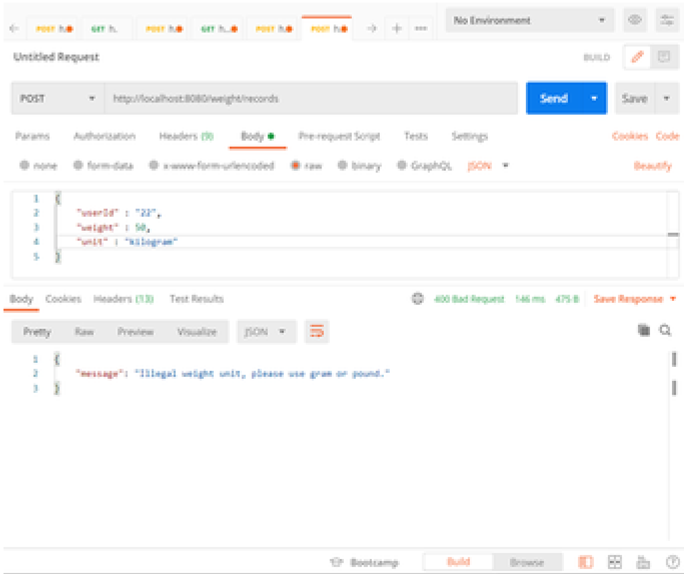
POST http://localhost:8080/weight/records

Body: raw JSON

```
{
    "userId" : "22",
    "weight" : 110.2,
    "unit" : "pound"
}
```

Body 200 OK 159 ms 449 B

```
Successfully recorded.
```

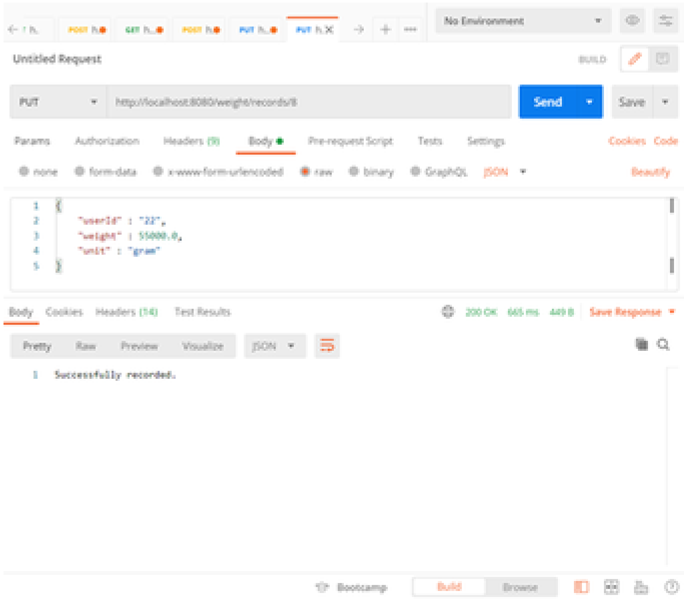


POST http://localhost:8080/weight/records

```
{
    "userId" : "22",
    "weight" : 50,
    "unit" : "kilogram"
}
```

400 Bad Request 146 ms 475 B

```
{
    "message": "Illegal weight unit, please use gram or pound."
}
```

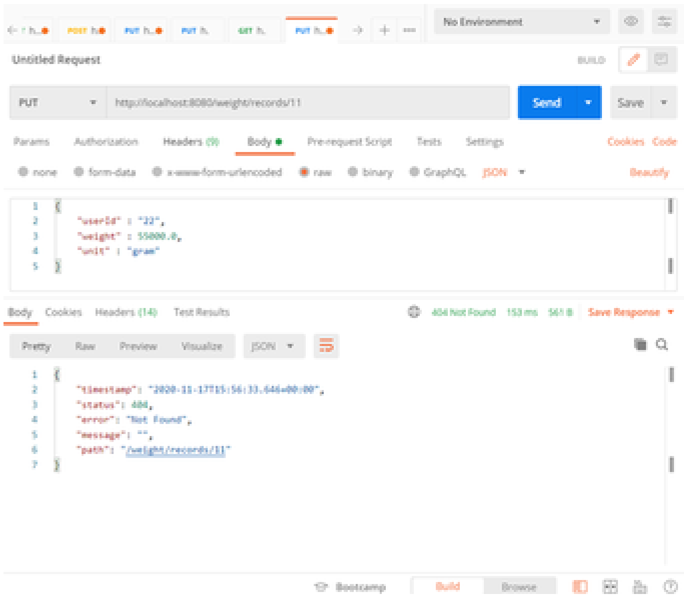


PUT http://localhost:8080/weight/records/8

```
{
    "userId" : "22",
    "weight" : 55000.0,
    "unit" : "gram"
}
```

