

Create IntelliJ IDEA Spring Project

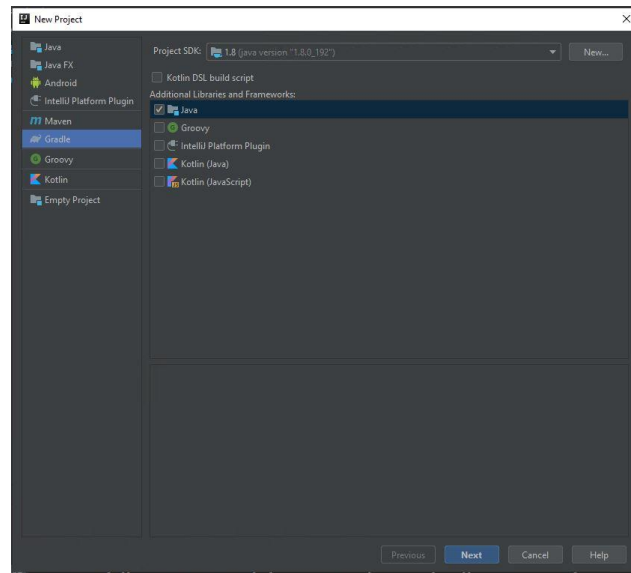
We need to download IntelliJ IDEA community edition from the official page.

(<https://www.jetbrains.com/idea/download/#section=windows>)

after it is installed correctly we can create a java gradle project

Steps:

- Open IntelliJ IDEA
- Go to File -> New -> Project



New Project

GroupId:

ArtifactId:

Version:

Previous Next Cancel Help

New Project

☒ Use auto-import

Group modules: ☐ using explicit module groups ☒ using qualified names

☒ Create separate module per source set

☒ Use default Gradle wrapper (recommended)

☐ Use local Gradle distribution

Gradle home:

Gradle JVM:

Previous Next Cancel Help

New Project

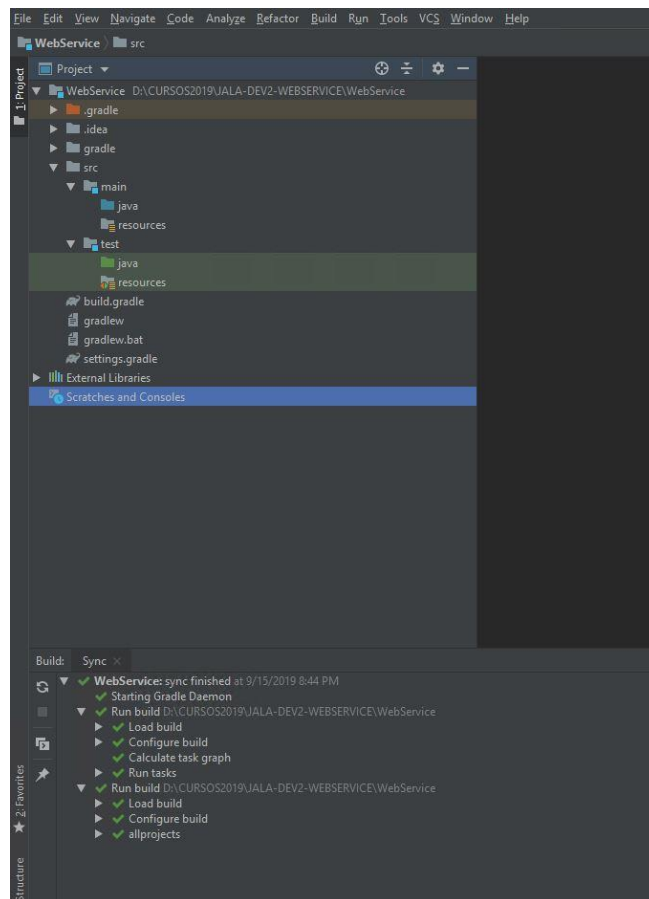
Project name:

Project location:

More Settings

Previous Finish Cancel Help

- Verify that the project was created with the following structure and the build was executed without errors



- Modify build.gradle file to could work with spring

build.gradle

```

1  plugins {
2      id 'java'
3  }
4
5  group 'WebService'
6  version '1.0-SNAPSHOT'
7
8  sourceCompatibility = 1.8
9
10 repositories {
11     mavenCentral()
12 }
13
14 dependencies {
15     testCompile group: 'junit', name: 'junit', version: '4.12'
16 }
17

