

SpringBoot 实战 (二) | 第一个 SpringBoot 工程详解

作者: [JavaFish](#)

原文链接: <https://ld246.com/article/1620986110993>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

微信公众号：一个优秀的废人

如有问题或建议，请后台留言，我会尽力解决你的问题。

前言

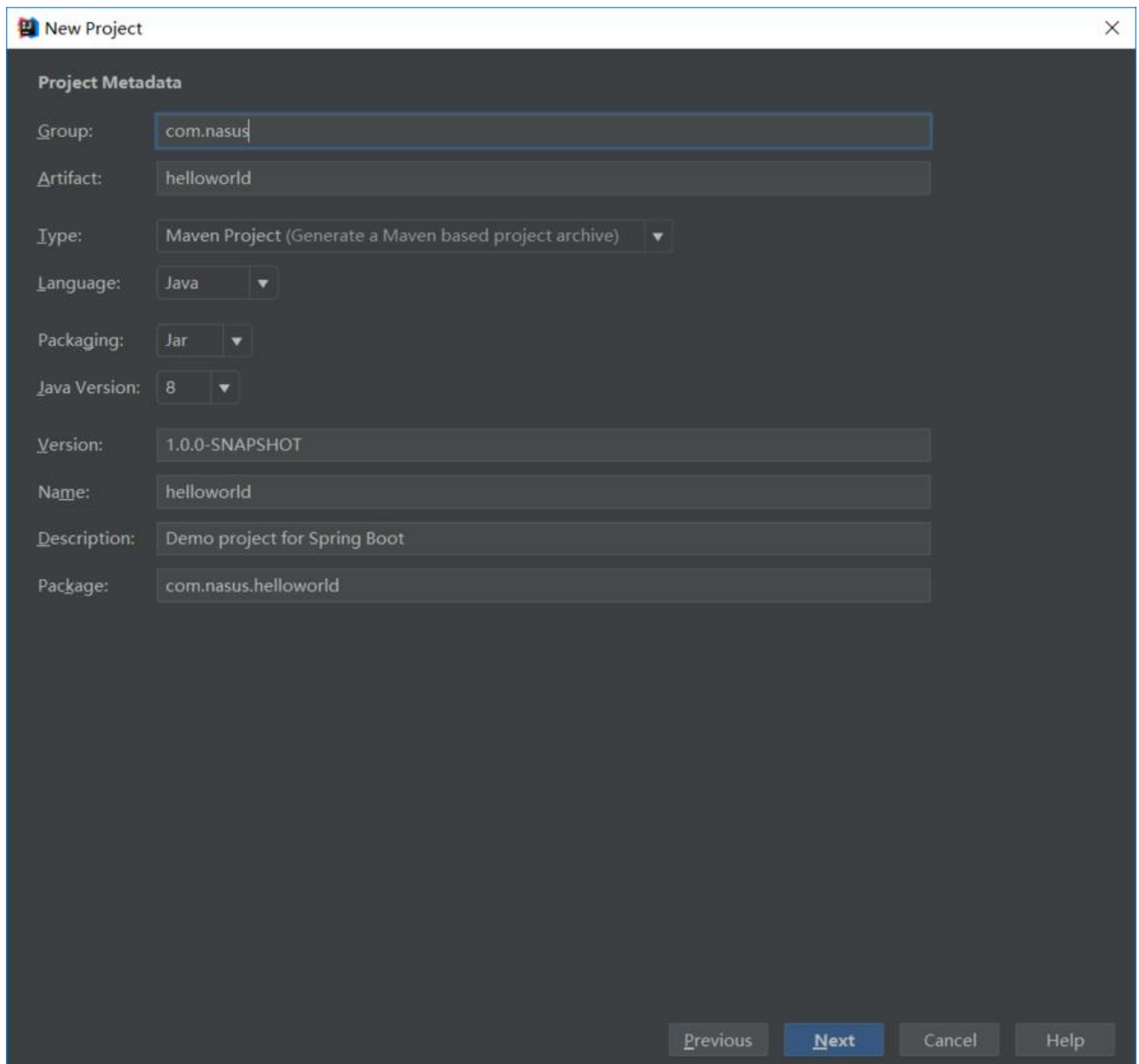
哎呦喂，按照以往的惯例今天周六我的安排应该是待在家学学猫叫啥的。但是今年这种日子就可能不复返了，没办法啊。前几天年轻，立下了一周至少更两篇文章的 flag 。废话少说，今天接着前给你们带来了第一个 SpringBoot 工程的详解。