200 OK 665 ms 449 B

```
Successfully recorded.
```

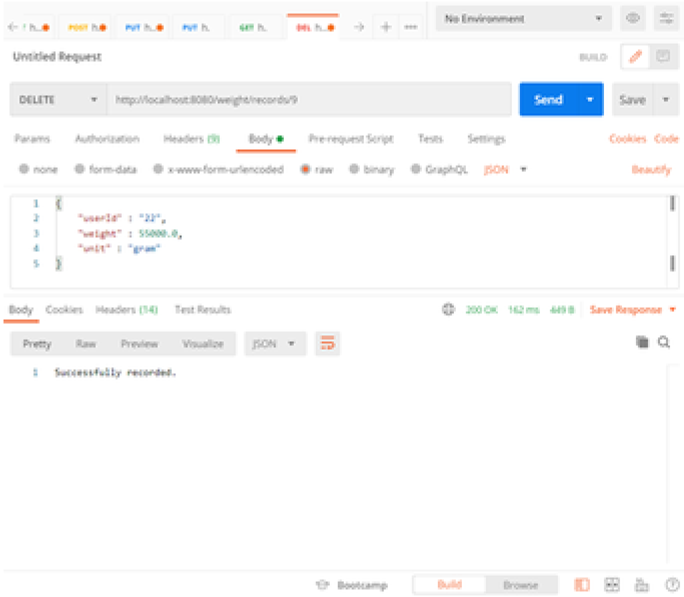


PUT http://localhost:8080/weight/records/11

```
{
    "userId" : "22",
    "weight" : 55000.0,
    "unit" : "gram"
}
```

404 Not Found 153 ms 561 B

```
{
    "timestamp": "2020-11-17T15:56:33.646+00:00",
    "status": 404,
    "error": "Not Found",
    "message": "",
    "path": "/weight/records/11"
}
```



DELETE http://localhost:8080/weight/records/9

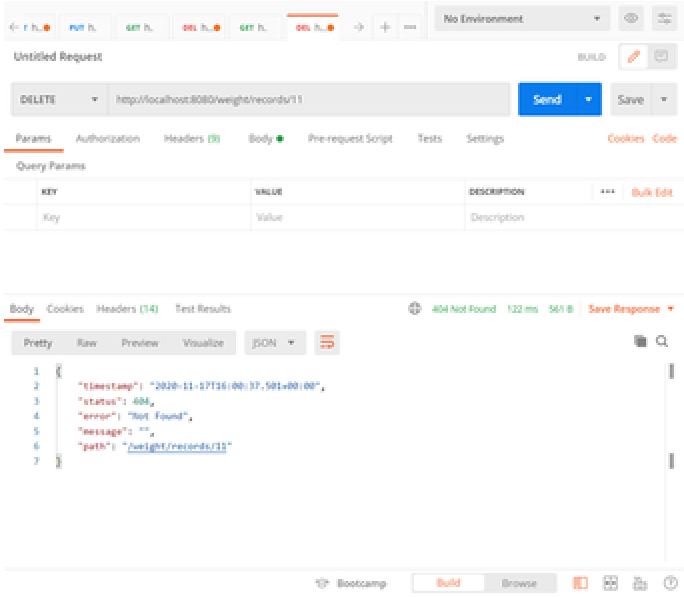
Body: raw JSON

```
{
    "userId" : "22",
    "weight" : 55000.0,
    "unit" : "gram"
}
```

200 OK 162 ms 449 B

JSON

```
Successfully recorded.
```



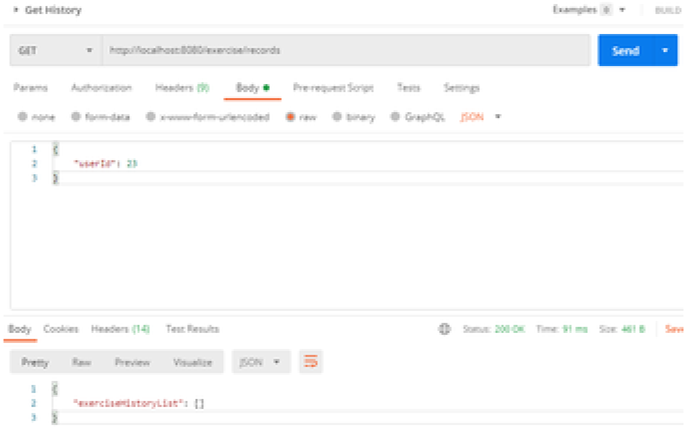
DELETE http://localhost:8080/weight/records/11

