

**DIRECT STEM**

**Visualization Tool for Composition**

**Of Cloud Computing Services**

**Advisor: Dr. Jiang Guo**

**Windows Instructions:**

**Frond End Requirements**

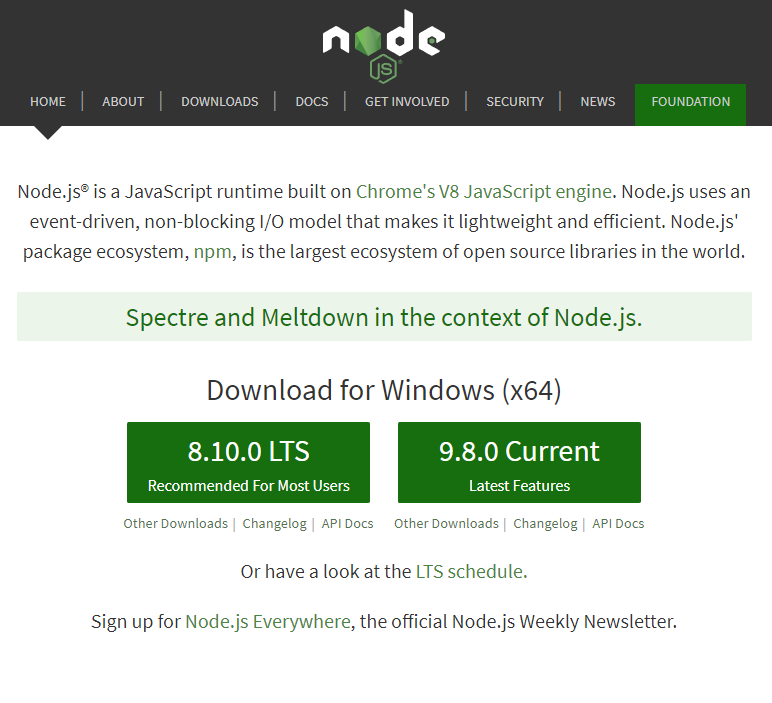
**Node.js and NPM**

Node.js is a platform for JavaScript. It is built on Chrome’s V8 JavaScript engine. Node.js uses an event-driven, non-blocking I/O model to remain lightweight and efficient. Node.js is needed to use Angular CLI which is one of our main tools for this project.

NPM stands for Node Package Manager. NPM lets you install and manage your project’s dependencies. All of our project’s front end dependencies are listed inside our **package.json** file and running **npm install** will immediately have all of the dependencies installed.

<https://nodejs.org/en/>

1. Download the Windows installer from the Node.js website.

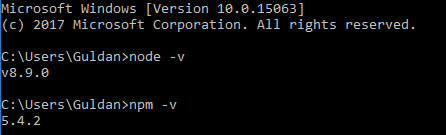


2. Run the installer.

3. Follow the directions of the installer. Choose the default values for the installer if you are not sure of what options to select.

4. Test if Node is properly installed by running the command **node -v** in the Windows command prompt.

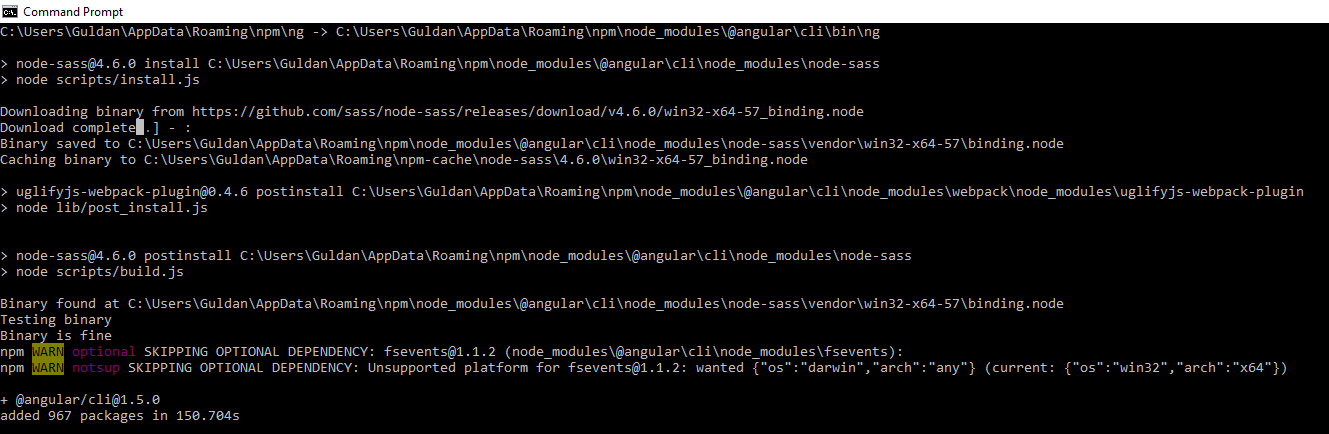
5. Test if NPM is also properly installed by running the command **npm -v** in the Windows command prompt.