第一个 SpringBoot 工程

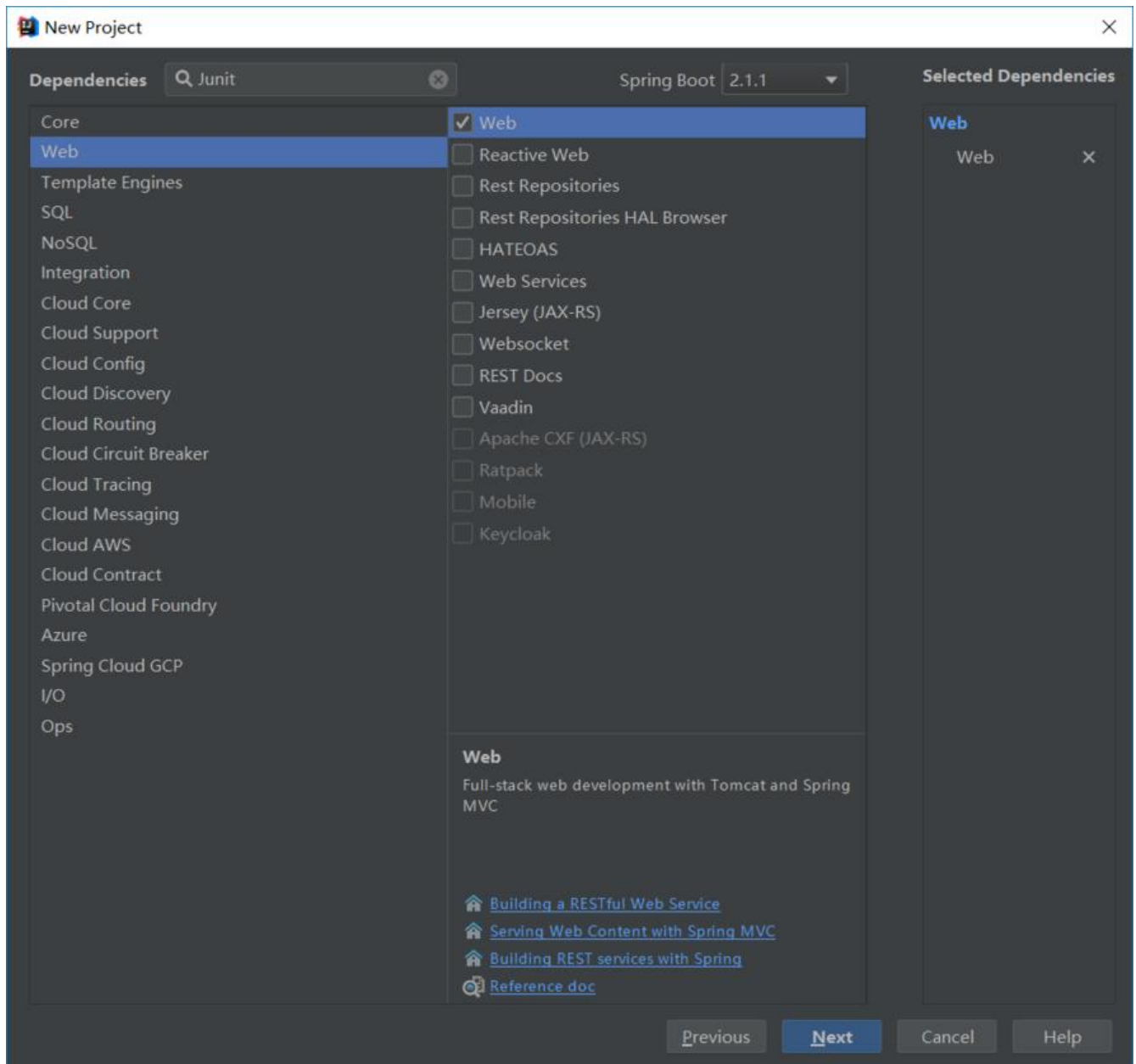
前文已经说过了 SpringBoot 工程的创建，这里不再赘述，还不会的朋友，请看下面这篇文章。