404 Not Found 122 ms 561 B

JSON

```
{
    "timestamp": "2020-11-17T16:00:37.501+00:00",
    "status": 404,
    "error": "Not Found",
    "message": "",
    "path": "/weight/records/11"
}
```

Exercise



GET http://localhost:8080/exercise/records

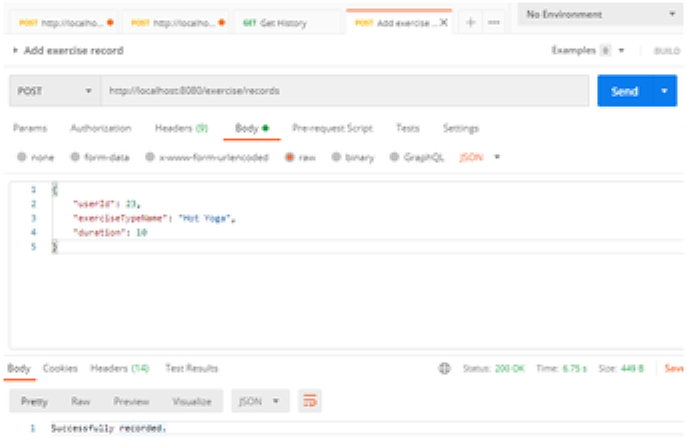
Body: raw JSON

```
{
    "userId": 23
}
```

Status: 200 OK Time: 91 ms Size: 461 B

JSON

```
{
    "exerciseHistoryList": []
}
```



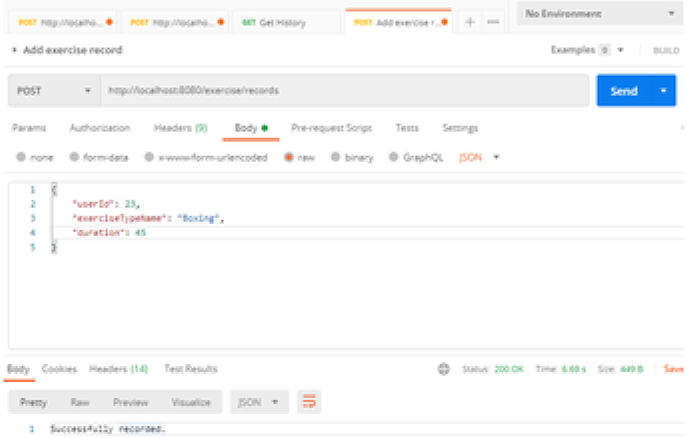
Add exercise record

POST http://localhost:8080/exercise/records

```
{
    "userId": 23,
    "exerciseTypeName": "Hot Yoga",
    "duration": 10
}
```

