Getting Started on Spring Boot Project

This guide will help you get started with the following:

1. Setup MySQL Schema locally - All Team Members
2. Create Spring Boot project - Team Lead
3. Create initial GitHub repository - Team Lead
4. Connect Git to GitHub repository - All Team Members (Except Team Lead)
5. Pulling in Collaborator changes - All Team Members

Team Lead is responsible for: #1-3 and #5
Team Members are responsible for #1, #4 and #5

1. Setup MySQL Schema locally

Assuming you have already installed MySQL locally via Coding Dojo, please use the MySQL WorkBench to:

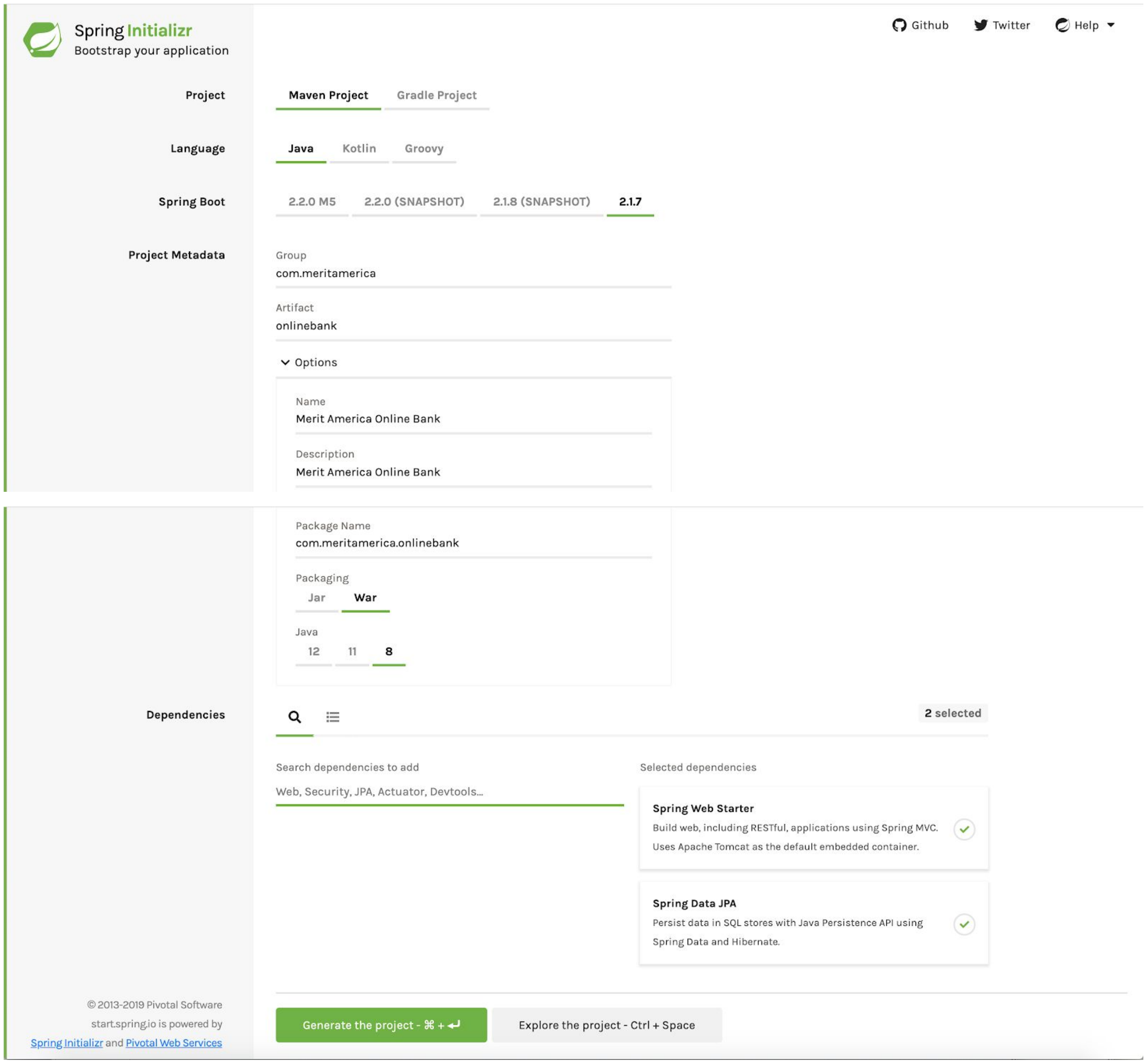
1. Create a new database schema called: onlinebank
2. Ensure you know the user credentials (e.g. user: root / password: root)

2. Creating Spring Boot project

As the team lead, you will be responsible for creating the initial Spring Boot project that will form the base source code that all team members will contribute to.

2.1 Creating Initial Spring Boot project

1. In your web browser, navigate to https://start.spring.io
2. Complete the wizard with the following values:


Github
Twitter
Help
Generate the project - ⌘ + ↵
Explore the project - Ctrl + Space

Spring Initializr
Bootstrap your application

Project: [✓ Maven Project] Gradle Project

Language: [✓ Java] Kotlin Groovy

Spring Boot: 2.2.0 M5 2.2.0 (SNAPSHOT) 2.1.8 (SNAPSHOT) [✓ 2.1.7]

Project Metadata

Group
com.meritamerica

Artifact
onlinebank

Options

Name
Merit America Online Bank

Description
Merit America Online Bank

Package Name
com.meritamerica.onlinebank

Packaging
Jar [✓ War]

Java
12 11 [✓ 8]

Dependencies

2 selected

Search dependencies to add
Web, Security, JPA, Actuator, Devtools...