[如何使用 IDEA 构建 Spring Boot 工程](#)

学过编程的都知道，学习一门新语言的第一个项目肯定是 Hello World 。那这里也不例外，我们先创建一个非常简单的 Hello World 工程。给大家讲解 SpringBoot 的项目目录。创建信息如下：



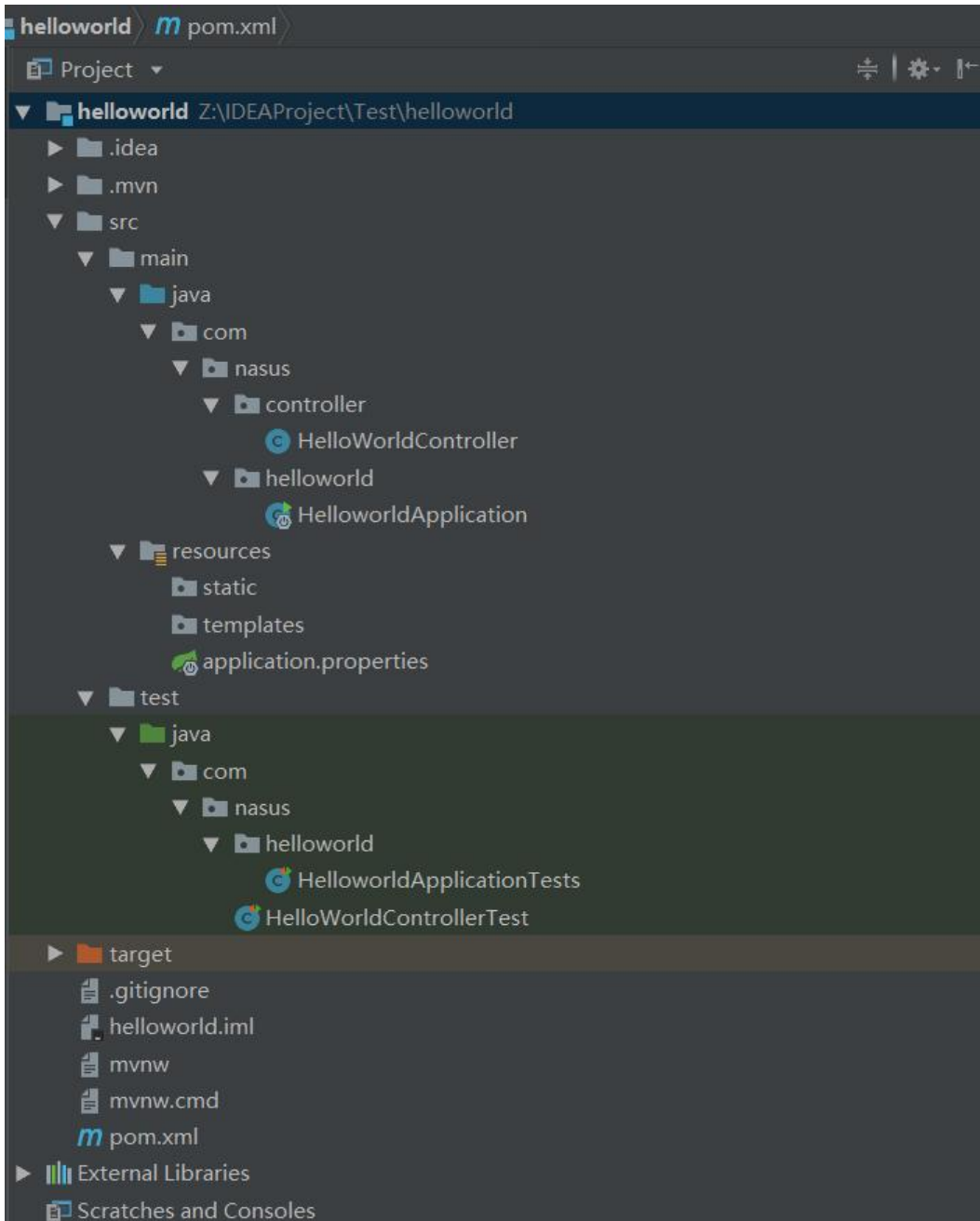
由于本文重点旨在讲解 SpringBoot 的项目目录。所以选择的依赖包非常简单，就选择 Web 足矣。



