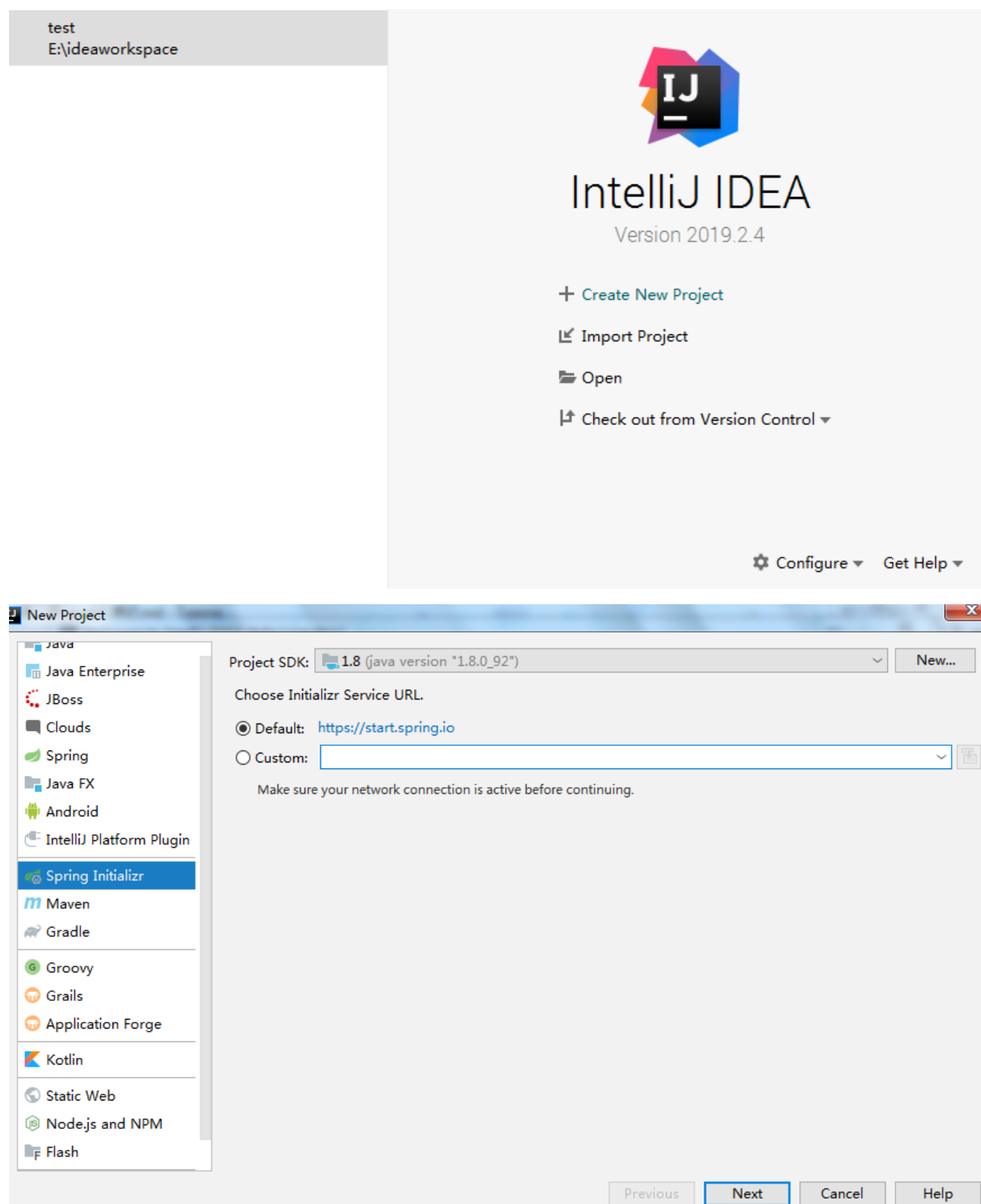