Status: 200 OK Time: 6.75 s Size: 449 B

```
Successfully recorded.
```



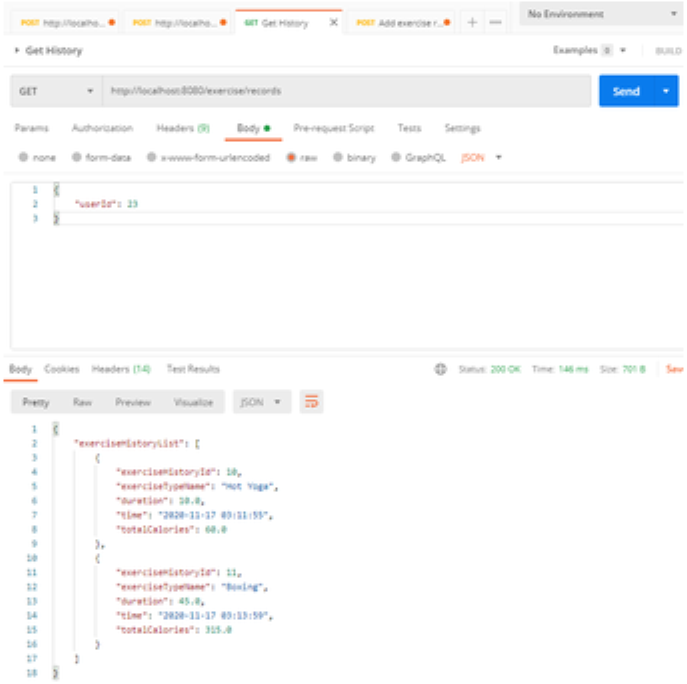
Add exercise record

POST http://localhost:8080/exercise/records

```
{
    "userId": 23,
    "exerciseTypeName": "Boxing",
    "duration": 45
}
```

Status: 200 OK Time: 6.69 s Size: 449 B

```
Successfully recorded.
```



Get History

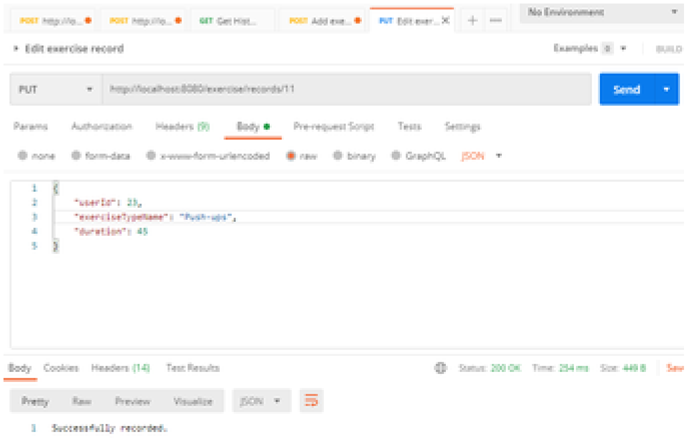
GET http://localhost:8080/exercise/records

Body: ● raw JSON

```
{
    "userId": 23
}
```

Status: 200 OK Time: 146 ms Size: 701 B

```
{
    "exerciseHistoryList": [
        {
            "exerciseHistoryId": 10,
            "exerciseTypeName": "Hot Yoga",
            "duration": 10.0,
            "time": "2020-11-17 03:11:55",
            "totalCalories": 60.0
        },
        {
            "exerciseHistoryId": 11,
            "exerciseTypeName": "Boxing",
            "duration": 45.0,
            "time": "2020-11-17 03:13:59",
            "totalCalories": 315.0
        }
    ]
}
```



Edit exercise record

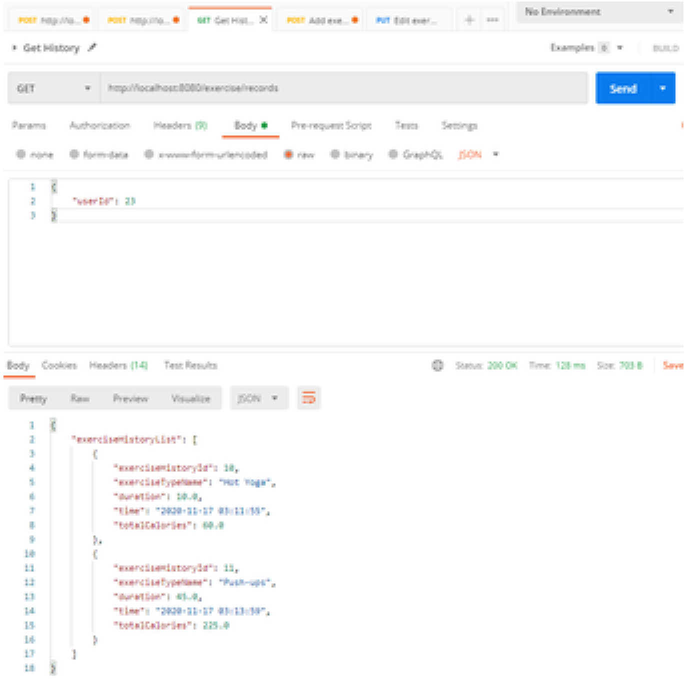
PUT http://localhost:8080/exercise/records/11

Body: ● raw JSON

```
{
    "userId": 23,
    "exerciseTypeName": "Push-ups",
    "duration": 45
}
```