SpringBoot 项目目录详解

创建成功之后的 SpringBoot 项目目录如下，以下对各主要目录的作用进行讲解：

- src 是整个工程的根目录，基本上做 web 开发你的代码大部分都放在这里。其中 main 目录下放置是你的 Java 代码；resource 目录，顾名思义就是放置配置文件、静态资源(static)以及前端模板(template)。
- test 目录就是放置你的单元测试代码。
- target 就是项目编译生成的目录，里面包含代码编译后的 class 文件以及一些静态资源和配置文件。
- External Libraries 这里放置了，pom.xml 导入的依赖包。
- 其他没提到的目录都是不重要的。



由上图项目目录，可以看到有几个文件，这些文件有些是我新建的，有些是项目生成的。下面我会讲：

- pom.xml 这个文件是整个项目最重要的文件，它包含了整个项目的依赖包。Maven 会根据这个文导入相关的我们开发需要的依赖包。代码如下：

可以看到 pom.xml 中一共有 4 个依赖，其中只有 Junit 是我手动加入的，用于单元测试。

其他的如 Spring Boot 启动父依赖、Spring Boot web 依赖、Spring Boot web test 依赖都是创建目时，勾选 web 选项卡而生成的。这几个依赖包的额作用就是 内嵌 Tomcat 容器，集成各 Spring 件。比如 如果没有依赖 web 包，Spring 的两大核心功能 IOC 和 AOP 将不会生效。

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/
MLSchema-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.nasus</groupId>
  <artifactId>helloworld</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>helloworld</name>
  <description>Demo project for Spring Boot</description>

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

  <!-- Spring Boot 启动父依赖 -->
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.1.1.RELEASE</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>

  <dependencies>
    <!-- Spring Boot web依赖 -->
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-web</artifactId>
    </dependency>

    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-test</artifactId>
      <scope>test</scope>
    </dependency>

    <!-- Junit -->
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.12</version>
    </dependency>
  </dependencies>

  <build>
    <plugins>
      <plugin>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-maven-plugin</artifactId>
      </plugin>
    </plugins>
  </build>
</project>

```

- HelloworldApplication.java 最为重要，它由项目生成，是整个工程的应用启动类 main 函数。代

由项目生成，代码如下：SpringApplication 引导应用，并将 HelloWorldApplication 本身作为参数递给 run 方法。具体 run 方法会启动嵌入式的 Tomcat 并初始化 Spring 环境及其各 Spring 组件。

需要注意的是，这个类必须放在其他类的上册目录，拿上述目录举个栗子，若其他 HelloWorldController.java 类位于 com.nasus.controller 下。则 HelloWorldApplication.java 类必须放置在 com.nasus 下或者 com 下（层级必须大于其他 java 类）。否则启动项目访问会报 Whitelabel Error Page 错误原因是项目扫描不到 @RestController、@RequestMapping 等注解配置的方法和类。



Whitelabel Error Page

This application has no explicit mapping for /error, so you are seeing this as a fallback.

Sat Jan 05 15:04:38 CST 2019

There was an unexpected error (type=Not Found, status=404).

No message available

