**Spring Boot – Angular Full Stack Application**

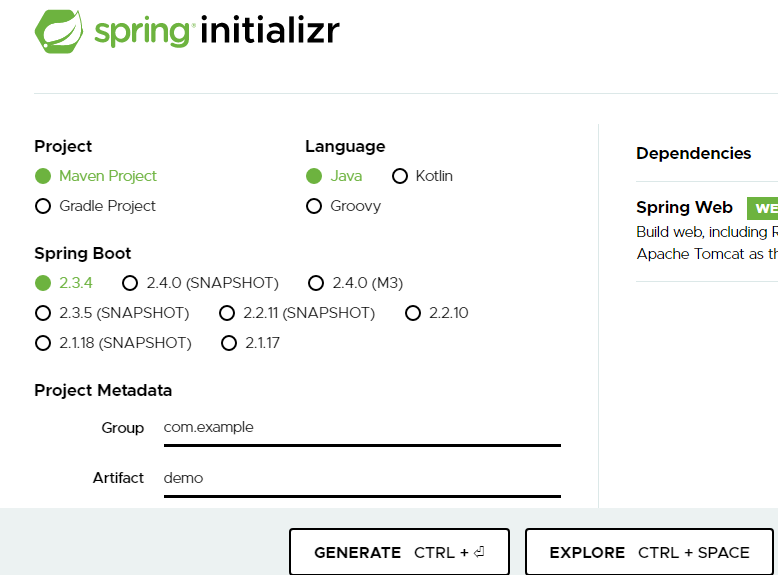
|  |  |
| --- | --- |
| **Sr No** | **Description** |
| 1 | Creating Spring Boot User Application - Backend |
| 2 | Creating Angular User Application and deploy as a WAR – Frontend |
| 3 | Create Modular Project Structure |
| 4 | Single Jenkins job to Deploy Frontend and Backend |

**1] Creating Spring Boot User Application – Backend**

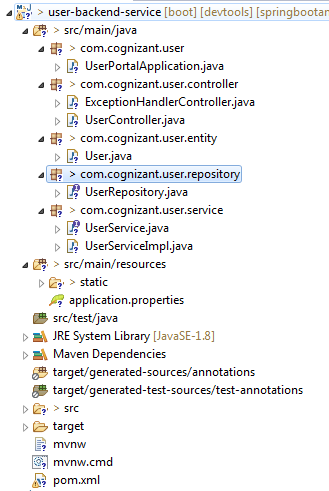
There are 4 ways using which you can create Spring Boot Application

* **Spring Maven Project** – Create a Maven Project and add your dependencies
* **Spring Starter Project Wizard** – Using STS
* **Spring Initializer** – Using start.spring.io
* **Spring Boot CLI** – Command Line Interface