```

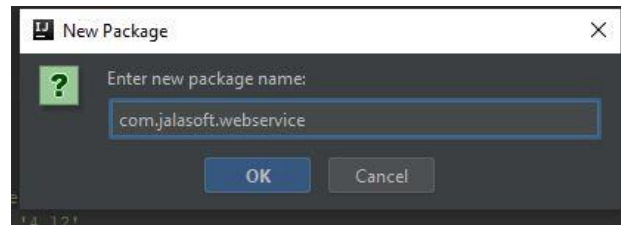
build.gradle with spring dependencies

```

1  buildscript {
2      repositories {
3          mavenCentral()
4      }
5      dependencies {
6          classpath("org.springframework.boot:spring-boot-gradle-plugin:2.1.6.RELEASE")
7      }
8  }
9
10 plugins {
11     id 'java'
12 }
13
14 apply plugin: 'org.springframework.boot'
15 apply plugin: 'io.spring.dependency-management'
16
17 group 'WebService'
18 version '1.0-SNAPSHOT'
19
20 sourceCompatibility = 1.8
21
22 repositories {
23     mavenCentral()
24 }
25
26 dependencies {
27     compile("org.springframework.boot:spring-boot-starter-web")
28     testCompile group: 'junit', name: 'junit', version: '4.12'
29 }

```

- Create packages



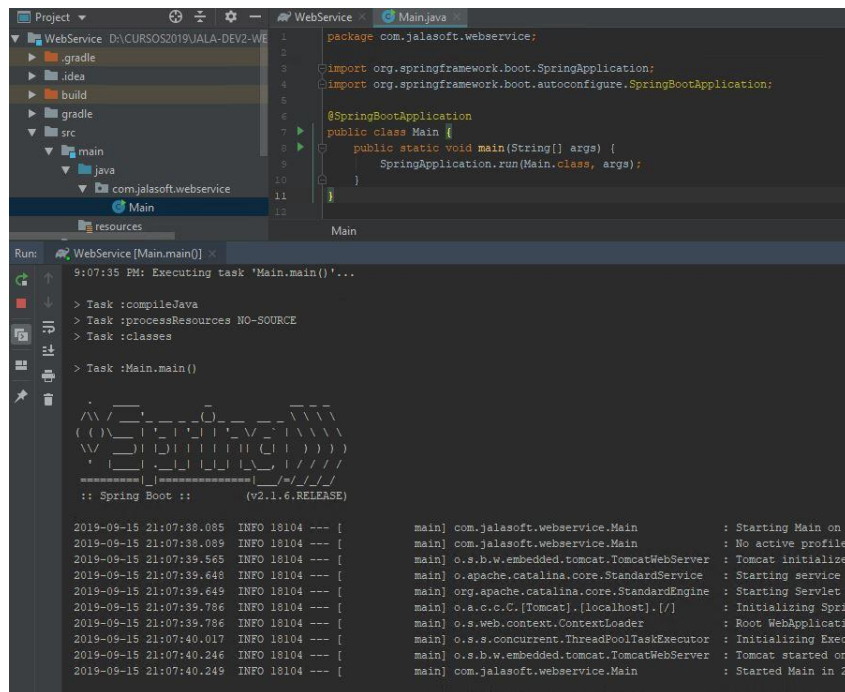
- Create Main.java file

```

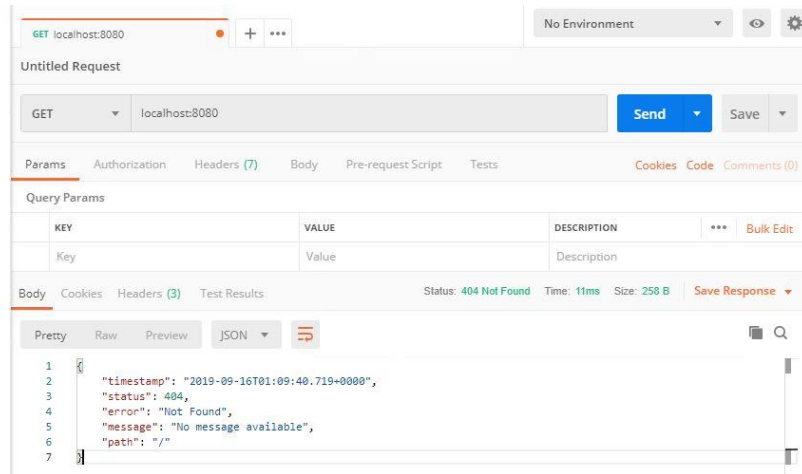
1 package com.jalasoft.webservice;
2
3 public class Main {
4     public static void main(String[] args) {
5         System.out.println("test");
6     }
7 }