```
package com.nasus.helloworld;
```

```
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
```

```
@SpringBootApplication
public class HelloWorldApplication {
    public static void main(String[] args) {
        SpringApplication.run(HelloWorldApplication.class, args);
    }
}
```

• HelloWorldController 是我手动编写的，代码如下：@RestController 和 @RequestMapping 解是来自 SpringMVC 的注解，它们不是 SpringBoot 的特定部分。

其中 @RestController 注解的作用是：提供实现了 REST API，可以服务 JSON、XML 或者其他。这是以 String 的形式渲染出结果。

其中 @RestController 注解的作用是：提供路由信息，"/" 路径的 HTTP Request 都会被映射到 sayHello 方法进行处理。

具体参考，Spring 官方的文档《[Spring Framework Document](#)》

```
package com.nasus.controller;
```

```
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
```

```
/**
 * Project Name:helloworld <br/>
 * Package Name:com.nasus.controller <br/>
 * Date:2019/1/5 13:59 <br/>
 * <b>Description:</b> TODO: 描述该类的作用 <br/>
```

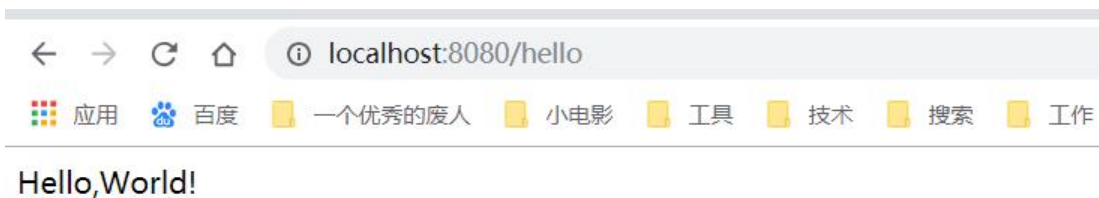
```

*
* @author <a href="mailto:turodog@foxmail.com">nasus</a> <br/>
* Copyright Notice =====
=====
* This file contains proprietary information of Eastcom Technologies Co. Ltd.
* Copying or reproduction without prior written approval is prohibited.
* Copyright (c) 2019 =====
=====
*/
@RestController
public class HelloWorldController {

    @RequestMapping("/hello")
    public String sayHello() {
        return "Hello,World!";
    }
}
}

```

写完 Controller 层的代码，我们就可以启动此项目。点击 IDEA 项目启动按钮，效果如下：



● 好的程序必须配备完善的单元测试。HelloWorldControllerTest.java 文件是由我编写的主要作用是测试 HelloWorldController.java 中的方法。这里用的是 Junit 依赖包进行单元测试，代码如下：里的逻辑就是测试 HelloWorldController.java 的 sayHello 方法输出的字符是否是 Hello,World!