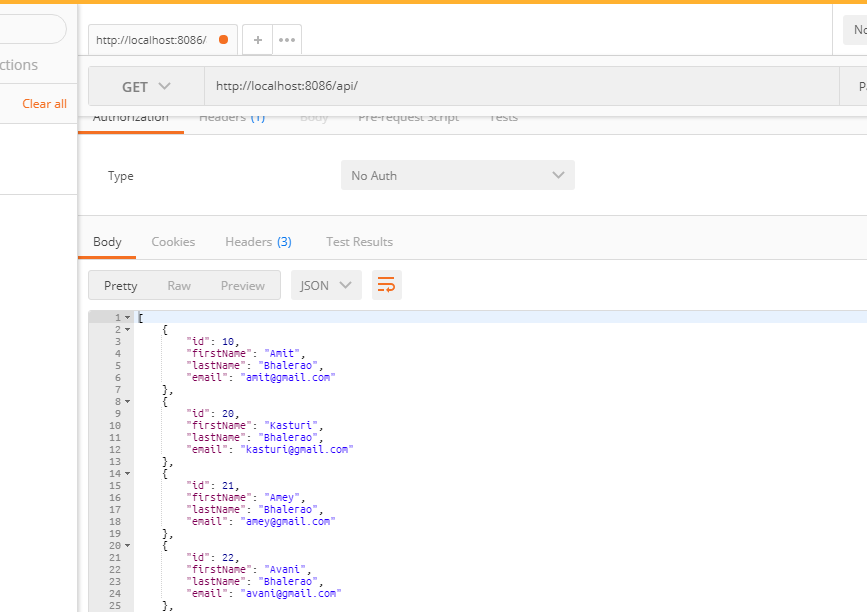
I am using Spring Initializer approach



**User Application using Spring Boot:**



Run the UserPortalApplication and check the different API’s using Postman



User Application is working as expected

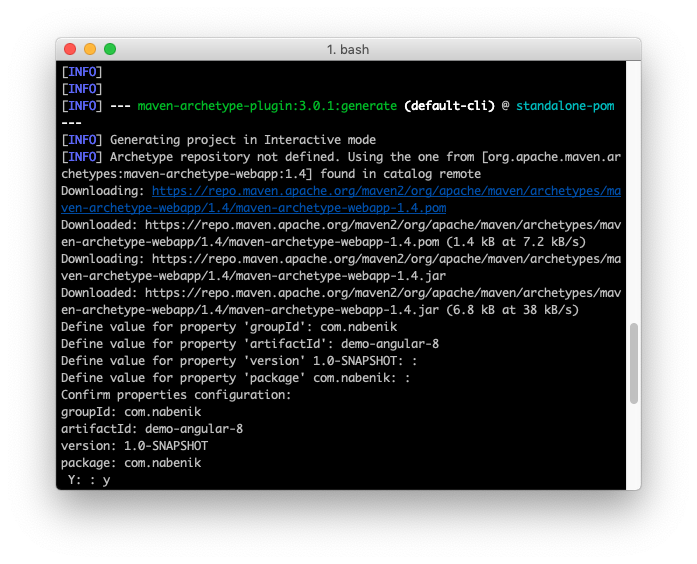
**2]** **Creating Angular User Application – Frontend**

**a) Bootstrapping a Simple Java Web Project**

Use below command to create a Maven Project :

mvn archetype:generate -DarchetypeGroupId=org.apache.maven.archetypes -DarchetypeArtifactId=maven-archetype-webapp –Darche

The interactive shell will ask you for project characteristics including groupId, artifactId (project name), and base package.

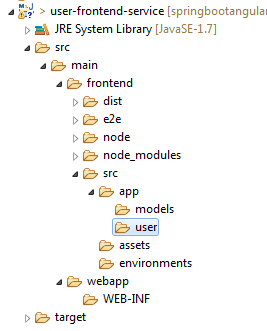


**b) Bootstrapping a Simple Angular Project**

demo-angular-8$ cd src/main/

demo-angular-8/src/main$ ng new frontend

This will bootstrap a Vanilla Angular project, and in fact, you could consider the src/main/frontend folder as a separate root (and also you could open this directly from VSCode). The final structure will be like this:



By default Angular project come with hooks from npm to ng but we need to add a hook in package.json to create a production quality build (buildProduction), please double check the base-href parameter since I'm using the default root from Java conventions (same as project name)

**Changes in package.json:**

"scripts": {

"ng": "ng",

"start": "ng serve",

"build": "ng build",

"buildProduction": "ng build --prod --base-href /user-frontend-service/",

"test": "ng test",

"lint": "ng lint",

"e2e": "ng e2e"

},

To test this build, we could execute npm run buildProduction.

**c) To Create a War of Angular Project we need change pom.xml as follows**

<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.amitb</groupId>

<artifactId>user-frontend-service</artifactId>

<version>1.0-SNAPSHOT</version>

<packaging>war</packaging>

<name>User Application Frontend</name>

<!-- FIXME change it to the project's website -->

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<maven.compiler.source>1.7</maven.compiler.source>

<maven.compiler.target>1.7</maven.compiler.target>

</properties>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.11</version>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<finalName>user-frontend-service</finalName>