Selected dependencies

Spring Web Starter
Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container. ✓

Spring Data JPA
Persist data in SQL stores with Java Persistence API using Spring Data and Hibernate. ✓

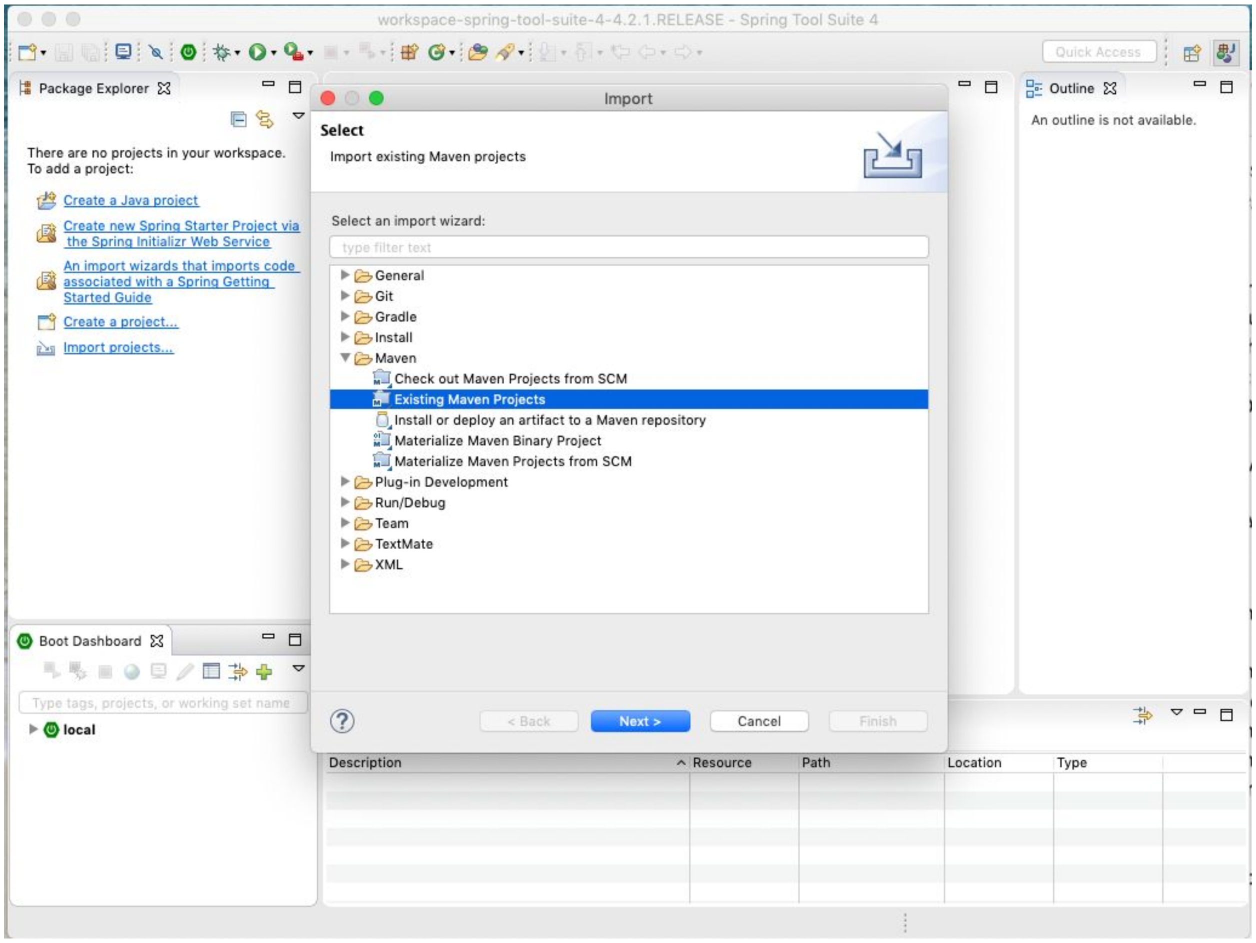
© 2013-2019 Pivotal Software
start.spring.io is powered by
Spring Initializr and Pivotal Web Services

3. Click the “Generate the project” button: this will download a zip file
4. Locate the zip file and extract it to your local Spring Boot projects folder

2.2 Add additional dependencies to Spring Boot project

Since we will be using JSP and JSTL as well as MySQL, we will need to add these dependencies to our Maven project.

1. Import this Spring Boot project into Spring Tools Suite



There are no projects in your workspace.
To add a project:

Create a Java project
Create new Spring Starter Project via the Spring Initializr Web Service
An import wizards that imports code associated with a Spring Getting Started Guide
Create a project...
Import projects...

An outline is not available.