**Angular CLI**

Angular CLI means Angular Command Line Interface. Angular CLI is a tool to initialize, develop, scaffold and maintain Angular applications. Angular is a TypeScript-based open-source front-end web application framework.

1. Open a Command Prompt and run the command **npm install -g @angular/cli.**



2. Check if Angular CLI is properly installed by running the command **ng --version.**

The prefix **ng** stands for “Angular” and it is how you start using the Angular CLI.



**Git (Optional)**

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency. GitHub is where developers store their projects and network. Our project is stored on GitHub for easy access.

<https://git-for-windows.github.io/>

1. Download the Windows installer from the git website.

2. Run the installer.

3. Follow the directions of the installer. Choose the default values of the installer if you are unsure of what options to select.

4. Open a Command Prompt or use Git Bash if you chose to add it in.

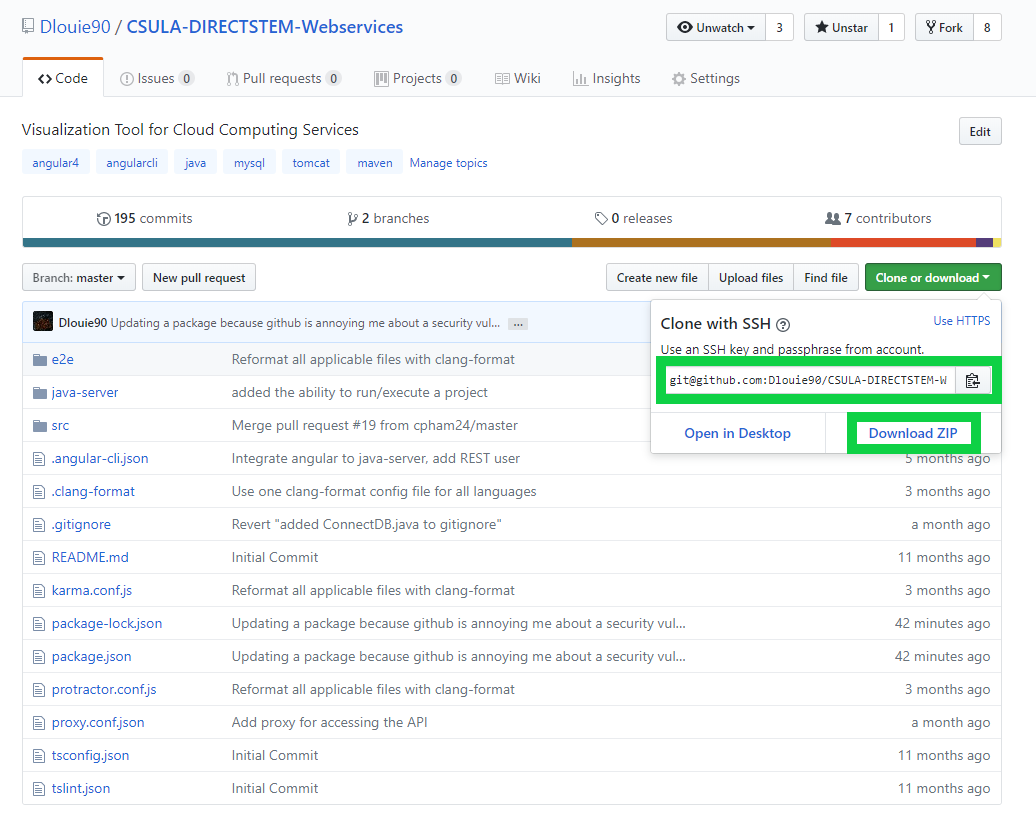
5. Run two commands to configure your username and email for git: **git config –global user.name “John Doe”** and **git config –global user.email** [johndoe@gmail.com](mailto:johndoe@gmail.com)