<plugins>

<plugin>

<groupId>com.github.eirslett</groupId>

<artifactId>frontend-maven-plugin</artifactId>

<version>1.6</version>

<configuration>

<workingDirectory>src/main/frontend</workingDirectory>

</configuration>

<executions>

<execution>

<id>install-node-and-npm</id>

<goals>

<goal>install-node-and-npm</goal>

</goals>

<configuration>

<nodeVersion>v10.16.1</nodeVersion>

</configuration>

</execution>

<execution>

<id>npm install</id>

<goals>

<goal>npm</goal>

</goals>

<configuration>

<arguments>install</arguments>

</configuration>

</execution>

<execution>

<id>npm build</id>

<goals>

<goal>npm</goal>

</goals>

<configuration>

<arguments>run buildProduction</arguments>

</configuration>

<phase>generate-resources</phase>

</execution>

</executions>

</plugin>

<plugin>

<artifactId>maven-war-plugin</artifactId>

<version>3.2.2</version>

<configuration>

<failOnMissingWebXml>false</failOnMissingWebXml>

<!-- Add frontend folder to war package -->

<webResources>

<resource>

<directory>src/main/frontend/dist/frontend</directory>

</resource>

</webResources>

</configuration>

</plugin>

</plugins>

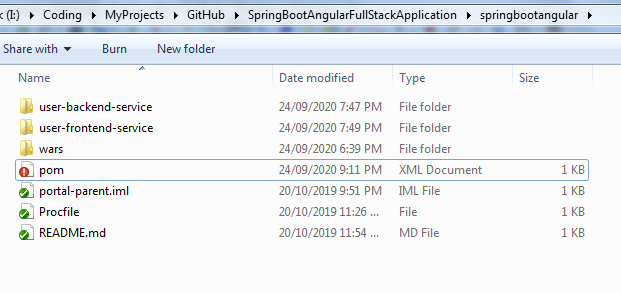
</build>

</project>

**d) To create a deployable war use : mvn clean install**

**3] Create Modular Project Structure**

1] Create a Modular Project Structure as follows



2] Create a pom.xml with below contents in it

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.devglan</groupId>

<artifactId>portal-parent</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>pom</packaging>

<name>Spring Boot Angular Full Stack Application</name>

<description>Spring Boot Angular Full Stack Application</description>

<modules>

<module>user-backend-service</module>

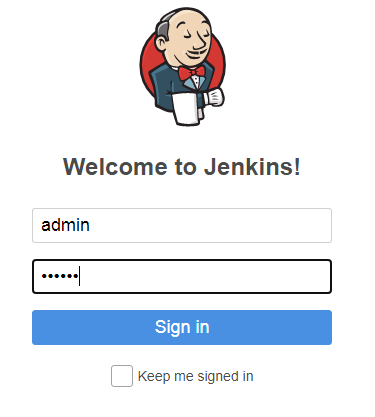
<module>user-frontend-service</module>

</modules>

</project>

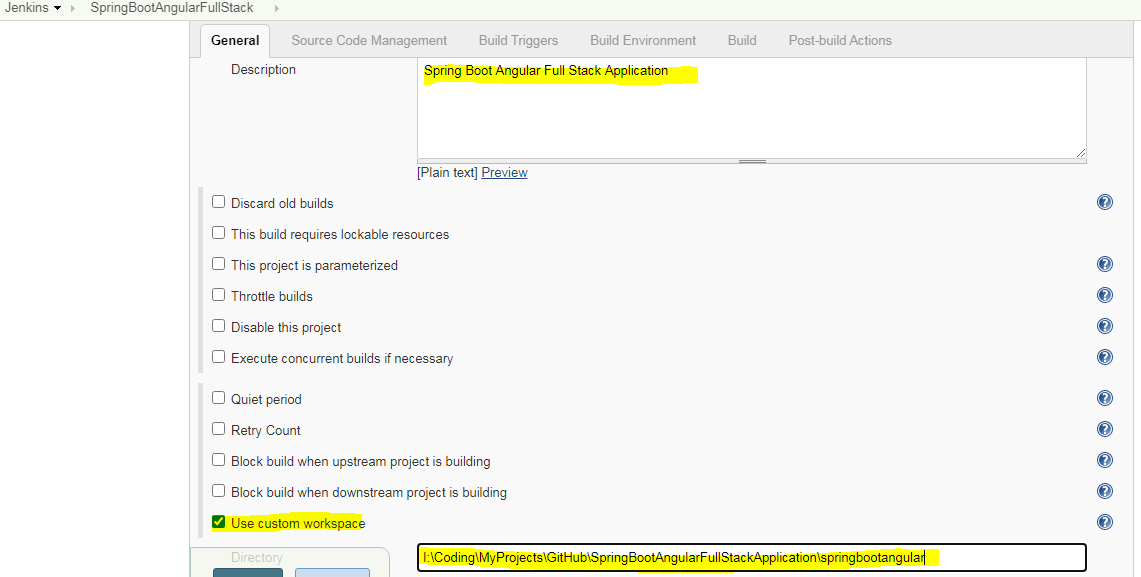
**4]** **Single Jenkins job to Deploy Frontend and Backend**

