SETUP A DEVELOPMENT ENVIRONMENT for java server app (middle tier)

folder name:

shopping-cart

Java Developer Kit 8 (JDK8)

DOWNLOAD AND INSTALL JDK 8

We will be using the latest version of Java throughout the semester

Navigate to Oracle's Java Development Kit (JDK) 8 download website

http://www.oracle.com/technetwork/java/javase/downloads/index.html

DOWNLOAD AND INSTALL JDK 8

Download and install the JDK for your particular operating system. Once the JDK has installed, verify you have the right version of Java installed. From your terminal or console, type the following

```
$ java -version
java version "1.8.0_77"
Java(TM) SE Runtime Environment
(build 1.8.0_77-b03)
```

SETUP MAVEN

DOWNLOAD AND INSTALL MAVEN

Maven is a Java dependency package manager. It simplifies managing the lifecycle of projects such as downloading libraries, compiling, running automated tests, and packaging projects for deployment. Download maven from

http://maven.apache.org/download.cgi

After installing, make sure the following command should print the version of maven

```
$ mvn -version
```

SPRING BOOT SETUP

ON macOS, INSTALL THE SPRING CLI USING BREW

Install the Spring CLI as described in their documentation

On macOSX, use brew to install spring boot

```
$ brew tap pivotal/tap
$ brew install springboot
```

If you don't have brew, on macOS, install it as follows:

```
$ ruby -e "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/i
nstall/master/install)"
```

CREATE A WEB APP USING THE SPRING CLI
(For TA to grading, you don't need to do this.)

From a command line terminal, use the spring command to create a simple spring web app called myapp, and then change to the new directory

```
$ spring init --dependencies=web myapp
```

```
$ cd myapp
```

MYSQL WORKBENCH

MySQL WORKBENCH

Download and install MySQL Workbench from

https://dev.mysql.com/downloads/workbench/

NODE.JS

DOWNLOAD AND INSTALL Node.js (macOS)

Homebrew makes the process of installation of Node a one-step process. By using Homebrew, we do not need to manually add the path of node executable.

Step1: From the terminal execute the command:

```
$ brew install node
```

Node is installed on your system now.

Test the installed packages (Restart your computer)

Test Node: open command prompt and type **node -v**. You should see the downloaded version of node as “v7.3.0”

Test NPM: In command prompt, type **npm -v**. You should see the downloaded version of npm as “3.10.10”

DOWNLOAD AND INSTALL Node.js (Windows)

1. Download the windows installer for Node.js from the Node.js website
2. Run the installer (the .msi file downloaded in Step 1)
3. Accept the license agreement and all defaults and click install.

Test the installed packages (Restart your computer)

Test Node: open command prompt and type **node -v** . You should see the downloaded version of node as “v7.3.0” (the version you just downloaded in previous step).

Test NPM: In command prompt, type **npm -v**. You should see the downloaded version of npm as “3.10.10” (the version you just downloaded in previous step).

Run the java server app (middle tier) folder name: shopping-cart

RUN the app

- Since I already finish the app, you don’t need to create it. In the command line, you need to cd to the app directory, i.e.
- $ cd shopping-cart