6. You need to generate a new SSH key for authentication so follow the instructions in <https://help.github.com/articles/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent/>

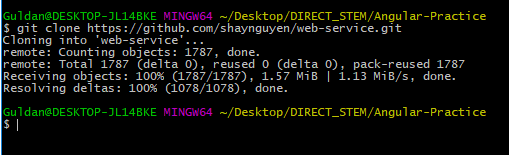
**Final Steps**

<https://github.com/Dlouie90/CSULA-DIRECTSTEM-Webservices>

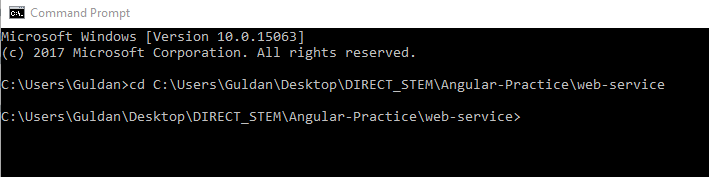
1. Download the project as a zip file and extract the files or clone the repository if you installed git.



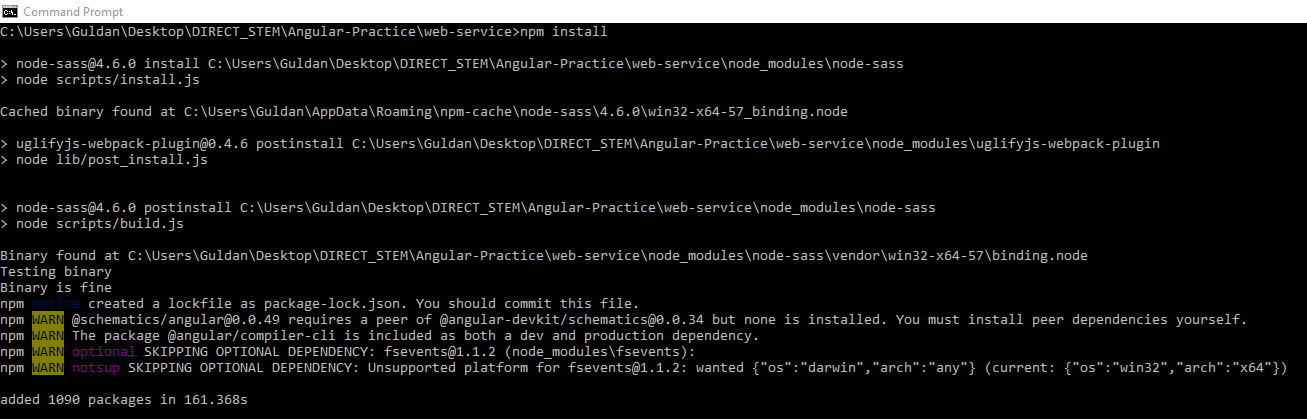
a. Git clone example



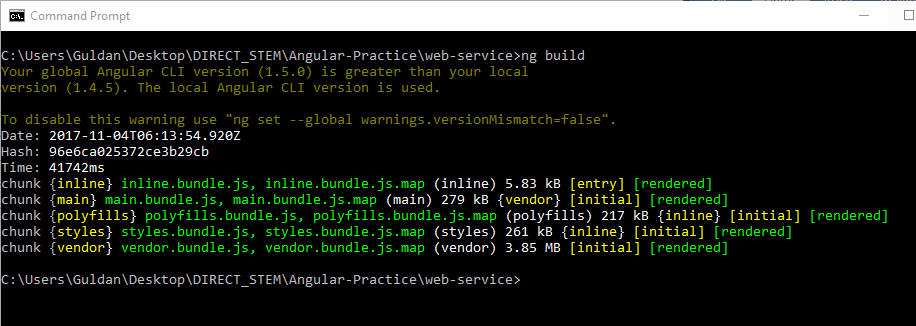
2. Open the command prompt and change your directory to the location of the project.