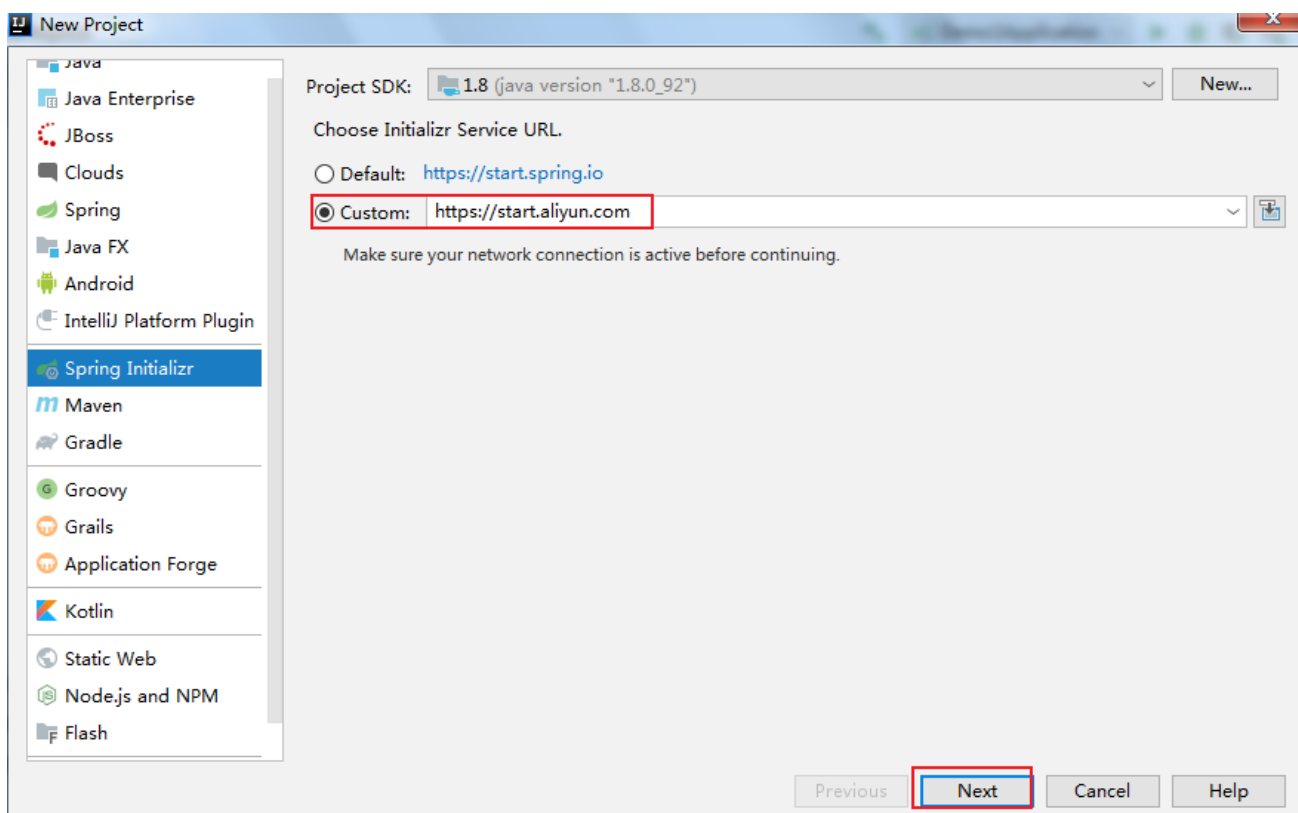