```

package com.nasus;

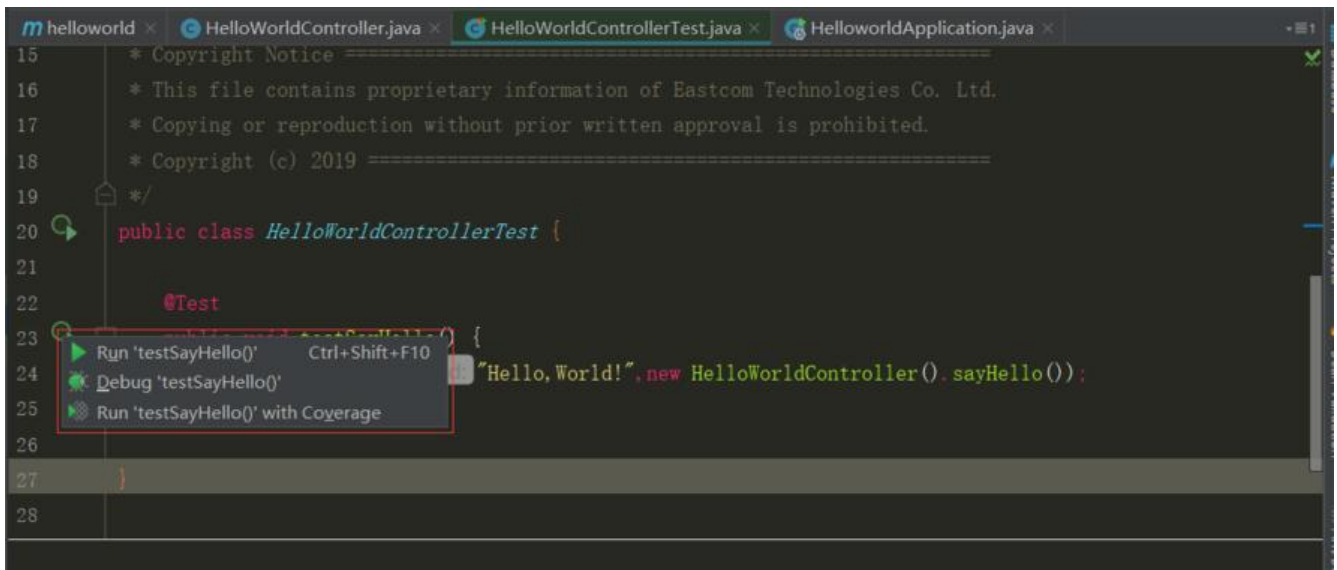
import static org.junit.Assert.assertEquals;
import com.nasus.controller.HelloWorldController;
import org.junit.Test;

/**
 * Project Name:helloworld <br/>
 * Package Name:com.nasus <br/>
 * Date:2019/1/5 14:01 <br/>
 * <b>Description:</b> TODO: 描述该类的作用 <br/>
 *
 * @author <a href="mailto:turodog@foxmail.com">nasus</a> <br/>
 * Copyright Notice =====
=====
* This file contains proprietary information of Eastcom Technologies Co. Ltd.
* Copying or reproduction without prior written approval is prohibited.
* Copyright (c) 2019 =====
=====

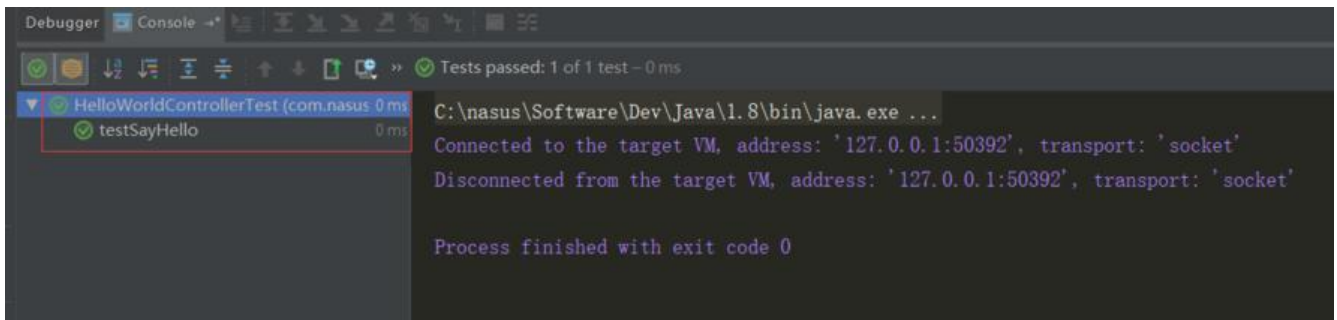
```

```
=====  
*/  
public class HelloWorldControllerTest {  
  
    @Test  
    public void testSayHello() {  
        assertEquals("Hello,World!",new HelloWorldController().sayHello());  
    }  
  