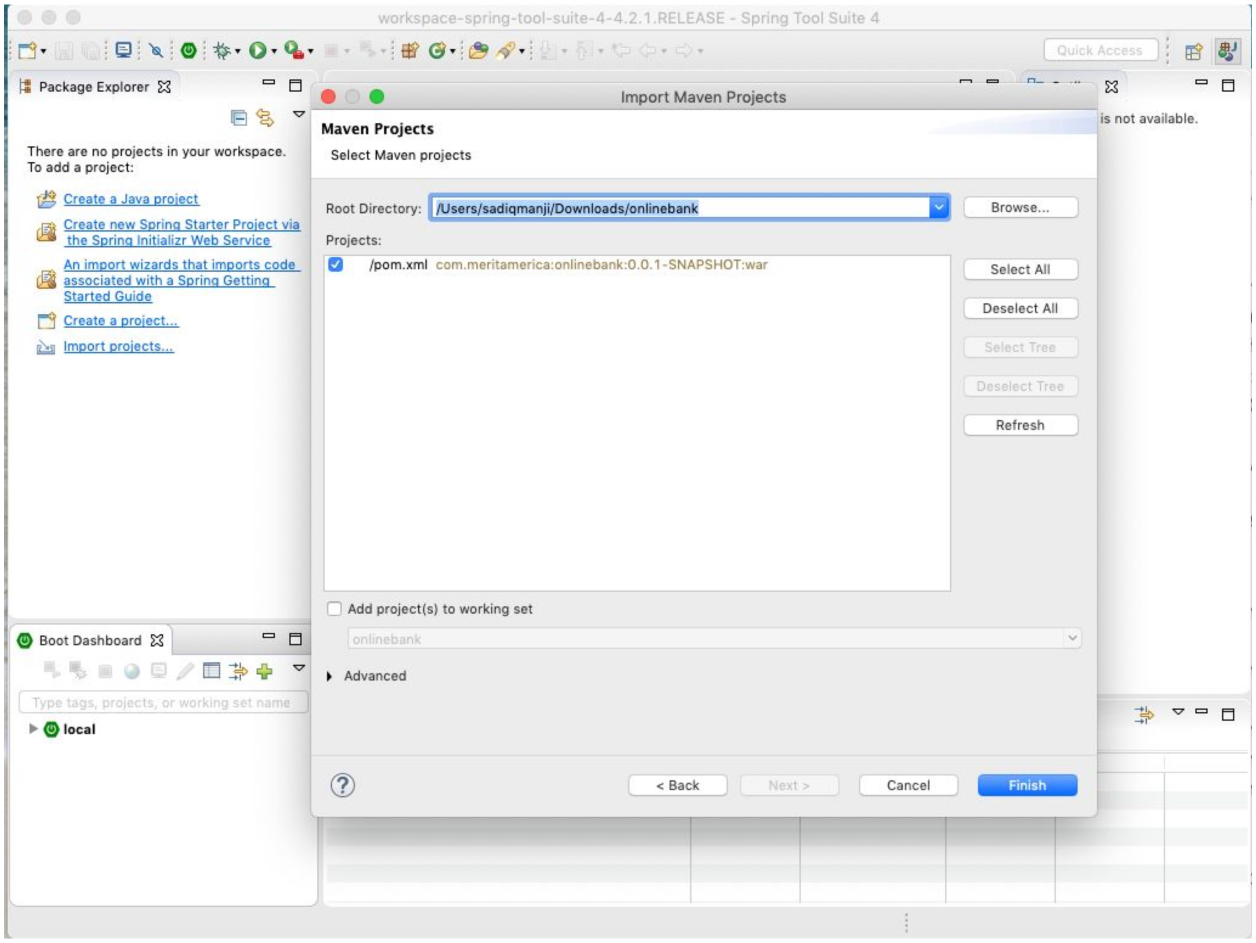
Select

Import existing Maven projects

Select an import wizard:

type filter text

- General
- Git
- Gradle
- Install
- Maven
  - Check out Maven Projects from SCM
  - Existing Maven Projects
  - Install or deploy an artifact to a Maven repository
  - Materialize Maven Binary Project
  - Materialize Maven Projects from SCM
- Plug-in Development
- Run/Debug
- Team
- TextMate
- XML


workspace-spring-tool-suite-4-4.2.1.RELEASE - Spring Tool Suite 4

There are no projects in your workspace.
To add a project:

Import Maven Projects

Maven Projects

Select Maven projects

Root Directory: /Users/sadiqmanji/Downloads/onlinebank

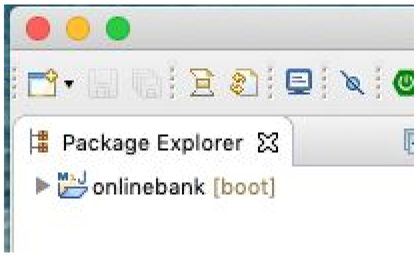
Projects:

[✓] /pom.xml com.meritamerica:onlinebank:0.0.1-SNAPSHOT:war

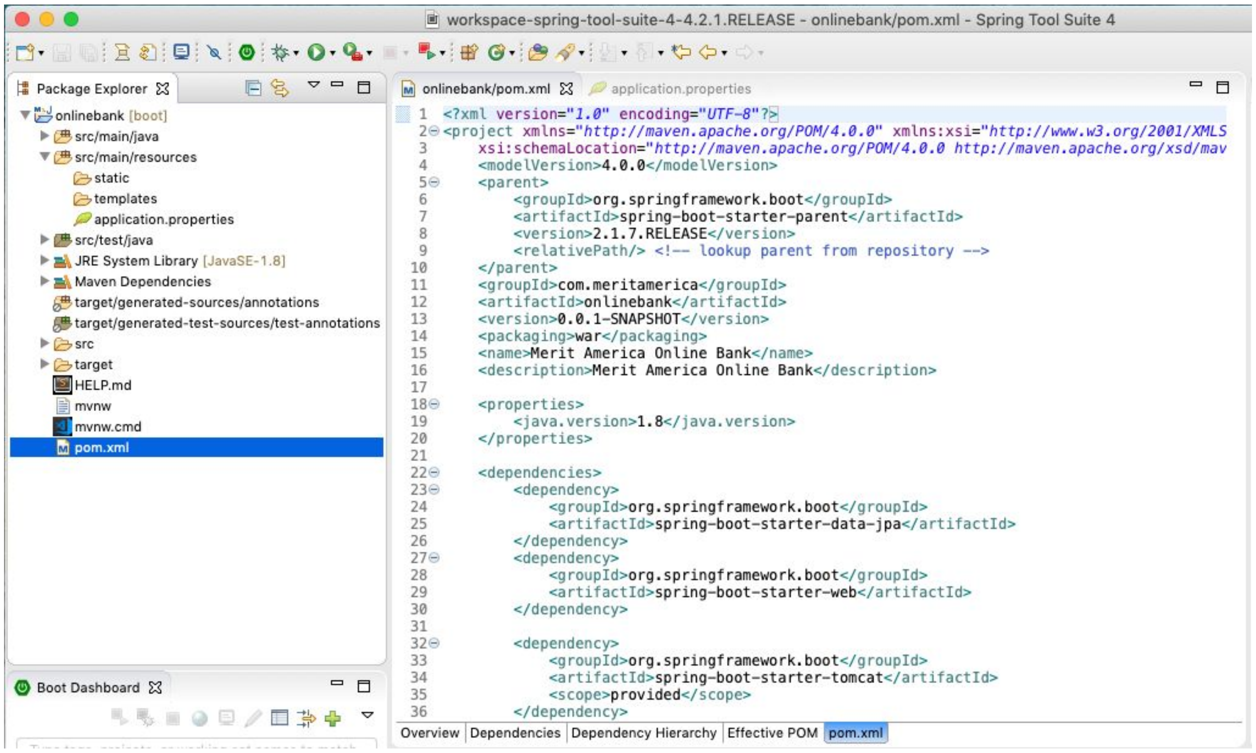
[ ] Add project(s) to working set

onlinebank

2. Wait until the project has the [boot] icon next to the project name in the Package Explorer (this may take a minute or two for Maven to do its thing)



3. Edit the pom.xml file



workspace-spring-tool-suite-4-4.2.1.RELEASE - onlinebank/pom.xml - Spring Tool Suite 4

Package Explorer

```
onlinebank [boot]
  src/main/java
  src/main/resources
    static
    templates
    application.properties
  src/test/java
  JRE System Library [JavaSE-1.8]
  Maven Dependencies
  target/generated-sources/annotations
  target/generated-test-sources/test-annotations
  src
  target
  HELP.md
  mvnw
  mvnw.cmd
  pom.xml
```

onlinebank/pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLS
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/mav
    <modelVersion>4.0.0</modelVersion>
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>2.1.7.RELEASE</version>
        <relativePath/> <!-- lookup parent from repository -->
    </parent>
    <groupId>com.meritamerica</groupId>
    <artifactId>onlinebank</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <packaging>war</packaging>
    <name>Merit America Online Bank</name>
    <description>Merit America Online Bank</description>

    <properties>
        <java.version>1.8</java.version>
    </properties>

    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-data-jpa</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-tomcat</artifactId>
            <scope>provided</scope>
        </dependency>
```

4. Ensure the dependencies section looks like the following:

```
<dependencies>
        <dependency>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-starter-data-jpa</artifactId>
        </dependency>
        <dependency>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-starter-web</artifactId>
        </dependency>
        <dependency>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-starter-tomcat</artifactId>
                <scope>provided</scope>
        </dependency>
        <dependency>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-starter-test</artifactId>
                <scope>test</scope>
        </dependency>
        <dependency>
                <groupId>org.apache.tomcat.embed</groupId>
                <artifactId>tomcat-embed-jasper</artifactId>
        </dependency>
        <dependency>
            <groupId>javax.servlet</groupId>
            <artifactId>jstl</artifactId>
        </dependency>
        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
            <scope>runtime</scope>
        </dependency>
    </dependencies>
```

5. Save the pom.xml file

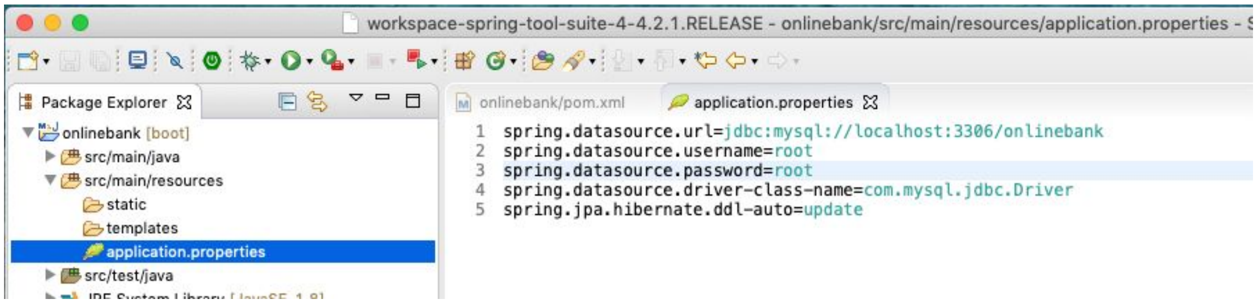
2.3 Add the README.MD file

Create a file called README.MD in the root of the project folder and simply add your name to the file.

2.4 Connect Spring Boot project to MySQL database

Since we are using Spring Data, we need to add our MySQL connection information to the project:

1. Edit the src/main/resources/application.properties file