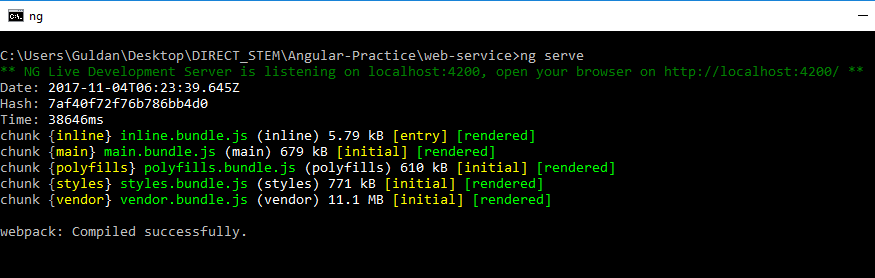
3. Run the command **npm install** in the project’s directory (where the package.json file is located) to install the dependencies required for the project. You should see a node\_modules folder in the directory of the project after it has been completed.



4. Run the command **ng build** to compile the application in the directory.



5. Finally, run the command **ng serve** to build the application and start a web server.



6. Then open a web browser and the project shall be available at <localhost:4200>.

**Note:** You will not be able to make REST api calls or any calls to the database until the backend is set up. In this current state, you can only view the web application with restrictive capabilities (e.g. not able to create projects)



**Back End Requirements**

**Java Development Kit**

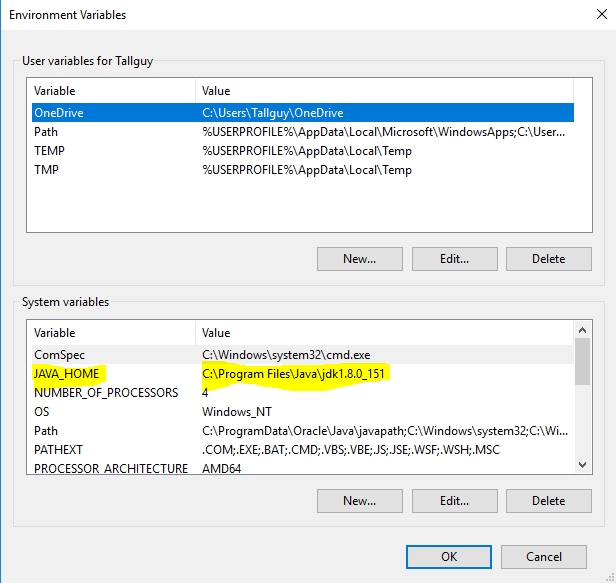
Java Development Kit is a software development environment used for developing Java applications. Our project requires at least JDK 1.7.

<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

1. Download the JDK installer for Windows from the Oracle website.

2. Run the installer.

3. Add **JAVA\_HOME** to the system environment variables in the control panel.



**MySQL**

MySQL is the database we use to store our data.

<https://dev.mysql.com/downloads/mysql/>

1. Download the Windows Installer for MySQL community edition.

2. Go through the installer to get MySQL installed. All the default options are fine, but you only need the server installed.

3. Either start MySQL Workbench or access MySQL through the MySQL Command Line so you can run the script located at **CSULA-DIRECTSTEM-Webservices\java-server\sql**.

**Apache Maven**

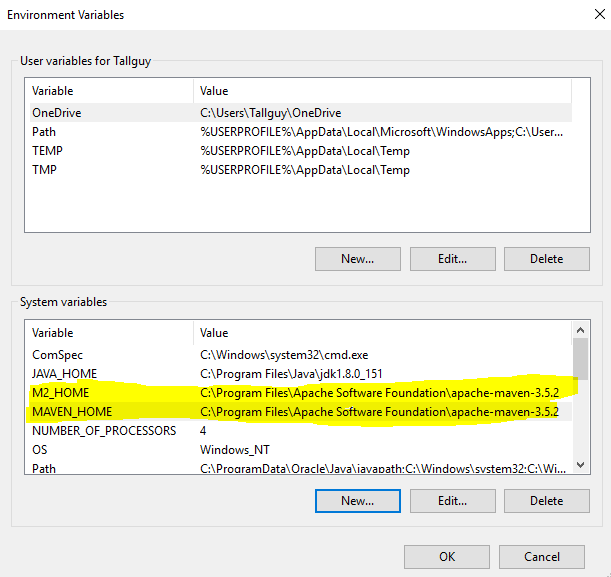
Apache Maven is a software project management and comprehension tool.