1] Login to Jenkins

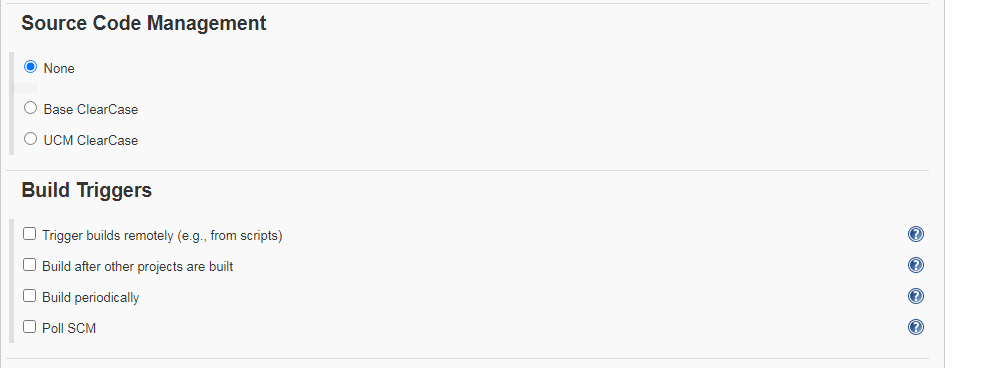


2] Click on New Item > Enter Job Name > FreeStyle > OK

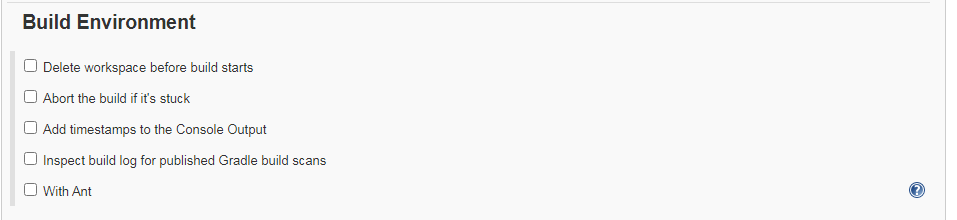
* **General:**



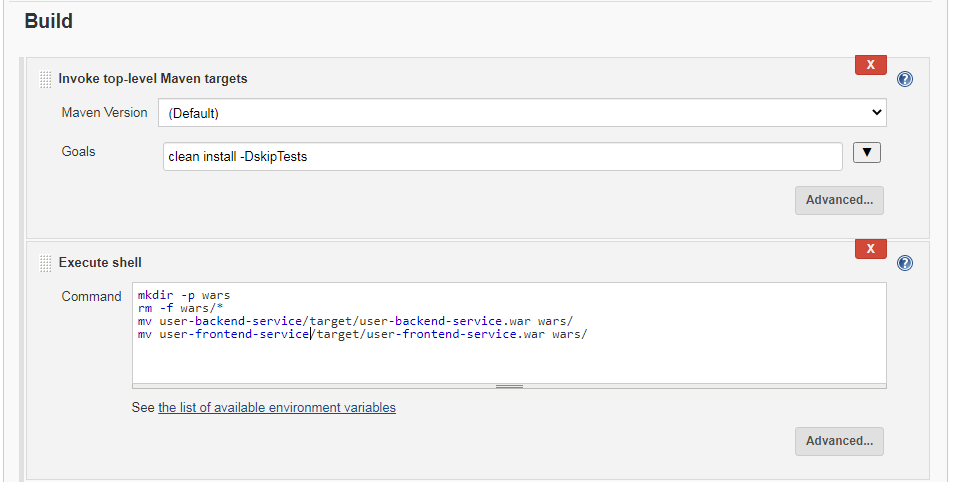
* **Source Code Management and Build Triggers**