```
spring.datasource.url=jdbc:mysql://localhost:3306/onlinebank
spring.datasource.username=root
spring.datasource.password=root
spring.datasource.driver-class-name=com.mysql.jdbc.Driver
spring.jpa.hibernate.ddl-auto=update
```

2. Input the database location, schema, and credentials per part 1 -- something like the following:

```
spring.datasource.url=jdbc:mysql://localhost:3306/onlinebank
spring.datasource.username=root
spring.datasource.password=root
spring.datasource.driver-class-name=com.mysql.jdbc.Driver
spring.jpa.hibernate.ddl-auto=update
```

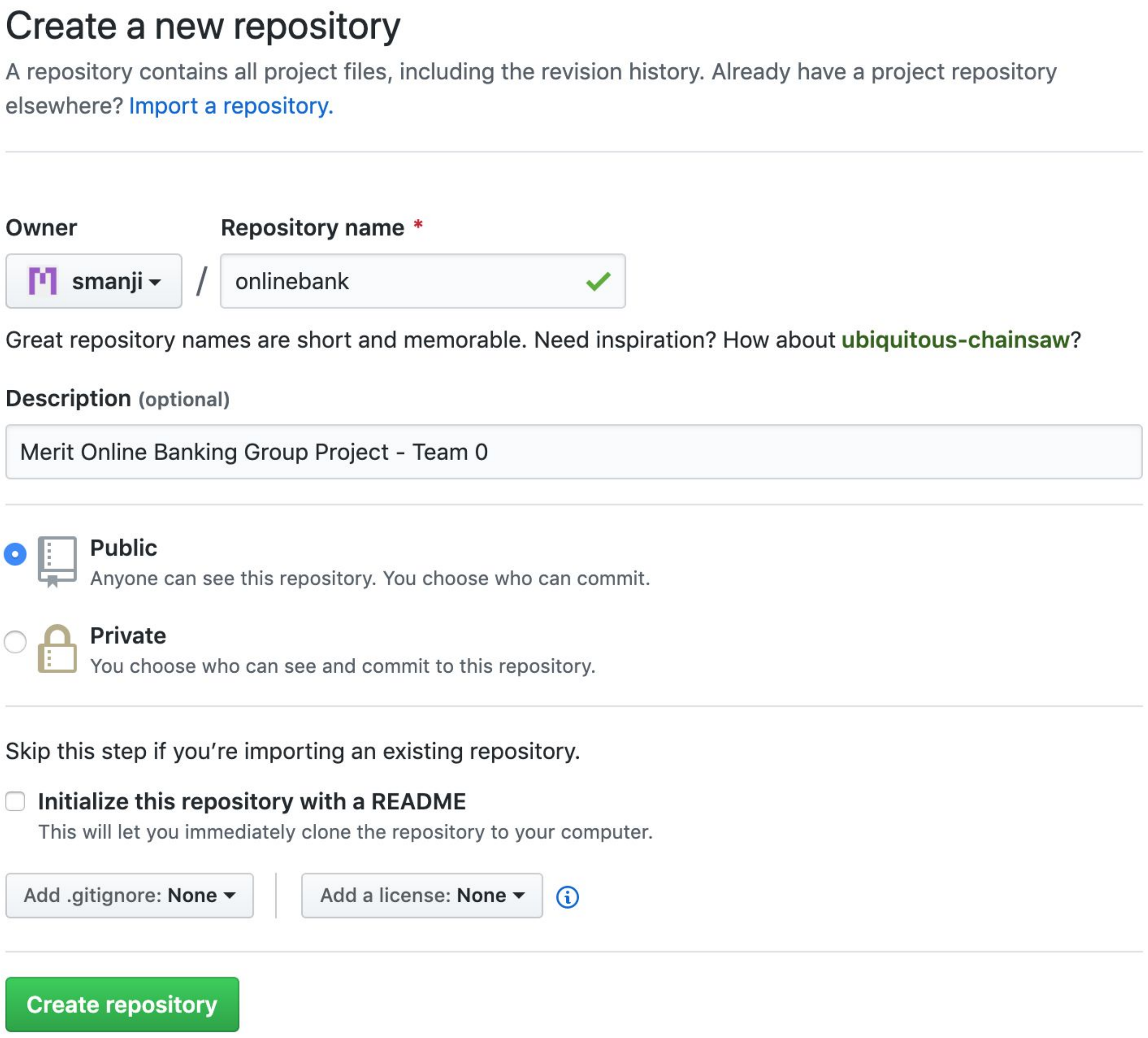
3. Save the application.properties file

3. Create Initial GitHub Repository

As the Team Lead, you will be responsible for creating a shared GitHub repository for your team and committing the above Spring Boot project to that repository.

1. In your onlinebank project folder, edit the .gitignore file:
    a. Add the following lines to the end of your file:
    ```
    ### Mac ###
    .DS_Store
    ```

2. Log into GitHub
3. Create a new repository with the following details (change your team # accordingly)



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner smanji / **Repository name** * onlinebank ✓

Great repository names are short and memorable. Need inspiration? How about **ubiquitous-chainsaw**?

Description (optional)

Merit Online Banking Group Project - Team 0

● **Public**
Anyone can see this repository. You choose who can commit.

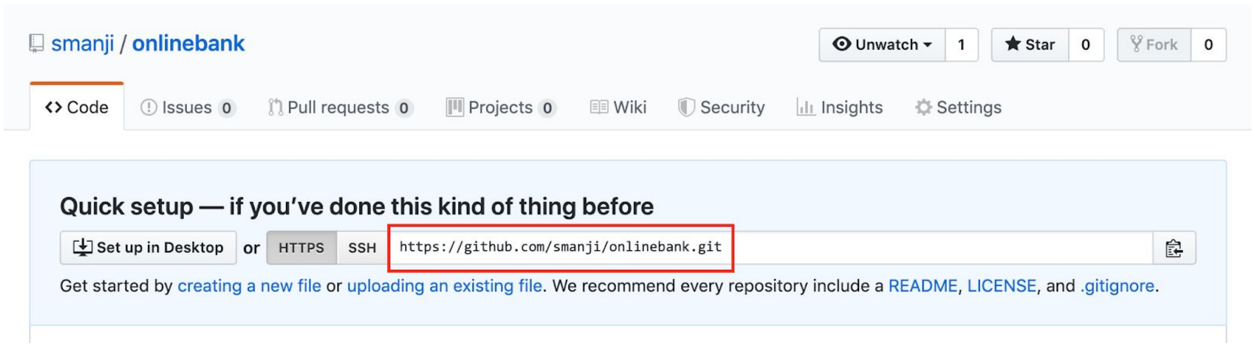
○ **Private**
You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

☐ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer.

Add .gitignore: **None** | Add a license: **None**

4. After clicking the "Create repository" button, you will be redirected to the following page. Make note of the url circled in red:



Quick setup — if you've done this kind of thing before

https://github.com/smanji/onlinebank.git