RUN YOUR SPRING BOOT FROM COMMAND LINE

You can run your app from your command line. First compile and package using mvn

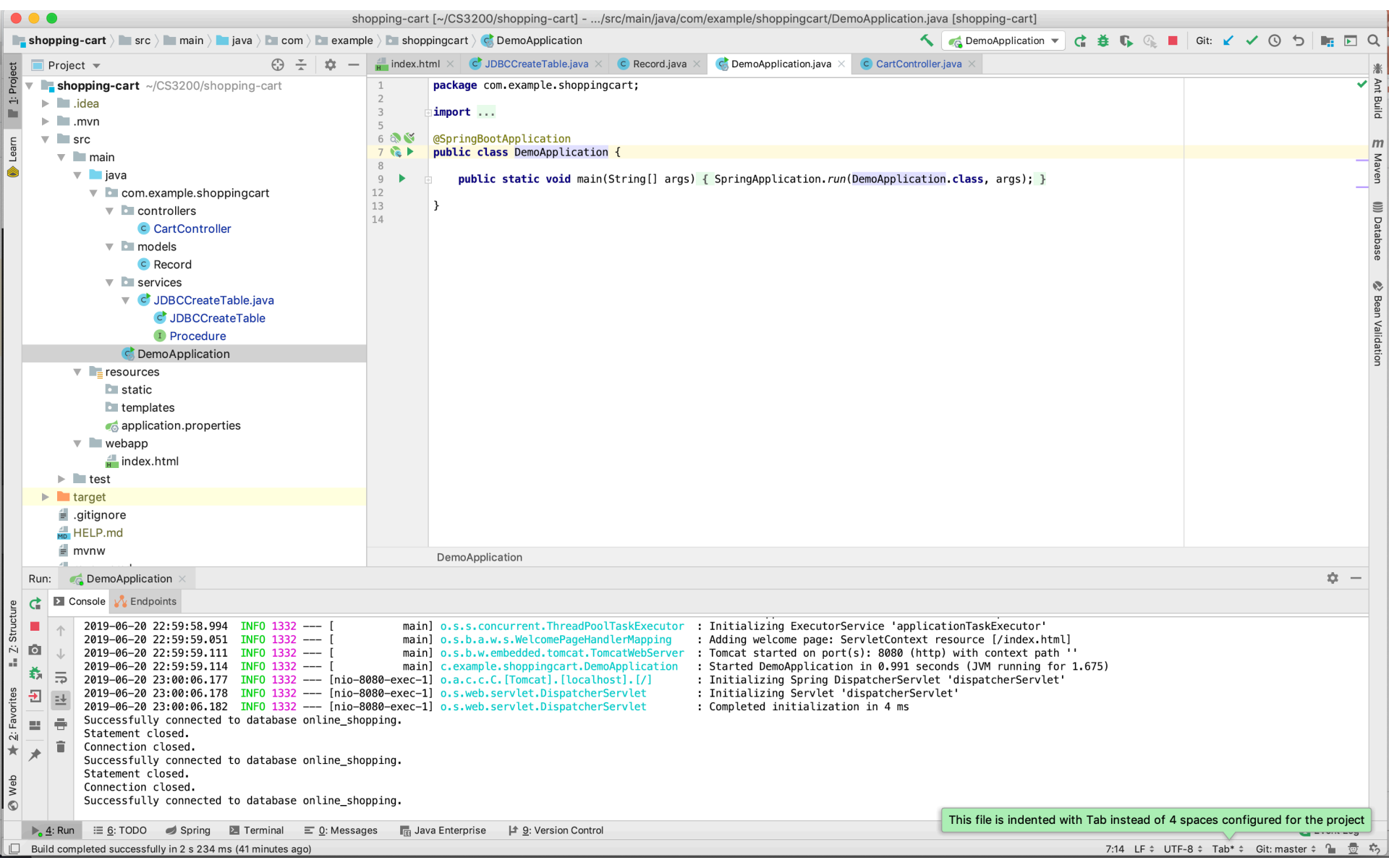
```
$ mvn clean install
```

This will package your app into a JAR file under the target directory. Run it as an executable packaged app

```
$ java -jar target/*.jar
```

Point your browser to **http://localhost:8080** to see app (However, since it's just a server, you can only see a title.)

- If you have intellij, you can run it through main function.
- see next slide.


shopping-cart [~/CS3200/shopping-cart] - .../src/main/java/com/example/shoppingcart/DemoApplication.java [shopping-cart]
shopping-cart ~/CS3200/shopping-cart
.idea
.mvn
src
main
java
com.example.shoppingcart
controllers
CartController
models
Record
services
JDBCCreateTable.java
JDBCCreateTable
Procedure
DemoApplication
resources
static
templates
application.properties
webapp
index.html
test
target
.gitignore
HELP.md
mvnw
index.html
JDBCCreateTable.java
Record.java
DemoApplication.java
CartController.java
package com.example.shoppingcart;
import ...
@SpringBootApplication
public class DemoApplication {
public static void main(String[] args) { SpringApplication.run(DemoApplication.class, args); }
}
DemoApplication
Run: DemoApplication
Console
Endpoints
2019-06-20 22:59:58.994 INFO 1332 --- [main] o.s.s.concurrent.ThreadPoolTaskExecutor : Initializing ExecutorService 'applicationTaskExecutor'
2019-06-20 22:59:59.051 INFO 1332 --- [main] o.s.b.a.w.s.WelcomePageHandlerMapping : Adding welcome page: ServletContext resource [/index.html]
2019-06-20 22:59:59.111 INFO 1332 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''
2019-06-20 22:59:59.114 INFO 1332 --- [main] c.example.shoppingcart.DemoApplication : Started DemoApplication in 0.991 seconds (JVM running for 1.675)
2019-06-20 23:00:06.177 INFO 1332 --- [nio-8080-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring DispatcherServlet 'dispatcherServlet'
2019-06-20 23:00:06.178 INFO 1332 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
2019-06-20 23:00:06.182 INFO 1332 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : Completed initialization in 4 ms
Successfully connected to database online_shopping.
Statement closed.
Connection closed.
Successfully connected to database online_shopping.
Statement closed.
Connection closed.
Successfully connected to database online_shopping.
This file is indented with Tab instead of 4 spaces configured for the project
Build completed successfully in 2 s 234 ms (41 minutes ago)

Run the react client app (front end) folder name: shopping-cart-client-react

Install the create react app CLI
(As a TA, You don't need to do it.)