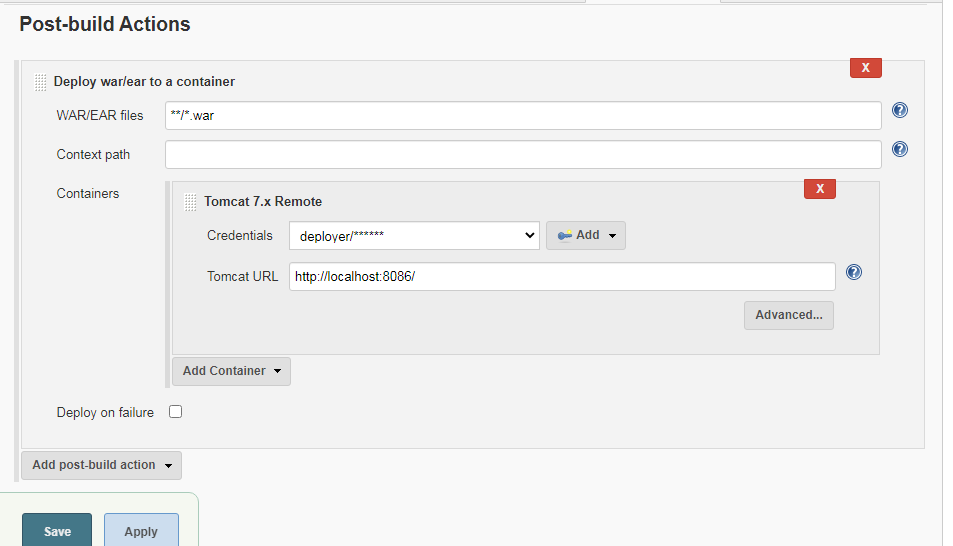
* **Build Environment**



* **Build**



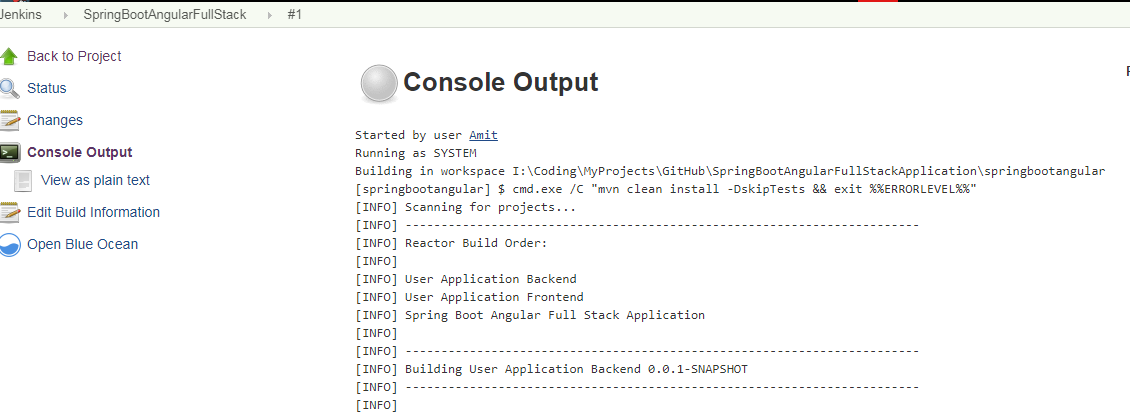
* **Post-Build Actions**

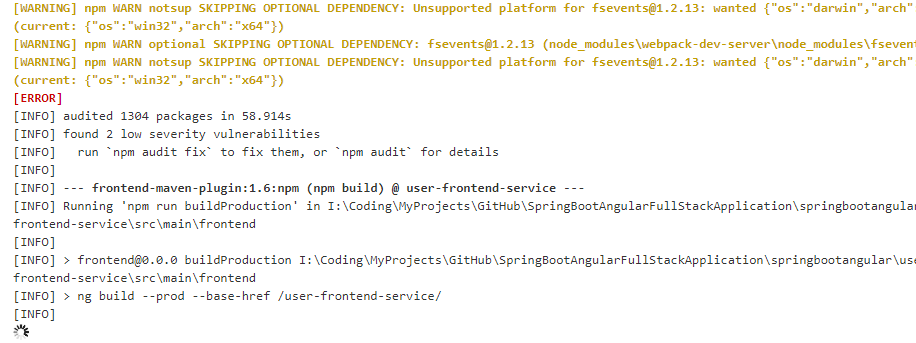


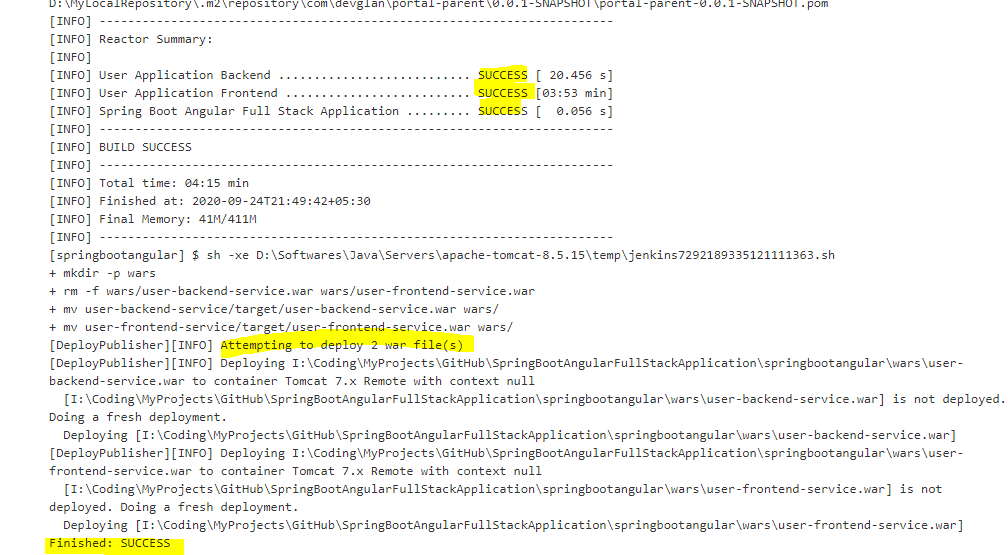
**Click on Save and Apply**

**Note:** Make sure your tomcat is started on running on port 8086 as per above configuration

3] Click on **Build Now**