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

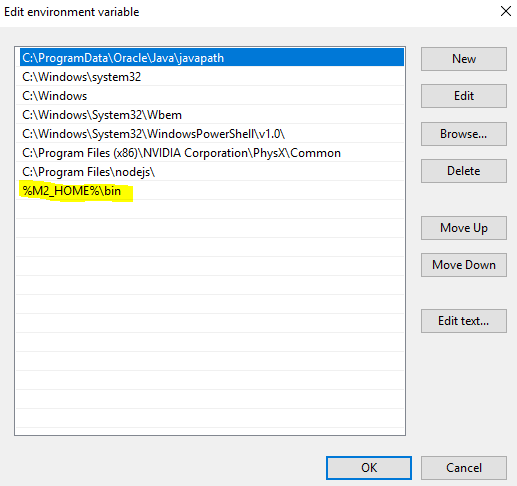
1. Download the **apache-maven-3.5.2-bin.zip** file from the Apache Maven website.

2. Extract the zip file to your desired location.

3. Add **M2\_HOME** and **MAVEN\_HOME** to the system variables in control panel.



4. Update the path variable to include **%M2\_HOME%\bin** so you can run the maven command.



5. Finally, to verify if maven is installed, run the command **mvn -version** in the command prompt.



**Final Steps Command Line Version**

1. Follow the Final Steps 1-4 in the Front-End instructions to setup the project.

2. Make sure the MySQL Windows Service is running by either using the MySQL Notifier or checking the list of services in Windows.

3. Use the command prompt or git bash to go to web-service\java-server directory.

4. Run the command **mvn tomcat7:run**. (Note: **Ctrl + C** should usually interrupt the server if you need to do so)

**Note:** You need **ng serve --proxy-config proxy.conf.json** or **npm start** (which is a shortcut for **ng serve --proxy-config proxy.conf.json –open**)

**Final Steps IntelliJ Version**

**Apache Tomcat**

<https://tomcat.apache.org/download-90.cgi>

1. Download the Apache Tomcat **Windows Service Installer** from the tomcat website.

2. Run the installer.

**Note:** You only need Apache Tomcat installed to run the project in IntelliJ.

**IntelliJ**

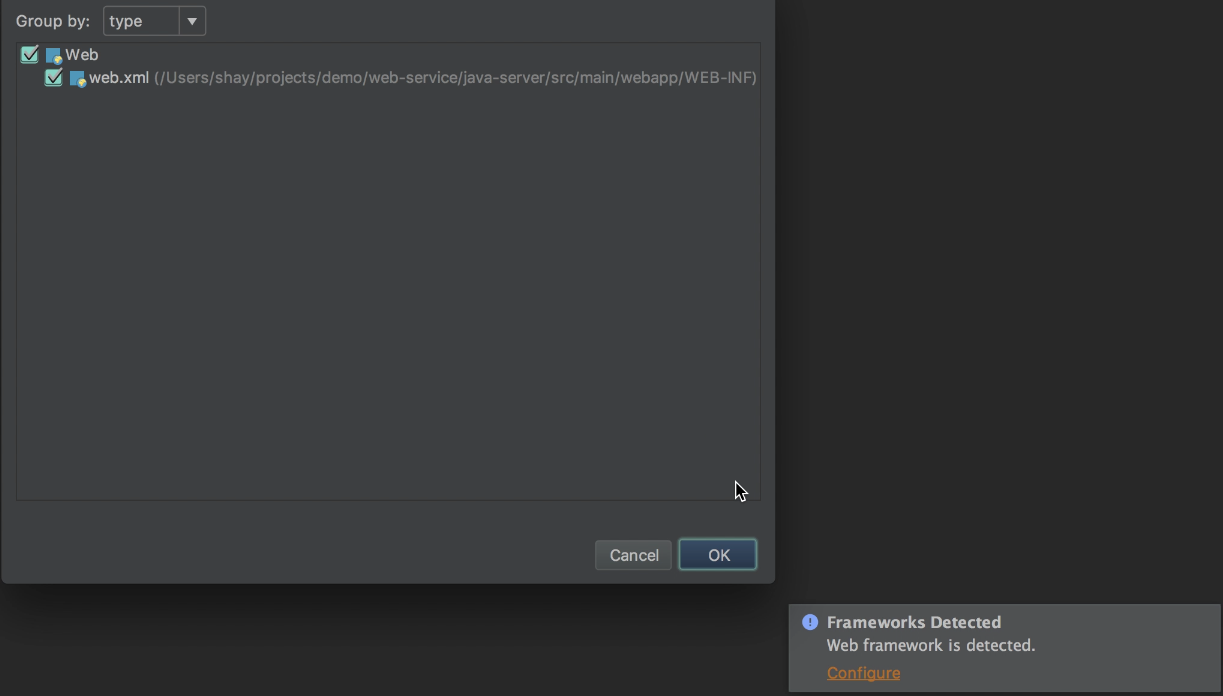
[https://www.jetbrains.com/idea/download/#section=windows](https://www.jetbrains.com/idea/download/%23section=windows)