Use `npm` to install the create react app CLI and create a boilerplate application

```
npm install -g create-react-app
create-react-app my-app
```

Start the application

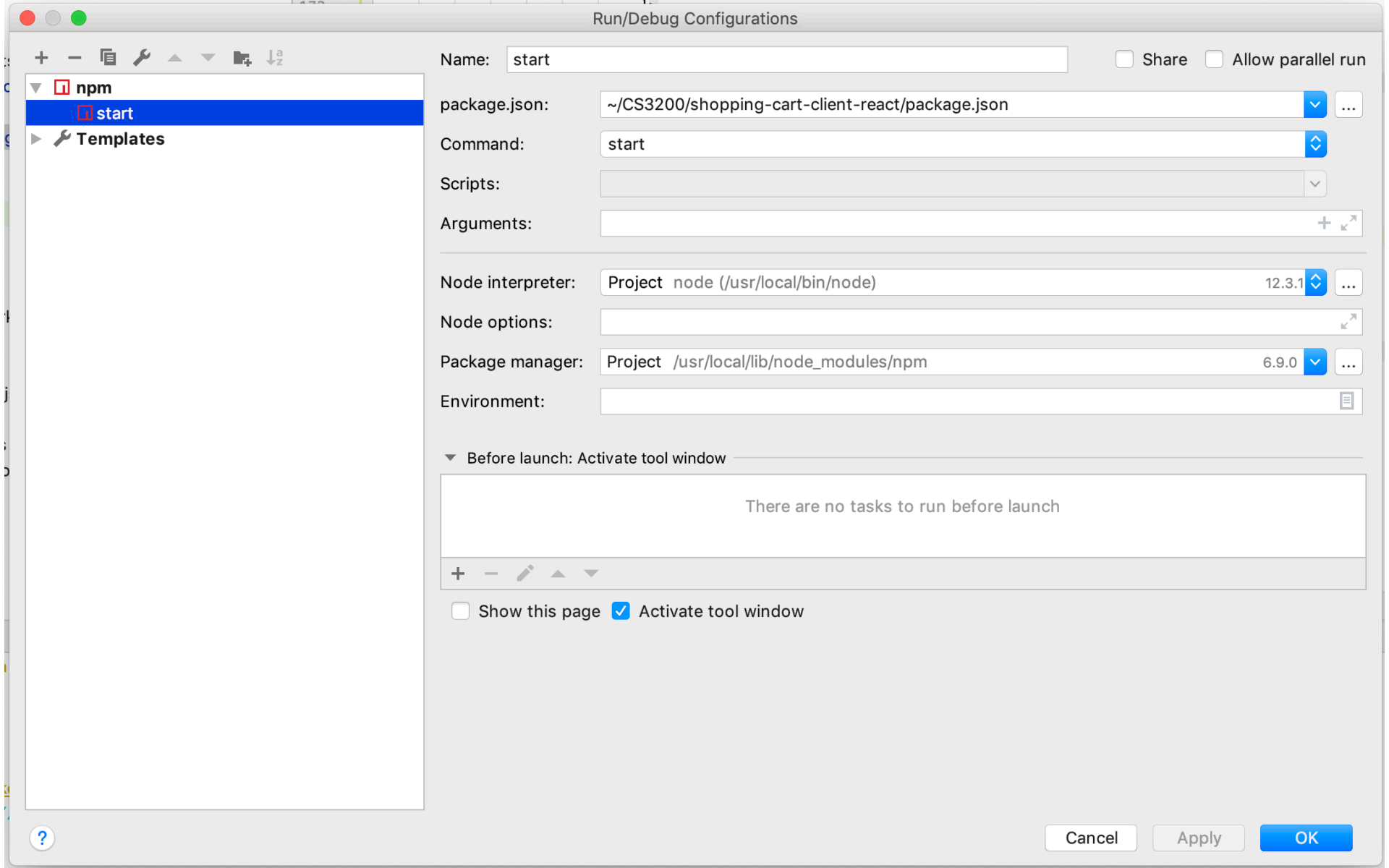
First, in the command line, you need to cd to the client app directory. i.e.

```
$ cd shopping-cart-client-react
```

Second, use “npm start” run the app.

```
$ npm start
```

- If you have intellij ultimate edition, you can run it through configurations.
- see next slide.


Run/Debug Configurations
npm
start
Templates
Name:
start
Share
Allow parallel run
package.json:
~/CS3200/shopping-cart-client-react/package.json
Command:
start
Scripts:
Arguments:
Node interpreter:
Project
node (/usr/local/bin/node)
12.3.1
Node options:
Package manager:
Project
/usr/local/lib/node_modules/npm
6.9.0
Environment:
Before launch: Activate tool window
There are no tasks to run before launch
Show this page
Activate tool window
Cancel
Apply
OK