README - DFESW12 Final Project

Why are we doing this?

To create a Spring Boot API, with utilisation of supporting tools, methodologies, and technologies, that encapsulates all fundamental and practical modules covered during the QA Software Engineering Bootcamp.

More specifically, we were required to create a Spring Boot API using:

- an application back-end developed using the language from your Programming Fundamentals module (e.g. Java)
- a managed database hosted locally or within the Cloud Provider examined during your Cloud Fundamentals module (e.g. H2 or MySQL (local / GCP)
- a means of making API calls (Postman) and a means of checking persistence (Workbench/H2 console) How I expected the challenge to go.

How I Expected the Challenge to Go

I knew I had to recap on quite a few topics beforehand and covered this in my risk assessment in terms of risk of lack of understanding and potential design flaws. This meant looking at various external resources to ensure that I was ready to complete the project. However, I did expect to finish it within good time.

What went well? / What didn't go as planned?

What went well:

- Planning stage on Jira. I felt like I understood the agile approach quite well and this helped in setting me up early on in the project. It was a useful reference point throughout as well
- Risk assessment. I've done these before and find it quite natural to go through the likelihood and probability of potential impacts, elaborating on mitigation strategies where appropriate
- Postman requests and linking with SQL. There didn't seem to be any issues with this.

What didn't go so well:

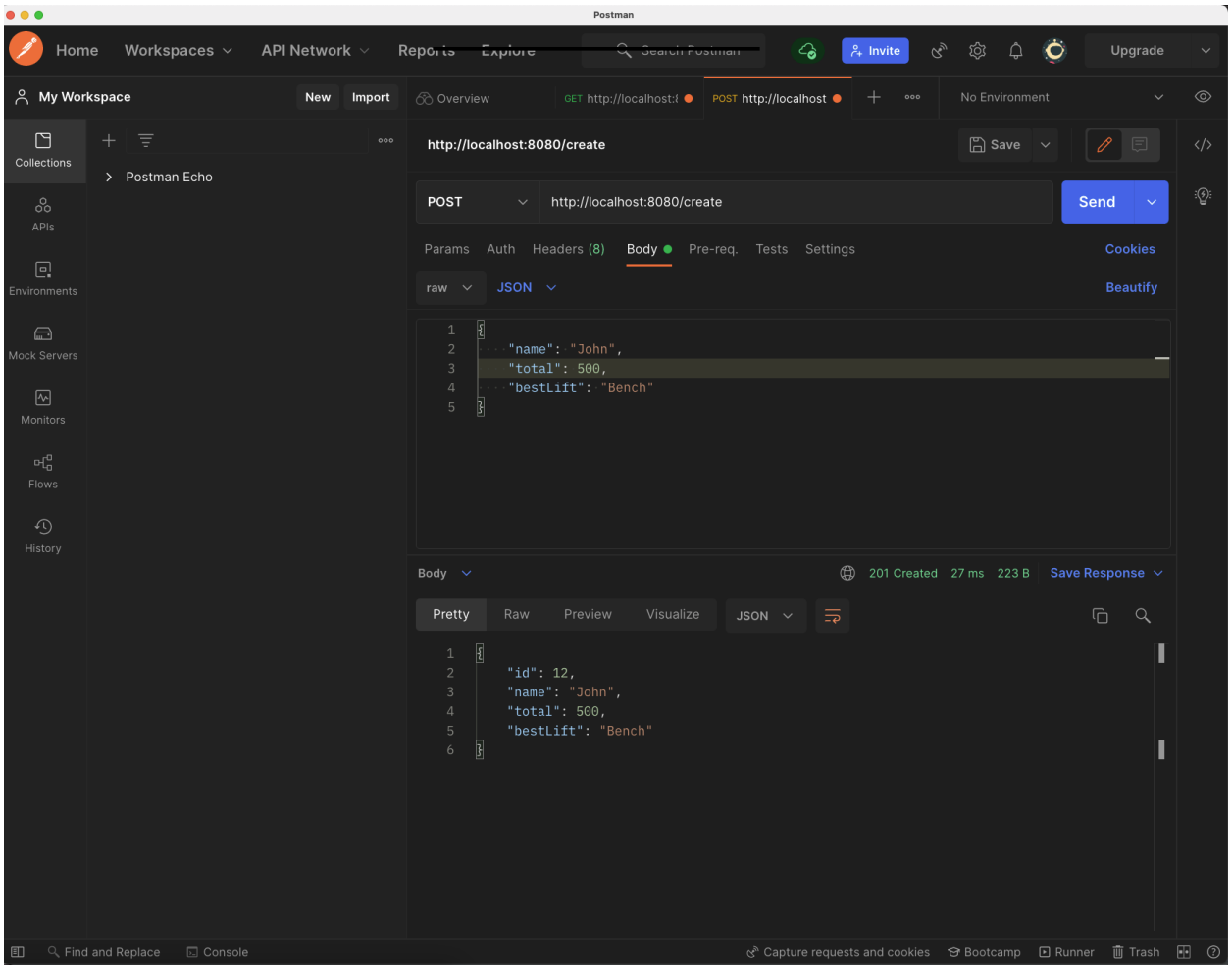
- The resources given to us throughout the bootcamp were not very good so it was a bit of a learning curve utilising other resources to ensure my understanding was at an appropriate level to complete the project
- My GitHub merge requests could have been a lot neater. I looked at the network diagram summarizing this and it doesn't look like best practice. However, I feel like my understanding is better now in regards to local and remote branches, along with merging them where appropriate in line with the feature branch model