1. Install IntelliJ IDEA Ultimate or Community from the JetBrains website.

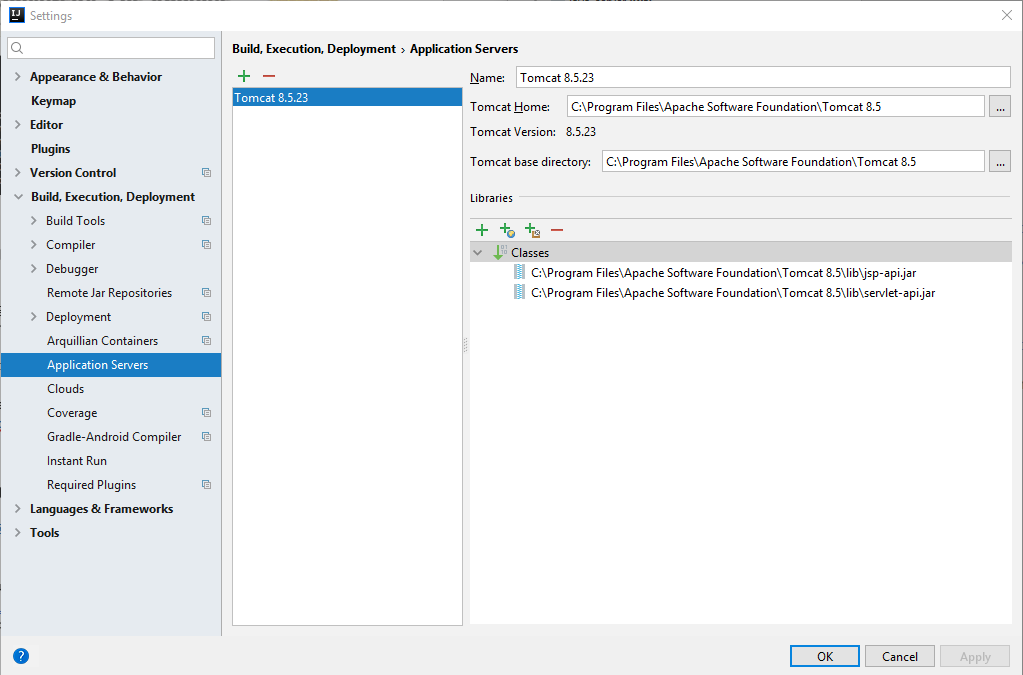
2. Follow the Final Steps 1-4 in the Front-End instructions to setup the project.

