**TASK**: Build a web application utilizing Java and a front-end language of your choice. We are open to how you want to approach this. However, the primary interest is demonstrating an application that can have persistent state of some sort and allow a change in data. Demonstrating a front-end + back-end that communicates via a REST API is the primary task on which you are being evaluated.

You **must** demonstrate proper test cases for your code. We are not evaluating based on code-coverage, but whether you understand proper testing strategy and usage. Examples of both unit and integration testing are strongly recommended. Practicals that are submitted without test coverage will be discarded and not considered.

If your practical requires special set-up to run, please provide documentation. If we cannot run your practical, we will discard it and end the interview process.

**REQUIREMENTS:**

1.) Create a repo at the version control software of your choice (typically github) and invite ahoneycutt@accendero.com and lbbateman@accendero.com so your code can be reviewed.

2.) Please push code through the process rather than one large commit at the end. We want to see your process as you develop.

**3.) PLEASE COMPLETE YOUR PRACTICAL NO LATER THAN WEDNESDAY, APRIL 21ST BY 8AM MST. PRACTICALS RECEIVED AFTER THIS POINT MAY NOT BE REVIEWED OR CONSIDERED.**

4.) Please provide a list of urls to all documentation that you used in the repository so we can review what secondary materials you consulted. Using sites like stack overflow is allowed and even encouraged, but we need to see how you apply other people's solutions in your own problem solving.

Failed to commit to GIT because of connectivity error with intelli Je (Error in terms of GIT authentication token and intellije version )

Urls Used

1. <https://docs.spring.io/spring-boot/docs/1.5.13.RELEASE/reference/html/boot-features-testing.html> tfor assertj in testing
2. Stackoverflow links
3. Assert official documentation
4. Springboot -- <https://spring.io/guides/tutorials/rest/>

Steps

1.Install IntelliJ , Postgre, Postman

2.Import project into intelli JE(demoAfterTest.Zip)

Import the database and for Postgre use the default port.

( open pgAdmin4

Create new database with name drugdb

Click on it and select restore select path and be careful and check format as sql)

)

3.Run the Program

(be sure to check the application property files for port number,

Also enter the Postgre database password, username you mentioned while installing it

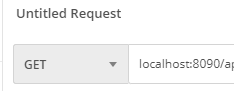
(The user is “Postgre ” and password is “password”

Open browser/postman

1.GET

localhost:8090/api/v1/drug

In postman



2.ADD POST

have to create mapping

Select PUT

3.delete(6 is id)

localhost:8090/api/v1/drug/6

Select the ID you want to delete

4.update(id and quantity)

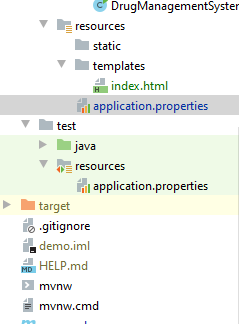
localhost:8090/api/v1/drug/1?quantity=6000

id=1

quantity =

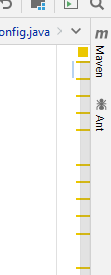
Please use the above format

Packaging and running.

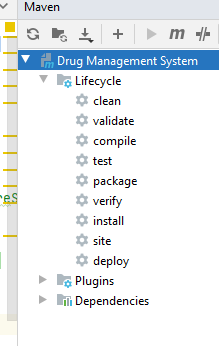


1. the target folder

2.Open Maven tab on the top right of your IDE



3.Clean the file by pressing on the

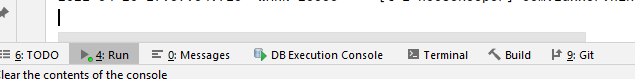


4.Select install and proceed

Now we see that the target folder is back

Expand it and we see a jar file will the name demo.0.0….SNAPSHOT,jar

Open terminal



This can be found on the bottom left of the screen

Write cd target to move to the target directory

* Java -jar demo.0.0….SNAPSHOT,jar

Press enter

Now we can run the application on different port but we have to mention the it .

* Java -jar demo.0.0….SNAPSHOT,jar --server.port=8090

We can run the above application using the above links from Postman