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

5. Open your git-bash
6. Change directories so that you are in the onlinebank folder with your Spring Boot project (use the cd command)
7. Type the following commands:
   a. `git init`
   b. `git add .`
   c. `git commit -m "first commit"`
   d. `git remote add origin https://github.com/smanji/onlinebank.git`
      i. Ensure to replace the url with the one provided by GitHub (circled in red in the screenshot above)
   e. `git push -u origin master`
8. Go back to the GitHub page in your browser and refresh the page
9. Observe that the project source code has been pushed to GitHub!

3.1 Second commit of JSP/JSTL configuration

Nice work! Since we are using JSP/JSTL, you will want to follow the steps in Coding Dojo's Spring Boot Templating to add the webapp/WEB-INF folders and configure the application.properties file to look there for your views.

You may also want to create some packages to standardize where the controllers, models, repositories, and services will go as well as any folders to store static web resources like CSS, JavaScript, and image files. Although, you can also work with your team to gain a consensus for this and implement these standards together.

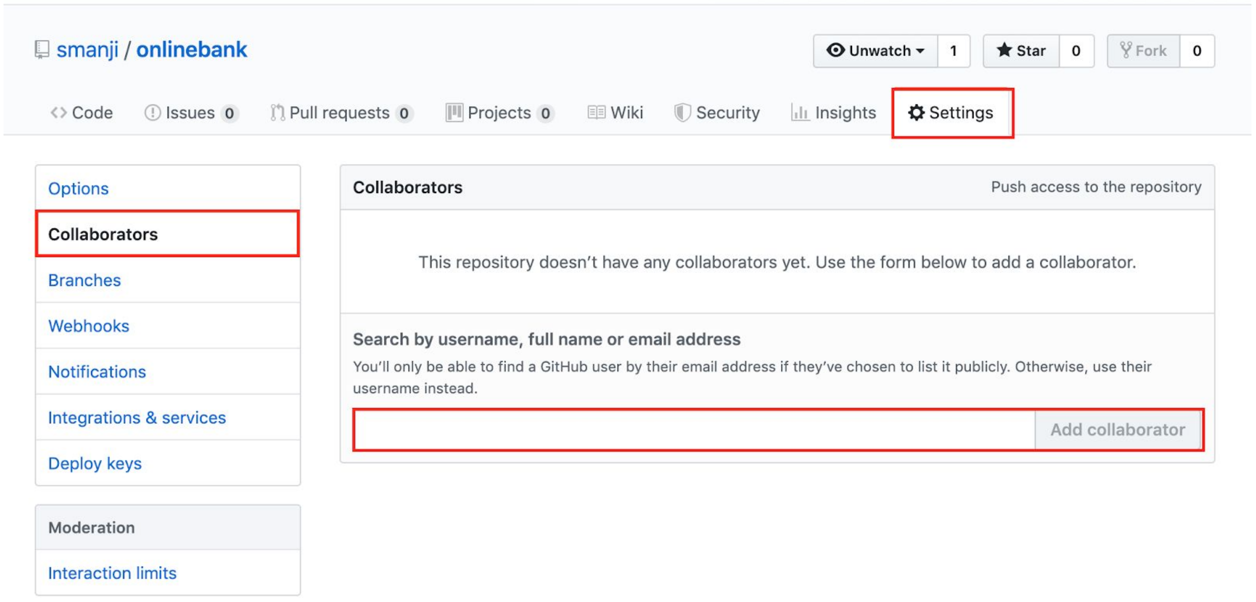
With those changes, you will want to perform the following git commands to push the new changes:

1. `git add .`
2. `git commit -m "Added Spring MVC configuration for views"`
3. `git push -u origin master`

3.2 Add your team members as collaborators

Now let's add your team members as collaborators to this GitHub repository so that they can push their contributions to the repository.

1. Click the Settings tab of your GitHub repository
2. Select the Collaborators tab on the left
3. Add each team member as a collaborator



Collaborators

Push access to the repository