```

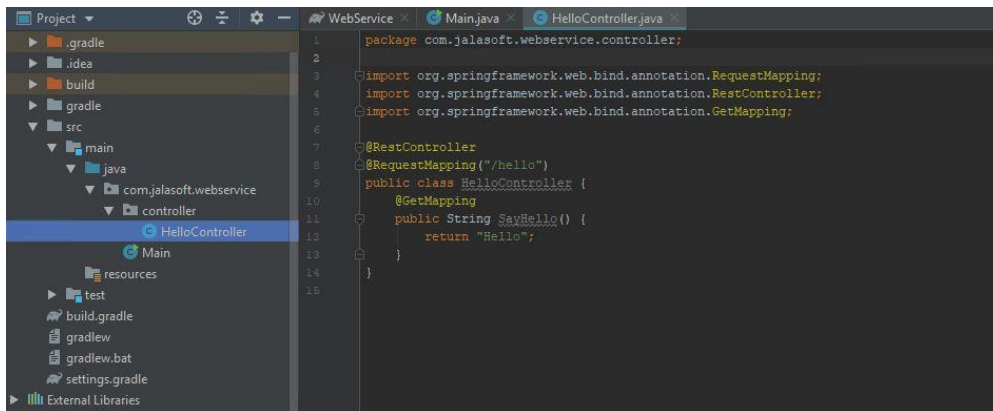
- Add spring annotations on Main class



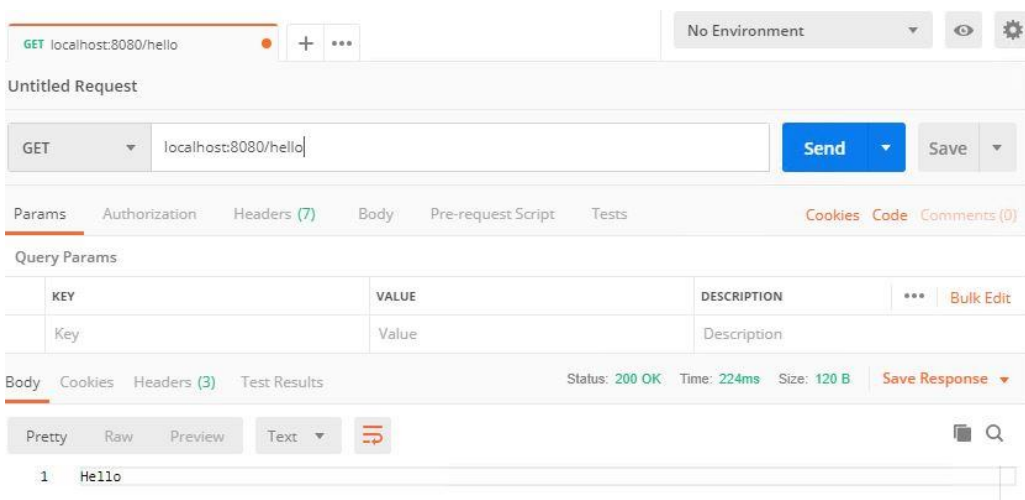
- Go to postman and verify if the web service is working.



- Create the first endpoint (localhost:8080/hello) for it, we need to add a new controller class and mapping it.



- Go to postman and verify if the web service is working (use the new endpoint).



- Create new endpoint to get parameters from client

```

1  import org.springframework.web.bind.annotation.RequestMapping;
2  import org.springframework.web.bind.annotation.RestController;
3  import org.springframework.web.bind.annotation.GetMapping;
4  import org.springframework.web.bind.annotation.RequestParam;
5
6  @RestController
7  @RequestMapping("/hello")
8  public class HelloController {
9      @GetMapping
10     public String SayHello() {
11         return "Hello";
12     }
13
14     @RequestMapping("/convert")
15     @GetMapping
16     public String SayHello(
17         @RequestParam(value="name", defaultValue="") String name,
18         @RequestParam(value="lastName", defaultValue="") String lastName) {
19         StringBuilder firstResponse = new StringBuilder();
20         firstResponse.append("Hi ").append(name).append(" ").append(lastName);
21         return firstResponse.toString();
22     }
23 }

```

- Go to postman and verify if the web service is working (use the new endpoint).