Status: 200 OK Time: 254 ms Size: 449 B

```
Successfully recorded.
```

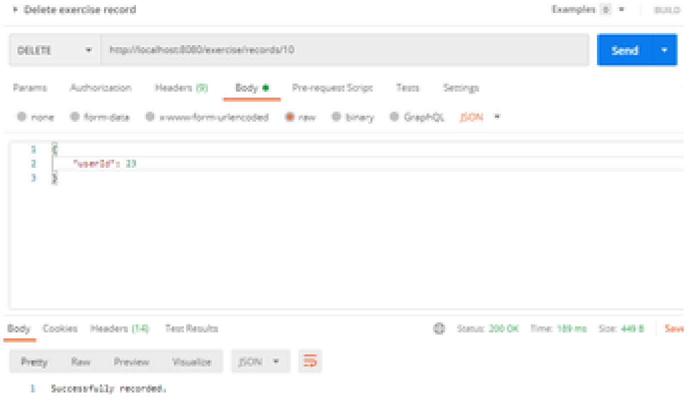


GET http://localhost:8080/exercise/records

```
{
    "userId": 23
}
```

Status: 200 OK Time: 128 ms Size: 703 B

```
{
    "exerciseHistoryList": [
        {
            "exerciseHistoryId": 10,
            "exerciseTypeName": "Hot Yoga",
            "duration": 10.0,
            "time": "2020-11-17 03:11:55",
            "totalCalories": 60.0
        },
        {
            "exerciseHistoryId": 11,
            "exerciseTypeName": "Push-ups",
            "duration": 45.0,
            "time": "2020-11-17 03:13:59",
            "totalCalories": 225.0
        }
    ]
}
```

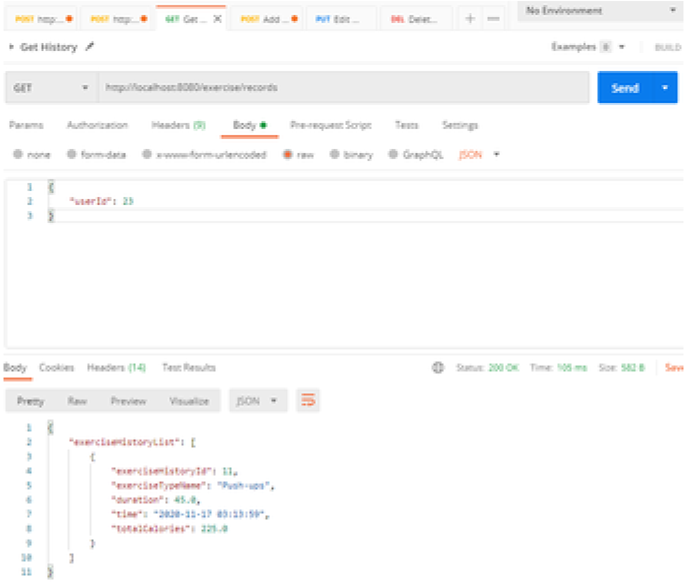


DELETE http://localhost:8080/exercise/records/10

```
{
    "userId": 23
}
```

Status: 200 OK Time: 189 ms Size: 449 B

```
Successfully recorded.
```

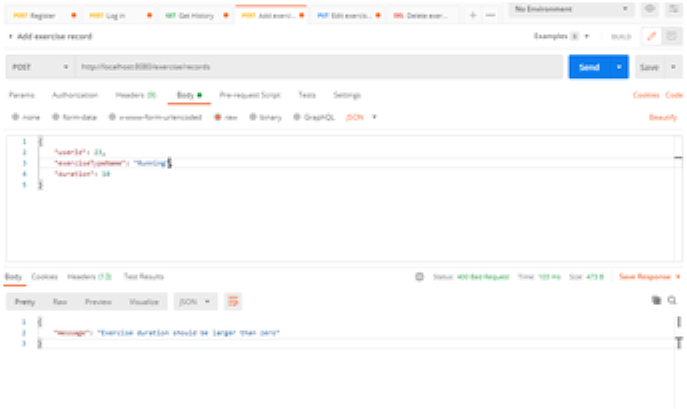


GET http://localhost:8080/exercise/records

```
{
    "userId": 23
}
```

Status: 200 OK Time: 105 ms Size: 582 B

```
{
    "exerciseHistoryList": [
        {
            "exerciseHistoryId": 11,
            "exerciseTypeName": "Push-ups",
            "duration": 45.0,
            "time": "2020-11-17 03:13:59",
            "totalCalories": 225.0
        }
    ]
}
```