}
```

编写完成之后，可以通过以下按钮启动单元测试类。



测试结果如下：可以看到红圈框住的地方，出现这个绿色标志证明单元测试没问题。sayhello 方法的结果是对的。



后语

我为什么要写这种这么简单的教程？

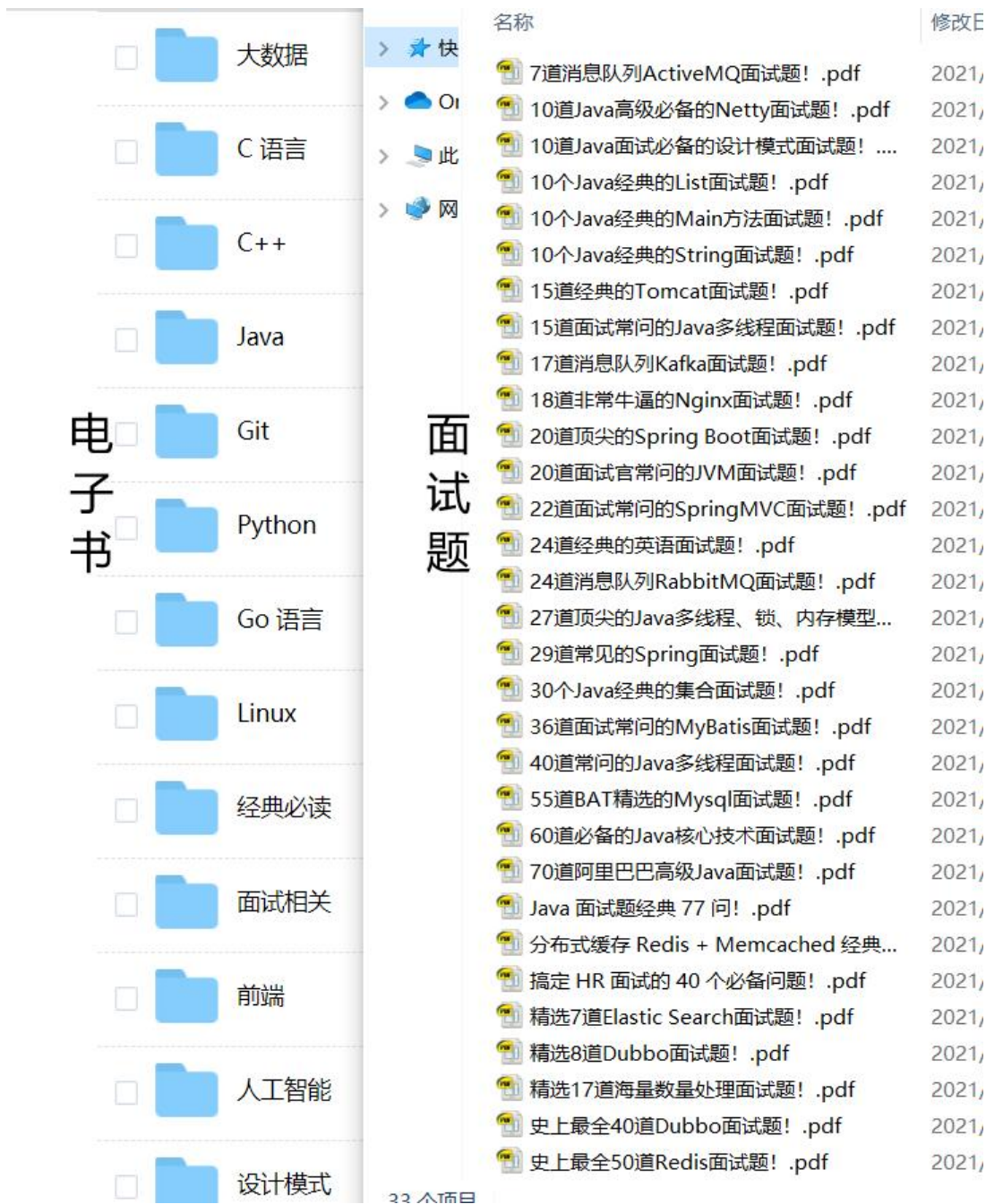
是这样的，我始终认为比我聪明的人有很多，但比我笨的人也不少。在中国有很多你认为众所周知的，其实有一车人根本不知道，这篇文章哪怕只帮助到一个人，足矣。

之后我打算出一个 SpringBoot 系列的教程，敬请关注与指正，本人也是一个小菜鸟在打怪升级中，本文有不正确的地方，烦请指正。一起学习一起进步。

如果看到这里，喜欢这篇文章的话，请帮点个好看。

初次见面，也不知道送你们啥。干脆就送**几百本电子书**和**2021最新面试资料**吧。微信搜索**JavaFish**复**电子书**送你 1000+ 本编程电子书；回复**面试**获取 50 套大厂面试题；回复**1024**送你一套完整的 jav 视频教程。

面试题都是有答案的，如下所示：有需要的就来拿吧，**绝对免费，无套路获取**。



The image shows a file explorer window with a sidebar on the left and a main pane on the right. The sidebar contains a list of folders: 大数据, C 语言, C++, Java, Git, Python, Go 语言, Linux, 经典必读, 面试相关, 前端, 人工智能, 设计模式. The main pane shows a folder named '面试题' (Interview Questions) containing a list of PDF files. The files are listed with their names and modification dates. The list includes:

名称	修改E
7道消息队列ActiveMQ面试题! .pdf	2021,