3. Start the IntelliJ application and open the web-service project.

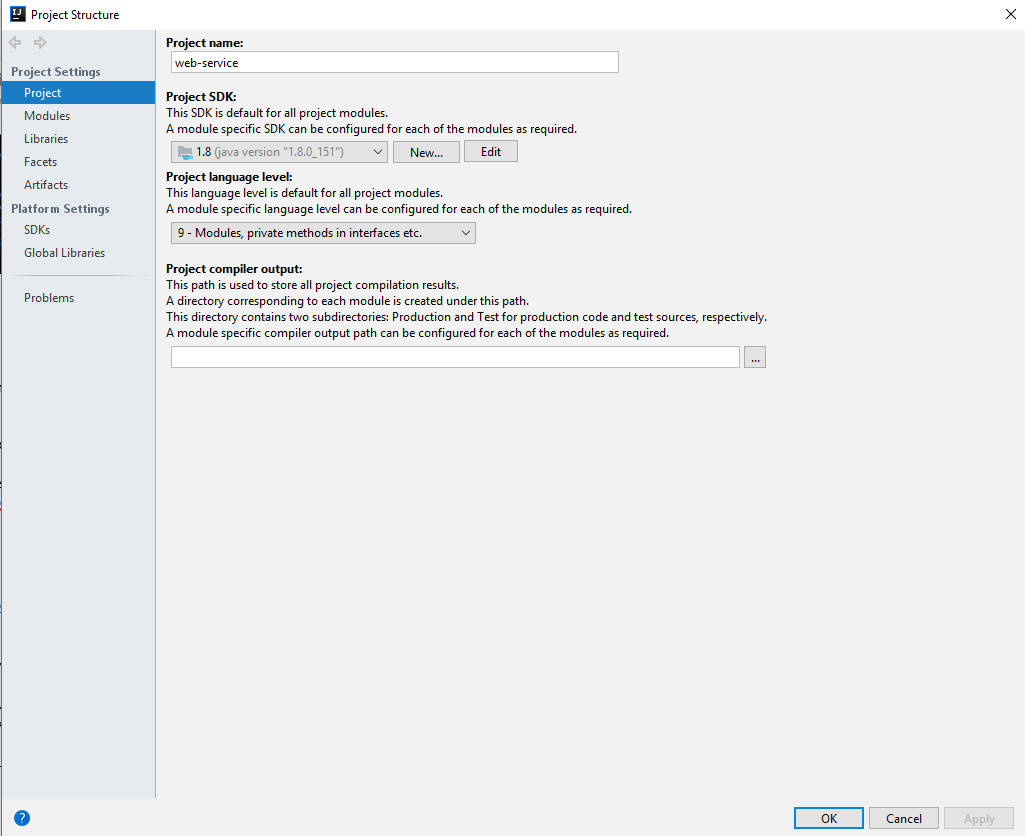
4. A pop-up should appear saying Frameworks Detected. Click on Configure and add the web.xml and click ok.



5. Go to File -> Settings -> Build, Execution, Deployment -> Application Servers and add your tomcat server.

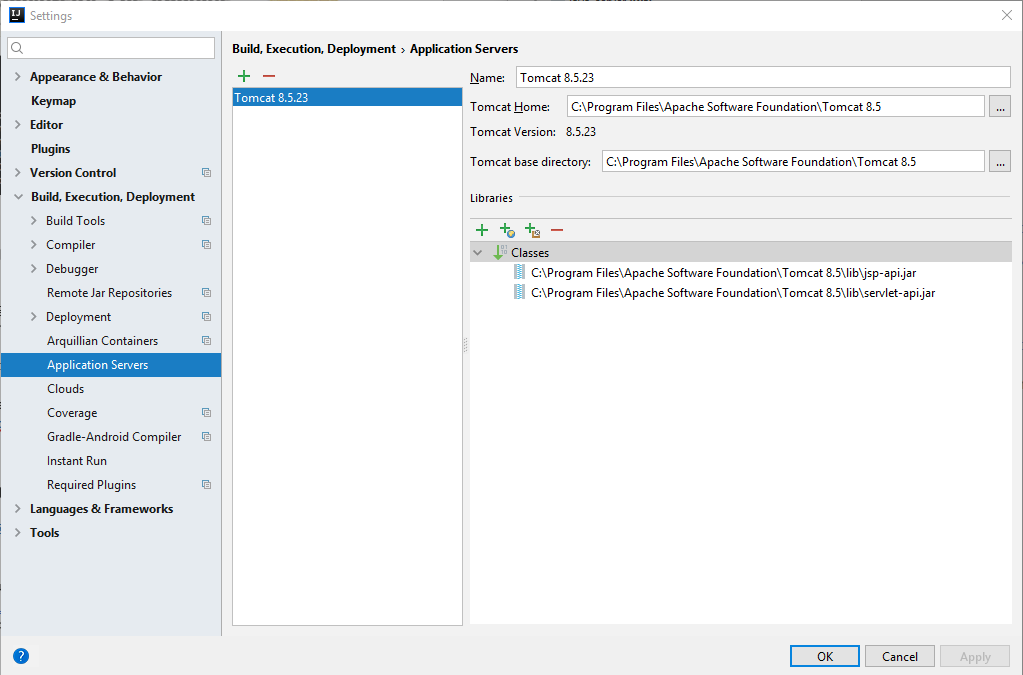


6. Go to File -> Project Structure -> Project Settings -> Project and make sure your Project SDK is set to your Java Development Kit (JDK).