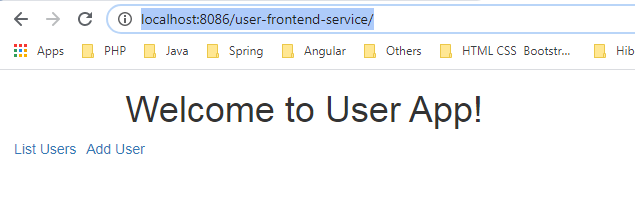




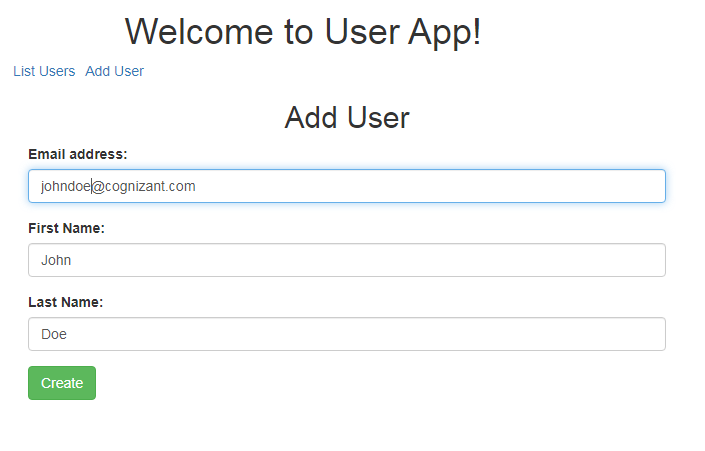
**Congratulations! Your Full Stack Spring Boot – Angular is deployed to Tomcat 7.0**

4] Open your favorite browser and hit below URL

1. <http://localhost:8086/user-frontend-service/>



1. Click on Add User



1. Click on List Users