This repository doesn't have any collaborators yet. Use the form below to add a collaborator.

Search by username, full name or email address

You'll only be able to find a GitHub user by their email address if they've chosen to list it publicly. Otherwise, use their username instead.

3.3 (Optional) Connect GitHub to Slack

Your team may be interested in having a Slack channel to communicate about the project -- and may also be interested in receiving messages on that Slack channel each time a team member commits code to the GitHub repository.

If interested, there is an integration available between Slack and GitHub and the installation instructions can be followed here: https://github.com/integrations/slack

4. Connect Git to GitHub Repository

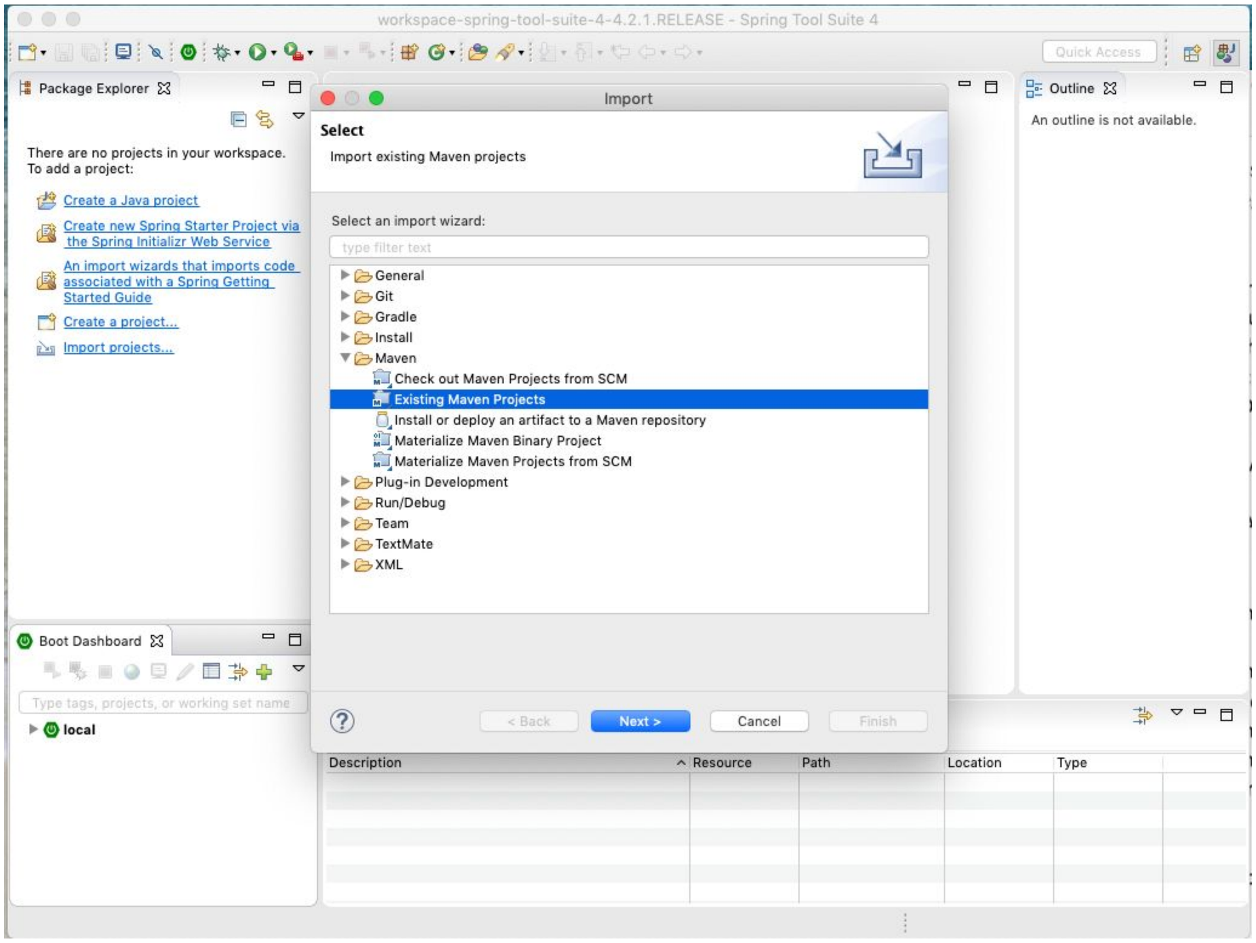
Note: As the Team Lead, you can skip this section

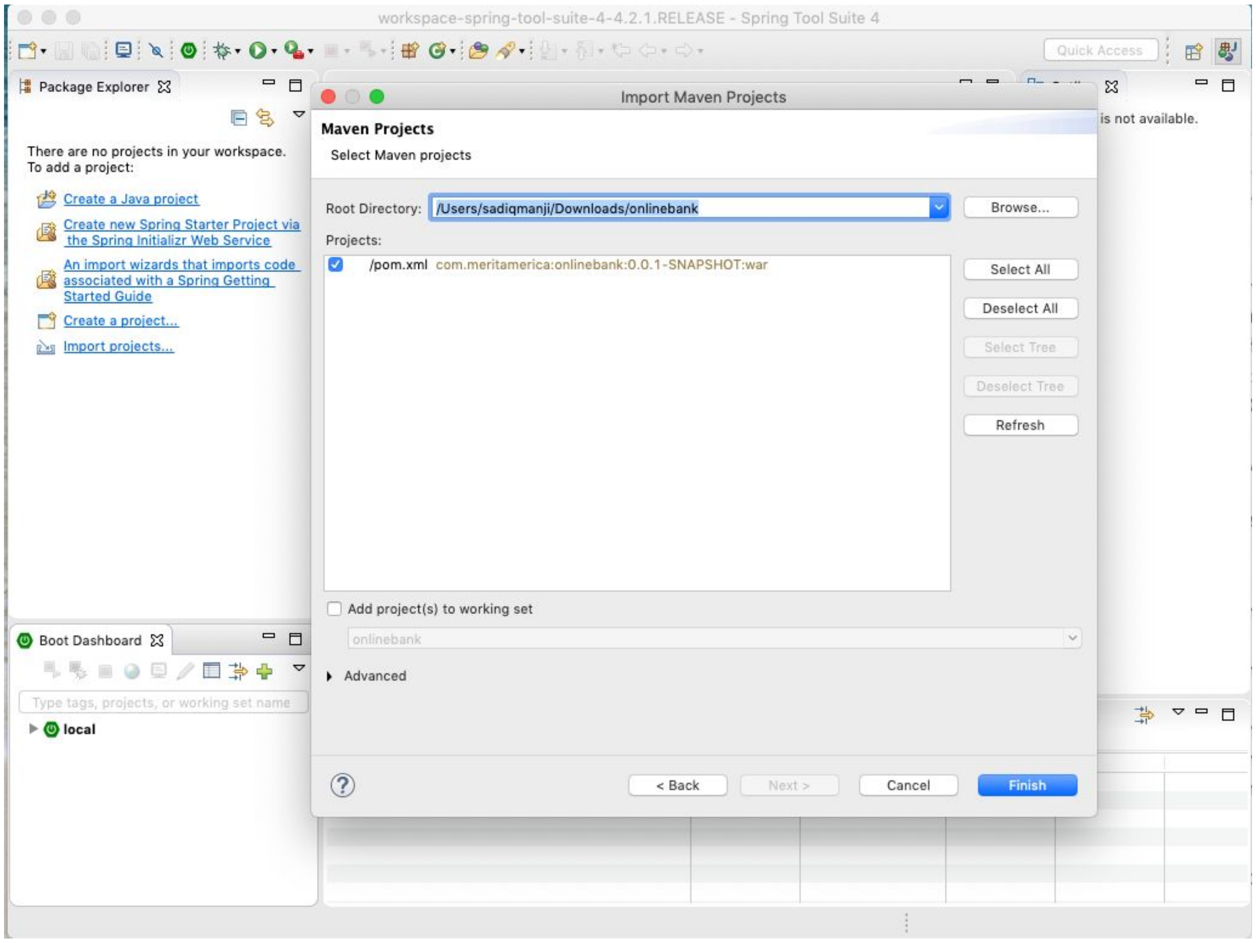
Now that your Team Lead has notified you that the project is up on GitHub, you'll want to clone the repository locally so you can build and run the project locally as well as contribute to the project and push your changes back to GitHub in order to collaborate with your team members.

1. Open git-bash
2. Change directories to a place where you would like to download the repository (e.g. your Spring Workspace)
3. Type the following command:

   ```
   git clone https://github.com/smanji/onlinebank.git
   ```

   i. Ensure this is actually the url to your team's repo as provided by your Team Lead
4. Observe that the project has been downloaded
5. Open your Spring Tools Suite
6. Import this project:


workspace-spring-tool-suite-4-4.2.1.RELEASE - Spring Tool Suite 4
Quick Access
Package Explorer
There are no projects in your workspace.
To add a project:
Create a Java project
Create new Spring Starter Project via the Spring Initializr Web Service
An import wizards that imports code associated with a Spring Getting Started Guide
Create a project...
