VSCode Lab Book

Table of Contents

Trying out the Environments

1. Open either of your favorite projects:

    a. Maven/Java Training Repository link

    b. Gradle/Java Training Repository link

    c. Node/Javascript Training Repository link

    d. Node/Python Training Repository link

2. Look around and explore

3. Open either on your machine or on gitpod or codespaces, or both!

4. Try some of the basic commands that you are used to in the terminal

5. Let's weigh the pros and cons

Adding Your Themes

1. Add and Choose your Theme

2. Discuss which ones you like

Change Some Settings

1. Disable the MiniMap in VSCode for either your workspace, user settings or both

2. Change the font size of your current workspace

3. View the list of settings and call them up quickly in JSON or using GUI in VSCode

4. Restore the MiniMap is you want it back.

Review your Keybindings

1. View the keybindings in your VSCode

2. Change or manipulate any that you deem important

3. Determine whether you want those as either User or Workspace

Using Bazel in VSCode

1. Open the Bazel Training Repository link

2. Look around

3. Open either on your machine or on gitpod or codespaces

4. Run some of the commands as seen in the *README.md* page:

```
$ bazel query "//src/main/java/com/xyzcorp:*"
$ bazel build "//src/main/java/com/xyzcorp:vs_code_bazel_runner"
$ bazel query "//src/test/java/com/xyzcorp:*"
$ bazel test "//src/test/java/com/xyzcorp:vs_code_bazel_tests"
```

5. We will also look at The Bazel Ls Demo project and show how we can turn on and off modules

Refactoring

1. Use a template from Ted Young's Blackjack Website

2. Open on Gitpod, Codespaces, or Local

3. Let's try some refactoring techniques!

Running Tests

1. In this lab we will try out Test Driven Development by adding a test for our `Wallet`

2. If there isn't a Wallet, create one

3. We we create a balance method that will return the balance of the wallet

4. View the reports if successful

Opening Pet Clinic in VSCode

1. Go to the PetClinic [Repository](#)

2. Click on the Gitpod or Codespaces Link

3. Choose Maven for the Build Tool

4. Install the Spring Boot Extension Pack

5. Navigate the project

6. Hit `F5` to run the project and open in the browser

Opening Python & Jupyter Notebooks in VSCode

1. Open the [Python Training Repository](#) link

2. Look around

3. Open in gitpod.io or on your local machine. If you are using gitpod, the Jupyter extension should automatically be downloaded. If you are running this on you local machine, download that extension

4. Let's run the Jupyter Notebook for some examples, and review the python code.

5. Be sure partronize "Head First Python, 3rd Edition"

Run an HTTP Editing

1. Install the Extension, REST Client Extension by Huachao Mao

2. In any of the java projects, create a file in *src/main/resources* called http_example.http

3. Run the following content:

```
GET https://postman-echo.com/get
Content-Type: application/json
```

4. Try some other commands. Use the reference from their [website](#)

5. Note, that you can use `#` to separate HTTP calls in the same document

6. Note, that payloads must have a carriage return after the headers

Docker Compose

1. Open [nfjs_kafka-half-day](#) in Gitpod.io

2. A docker-compose up will already be running

3. Run a docker compose up for container `web` only

4. Open and expose the ports for 9091 and 8000

5. Let's follow the Producer example to produce information to kafka

SQL Tools

1. Install the SQL Tools Extension by Matheus Teixerira

2. Install the SQL Tools MySQL/MariaDB/TiDB Extension by Matheus Teixerira

3. Open a terminal and run: `docker run -p 3306:3306 -d sakiladb/mysql:5.6`

4. Open the SQL Tools Window

5. Click *Add Connection*

6. In *Connection Name* enter `sakila`

7. In *Database* enter `sakila`

8. In *Username* enter `sakila`

9. In *Password Mode* select, `Save as plaintext in settings`

10. In *Password* enter `p_ssW0rd`

11. Click Save Connection

12. Run the following: `SELECT * FROM actor LIMIT 50;` and view the data on the right

13. Create a file anywhere in the project. If in a Java project try the *src/main/resources* directory, call it *actors.sql*

14. Try adding the `SELECT * FROM actor LIMIT 50;` and Click *Run on active connection* and view the data on the right

15. Explore the Sakila Database and try some queries

Version Control

1. Let's setup teams where we collaborate on a project

2. Fork any of the projects we used: Maven, Gradle, or Bazel

3. Add more features to `Wallet`.

4. Practice performing commits, pushes, and pulls

5. Force and orchestrate a conflict