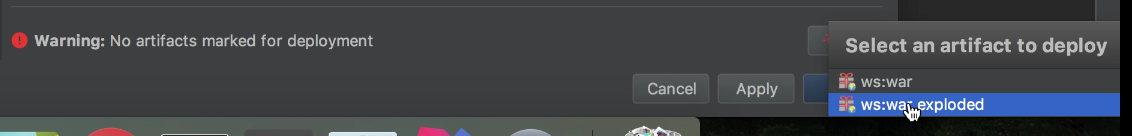


7. In your project, go to java-server -> pom.xml. Then right click and click on Add as a Maven Project. This will allow all the dependencies needed for this project to be downloaded. This may take a few minutes.

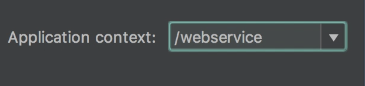
8. Go to Run -> Edit Configurations and click on the + symbol to add a local Tomcat server to the configurations.



9. In the bottom, It will say Warning: No artifacts marked for deployment and Fix on the right side. Click on Fix and select ws:war exploded.



10. On the deployment tab, there is an application context field. Type /webservice and click ok.



11. Now run your tomcat server in IntelliJ and go to the webservice directory and use the command **npm start**. Then you should be able to make REST API calls to the backend.

**Mac and Linux Instructions**

**How to set up Angular on Unix-based Operating Systems**

(tested on Mac OS X 10.13.1 and Linux Mint 17)

**Assumption:** This guide assumes that you are familiar with the command line interface (i.e.: Terminal) of Unix-based systems, and you are not afraid to use it. Both Mac and Linux depend heavily on your proficiency with the Terminal.

**Step 0:** If you are on Mac OS X, install Homebrew if you have not already done so. This is optional, but you will find that Homebrew will be useful for other projects as well, so it is recommended that you install it anyway. You can find instructions on how to install Homebrew on the [main page of the Homebrew project](https://brew.sh/). Alternatively, copy and paste this script in to your Terminal prompt, then execute it:

/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"

**Step 1:** Next, you need npm. Fortunately, it is fairly easy to obtain npm. Since npm depends on nodejs, you only need to install nodejs, and npm will be accessible. You can either [download the installer](https://nodejs.org/en/) and install nodejs by following the graphical interface of the installer package, or alternatively use Homebrew from step 0 and install in your Terminal using this script:

brew install node

If you are on Linux, you can [install node using your distro’s respective package manager](https://nodejs.org/en/download/package-manager/). For instance, for Ubuntu/Debian-based distributions, you’d execute this:

curl -sL https://deb.nodesource.com/setup\_9.x | sudo -E bash -

sudo apt-get install nodejs

To see whether you have set up npm properly, copy and paste this script into your Terminal, then execute it:

npm -v

If you are given a version number (5.6.0 at the time this guide was written), then you are good to go.

**Step 2:** Once you have npm set up, you are ready to install angular. The following scripts are applicable to both Mac and Linux. What you need is angular’s CLI (Command-Line Interface), and you can install it by executing this script:

npm install -g @angular/cli

After that script has finished running, navigate to the project’s folder/directory and execute the following script:

npm install

**Step 3:** If there is no error, you should be good to go. You can execute the following script to start up the development server on your local machine:

ng serve

Navigate your web browser to <http://localhost:4200> and you should be able to see the front-end’s interface.

**Step 4:** If you ran into issues, please check the error message on Google to see if there is any easy solution. Otherwise, please contact the team and see if we can help you fix it.

**Jersey Webservice Examples**

1. Download or clone the project at <https://github.com/shaynguyen/web-service-tool>.

2. Open the command prompt and change the directory to the project.

3. Checkout to the connection\_work branch.

4. Run the command **mvn package**.

5. Copy the .war file in the target folder to your tomcat webapps folder then restart your tomcat server.

6. The webservices created are simple math calculations. Available at localhost:8080/webservice/rest/\* by default. See <localhost:8080/webservice/rest/application.wadl> to see all endpoints.