10道Java高级必备的Netty面试题! .pdf	2021,
10道Java面试必备的设计模式面试题!	2021,
10个Java经典的List面试题! .pdf	2021,
10个Java经典的Main方法面试题! .pdf	2021,
10个Java经典的String面试题! .pdf	2021,
15道经典的Tomcat面试题! .pdf	2021,
15道面试常问的Java多线程面试题! .pdf	2021,
17道消息队列Kafka面试题! .pdf	2021,
18道非常牛逼的Nginx面试题! .pdf	2021,
20道顶尖的Spring Boot面试题! .pdf	2021,
20道面试官常问的JVM面试题! .pdf	2021,
22道面试常问的SpringMVC面试题! .pdf	2021,
24道经典的英语面试题! .pdf	2021,
24道消息队列RabbitMQ面试题! .pdf	2021,
27道顶尖的Java多线程、锁、内存模型...	2021,
29道常见的Spring面试题! .pdf	2021,
30个Java经典的集合面试题! .pdf	2021,
36道面试常问的MyBatis面试题! .pdf	2021,
40道常问的Java多线程面试题! .pdf	2021,
55道BAT精选的Mysql面试题! .pdf	2021,
60道必备的Java核心技术面试题! .pdf	2021,
70道阿里巴巴高级Java面试题! .pdf	2021,
Java 面试题经典 77 问! .pdf	2021,
分布式缓存 Redis + Memcached 经典...	2021,
搞定 HR 面试的 40 个必备问题! .pdf	2021,
精选7道Elastic Search面试题! .pdf	2021,
精选8道Dubbo面试题! .pdf	2021,
精选17道海量数量处理面试题! .pdf	2021,
史上最全40道Dubbo面试题! .pdf	2021,
史上最全50道Redis面试题! .pdf	2021,



微信搜一搜

Q JavaFish