Possible improvements for future revisions of the project.

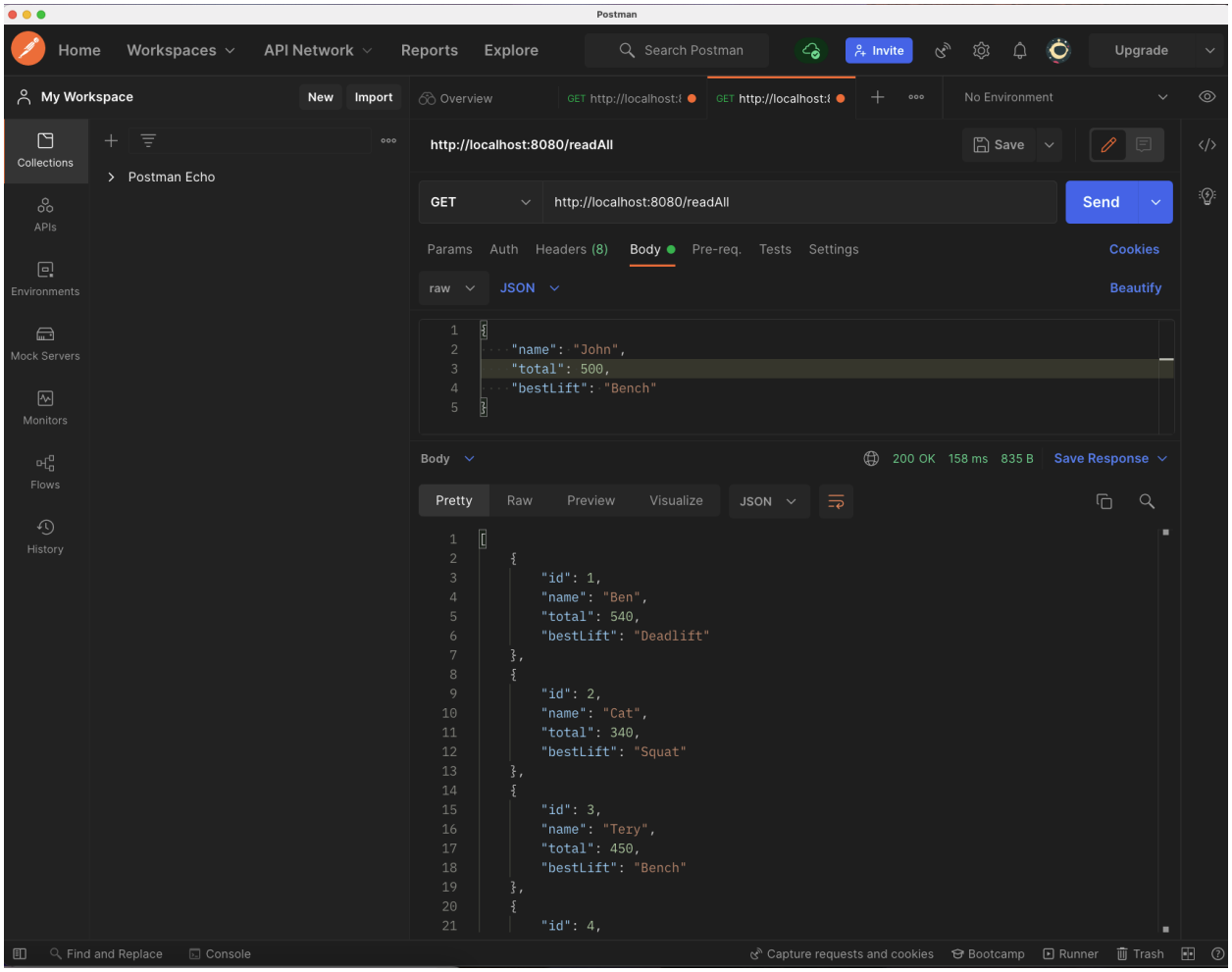
- Now I know how to use GitHub much better so I should be able to incorporate the feature branch model more effectively in the future.
- Learn more about testing
- The risk assessment could be more extensive
- I could potentially try and incorporate some of the stretch goals for the project