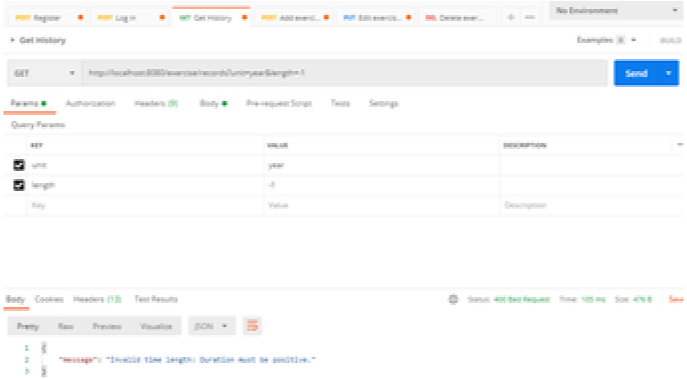
Add exercise record

POST http://localhost:8080/exercise/records

```
{
    "userId": 23,
    "exerciseTypeName": "Running",
    "duration": 10
}
```

Status: 400 Bad Request Time: 105 ms Size: 473 B

```
{
    "message": "Exercise duration should be larger than zero"
}
```



Get History

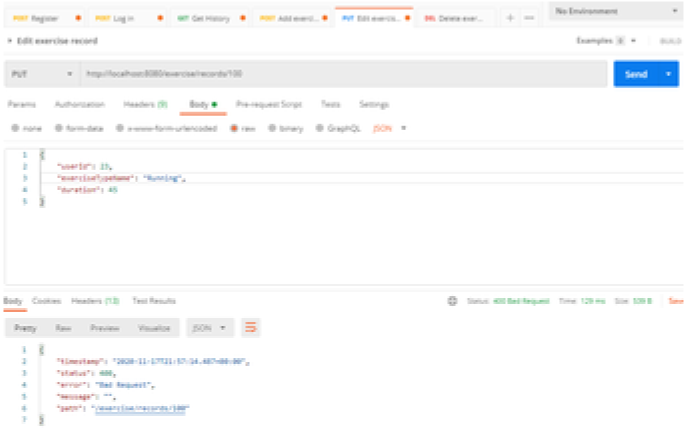
GET http://localhost:8080/exercise/records?unit=year&length=-1

Query Params

	KEY	VALUE	DESCRIPTION
☑	unit	year	
☑	length	-1	
	Key	Value	Description

Status: 400 Bad Request Time: 105 ms Size: 476 B

```
{
    "message": "Invalid time length: Duration must be positive."
}
```



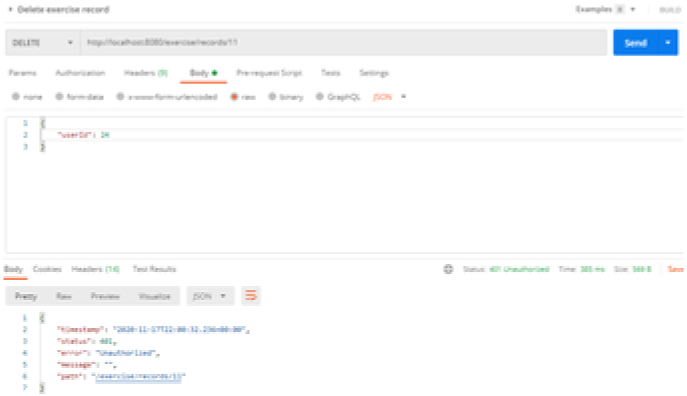
Edit exercise record

PUT http://localhost:8080/exercise/records/100

```
{
    "userId": 23,
    "exerciseTypeName": "Running",
    "duration": 45
}
```

Status: 400 Bad Request Time: 129 ms Size: 509 B

```
{
    "timestamp": "2020-11-17T21:57:24.487+00:00",
    "status": 400,
    "error": "Bad Request",
    "message": "",
    "path": "/exercise/records/100"
}
```



Delete exercise record

DELETE http://localhost:8080/exercise/records/11

```
{
    "userId": 24
}
```

Status: 401 Unauthorized Time: 385 ms Size: 569 B

```
{
    "timestamp": "2020-11-17T22:00:32.236+00:00",
    "status": 401,
    "error": "Unauthorized",
    "message": "",
    "path": "/exercise/records/11"
}
```