Import projects...
Import
Select
Import existing Maven projects
Select an import wizard:
type filter text
General
Git
Gradle
Install
Maven
Check out Maven Projects from SCM
Existing Maven Projects
Install or deploy an artifact to a Maven repository
Materialize Maven Binary Project
Materialize Maven Projects from SCM
Plug-in Development
Run/Debug
Team
TextMate
XML
< Back
Next >
Cancel
Finish
Outline
An outline is not available.
Boot Dashboard
Type tags, projects, or working set name
local
Description
Resource
Path
Location
Type



Root Directory: /Users/sadiqmanji/Downloads/onlinebank

Projects:

☑ /pom.xml com.meritamerica:onlinebank:0.0.1-SNAPSHOT:war

☐ Add project(s) to working set

onlinebank

7. Edit and save the README.MD file: add your name to the end of the file
8. Go back to your git-bash
9. Ensure you are in the project folder
10. Type the following commands:
    a. `git add .`
    b. `git commit -m “Added my name to the README.MD”`
    c. `git push -u origin master`
11. Go to the GitHub repository and check that your changes were pushed. This is how you will push your changes going forward.
12. Slack your team members so they know you’ve successfully cloned the project and added your name to the README.MD.

5. Pulling in collaborator changes

When you know that a team member has pushed their changes, open your git-bash and go to your project folder and type the following command:

a. `git pull`

Take a look at the README.MD file in your Spring Tools Suite and observe that the changes have been pulled in.

In the future, when you perform a `git pull`, you will see the listing of files that have been changed.

Before you start to make changes, it's always a good idea to perform a `git pull` to ensure you are working on the latest source code to minimize conflicts. Ideally, you will not be working on the same files as your team members to further minimize the risk of conflicts.