Screenshots showing your postman requests and the output from the API

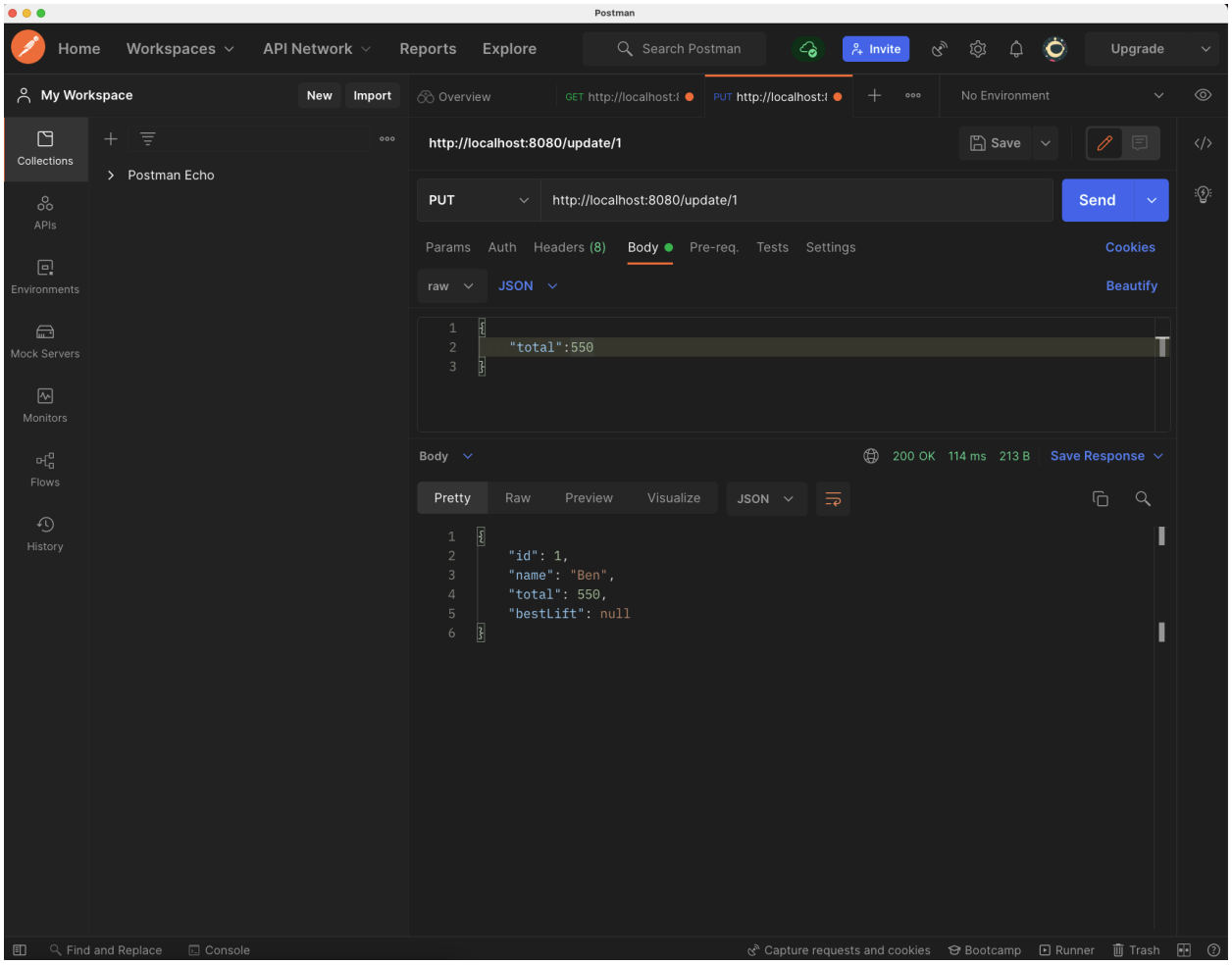
Create:



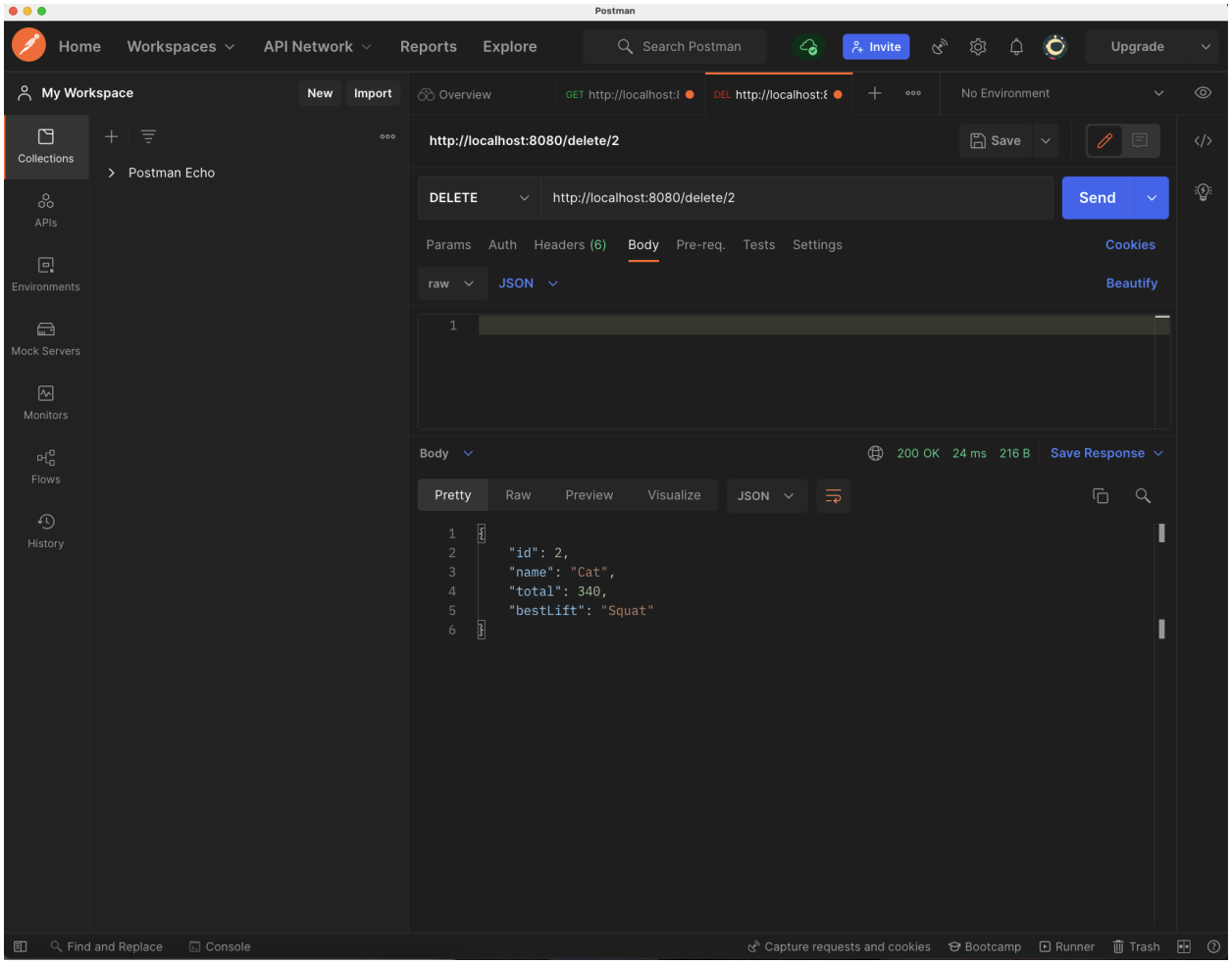
Read:


Postman
Home
Workspaces
API Network
Reports
Explore
Search Postman
Invite
Upgrade
My Workspace
New
Import
Overview
GET http://localhost:8
GET http://localhost:8
No Environment
Collections
APIs
Environments
Mock Servers
Monitors
Flows
History
Postman Echo
http://localhost:8080/readAll
Save
GET
http://localhost:8080/readAll
Send
Params
Auth
Headers (8)
Body
Pre-req.
Tests
Settings
Cookies
raw
JSON
Beautify
{
"name": "John",
"total": 500,
"bestLift": "Bench"
}
Body
200 OK
158 ms
835 B
Save Response
Pretty
Raw
Preview
Visualize
JSON
[
{
"id": 1,
"name": "Ben",
"total": 540,
"bestLift": "Deadlift"
},
{
"id": 2,
"name": "Cat",
"total": 340,
"bestLift": "Squat"
},
{
"id": 3,
"name": "Tery",
"total": 450,
"bestLift": "Bench"
},
{
"id": 4,
Find and Replace
Console
Capture requests and cookies
Bootcamp
Runner
Trash

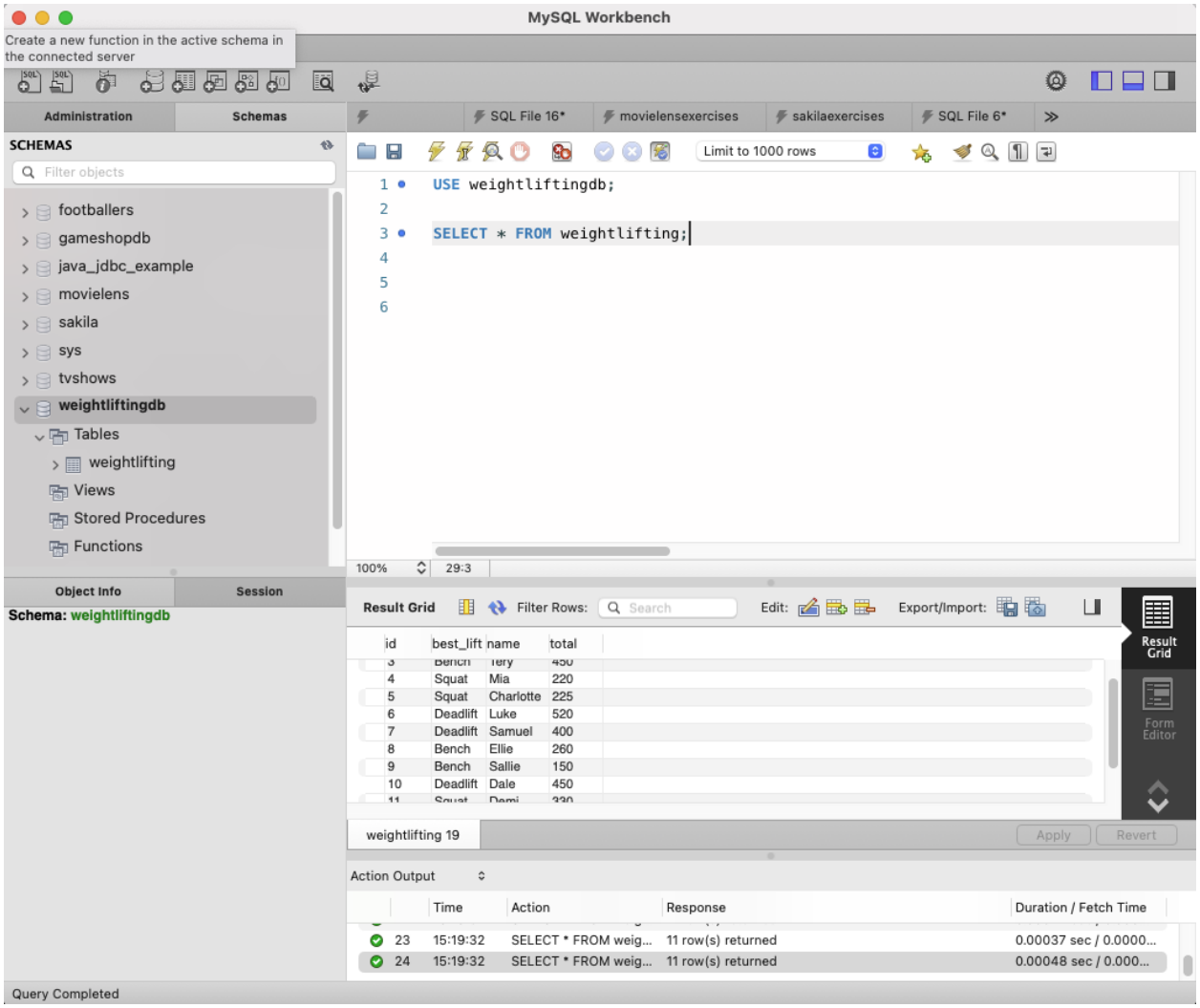
Update:


Postman
Home
Workspaces
API Network
Reports
Explore
Search Postman
Invite
Upgrade
My Workspace
New
Import
Overview
GET http://localhost:
PUT http://localhost:
No Environment
Collections
APIs
Environments
Mock Servers
Monitors
Flows
History
Postman Echo
http://localhost:8080/update/1
Save
PUT
http://localhost:8080/update/1
Send
Params
Auth
Headers (8)
Body
Pre-req.
Tests
Settings
Cookies
raw
JSON
Beautify
1 {
2 "total":550
3 }
Body
200 OK
114 ms
213 B
Save Response
Pretty
Raw
Preview
Visualize
JSON
1 {
2 "id": 1,
3 "name": "Ben",
4 "total": 550,
5 "bestLift": null
6 }
Find and Replace
Console
Capture requests and cookies
Bootcamp
Runner
Trash

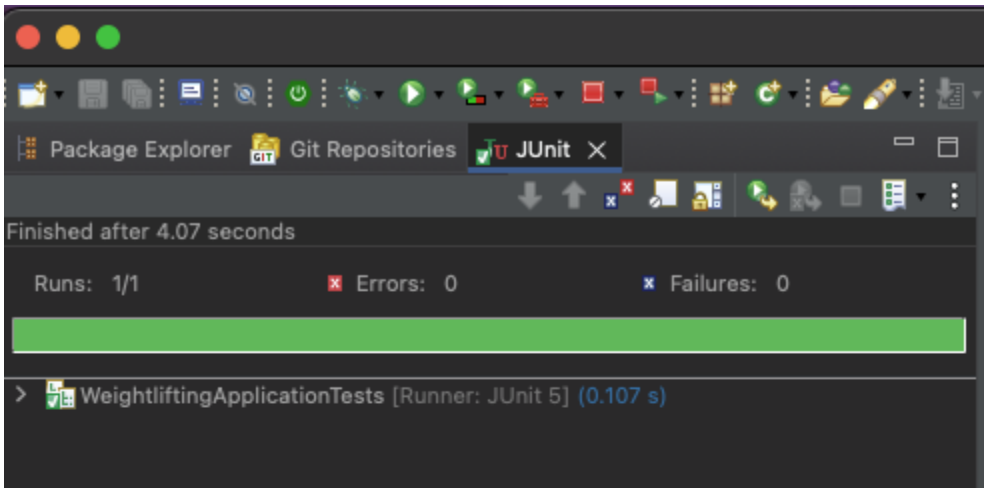
Delete:


Postman
Home
Workspaces
API Network
Reports
Explore
Search Postman
Invite
Upgrade
My Workspace
New
Import
Overview
GET http://localhost:
DEL http://localhost:
No Environment
Collections
APIs
Environments
Mock Servers
Monitors
Flows
History
Postman Echo
http://localhost:8080/delete/2
Save
DELETE
http://localhost:8080/delete/2
Send
Params
Auth
Headers (6)
Body
Pre-req.
Tests
Settings
Cookies
raw
JSON
Beautify
1
Body
200 OK
24 ms
216 B
Save Response
Pretty
Raw
Preview
Visualize
JSON
1 {
2 "id": 2,
3 "name": "Cat",
4 "total": 340,
5 "bestLift": "Squat"
6 }
Find and Replace
Console
Capture requests and cookies
Bootcamp
Runner
Trash

Screenshots of your database to prove that data is being persisted



Screenshot of your test results



Link to Jira Board

https://benattwell.atlassian.net/jira/software/projects/DFEFINAL/boards/3