第二周 JavaEE 笔记

使用SpringInitializr方式构建SpringBoot项目



改为从阿里云服务器创建springboot项目



The 'New Project' dialog in IntelliJ IDEA. The left sidebar shows various project types, with 'Spring Initializr' selected. The main area shows 'Project SDK' set to '1.8 (java version "1.8.0_92")'. Under 'Choose Initializr Service URL', the 'Custom' option is selected with the URL 'https://start.aliyun.com'. A note below states 'Make sure your network connection is active before continuing.' At the bottom, the 'Next' button is highlighted.

Project SDK: 1.8 (java version "1.8.0_92") New...

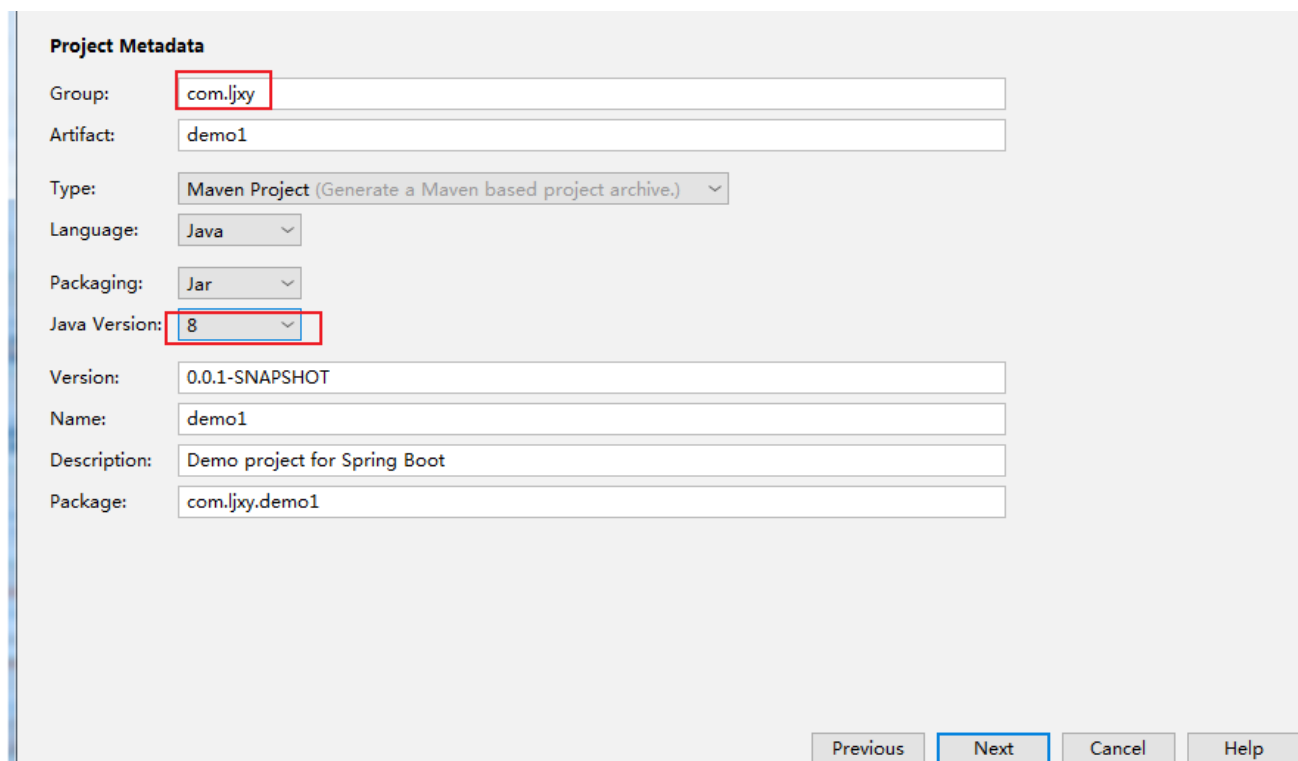
Choose Initializr Service URL.

☐ Default: <https://start.spring.io>

☒ Custom: <https://start.aliyun.com>

Make sure your network connection is active before continuing.

Previous Next Cancel Help



The 'Project Metadata' dialog in IntelliJ IDEA. It contains several input fields for project configuration. The 'Group' field is 'com.ljxy', 'Artifact' is 'demo1', 'Type' is 'Maven Project (Generate a Maven based project archive.)', 'Language' is 'Java', 'Packaging' is 'Jar', and 'Java Version' is '8'. Other fields include 'Version' (0.0.1-SNAPSHOT), 'Name' (demo1), 'Description' (Demo project for Spring Boot), and 'Package' (com.ljxy.demo1). The 'Next' button at the bottom is highlighted.

Project Metadata

Group: com.ljxy

Artifact: demo1

Type: Maven Project (Generate a Maven based project archive.)

Language: Java

Packaging: Jar

Java Version: 8

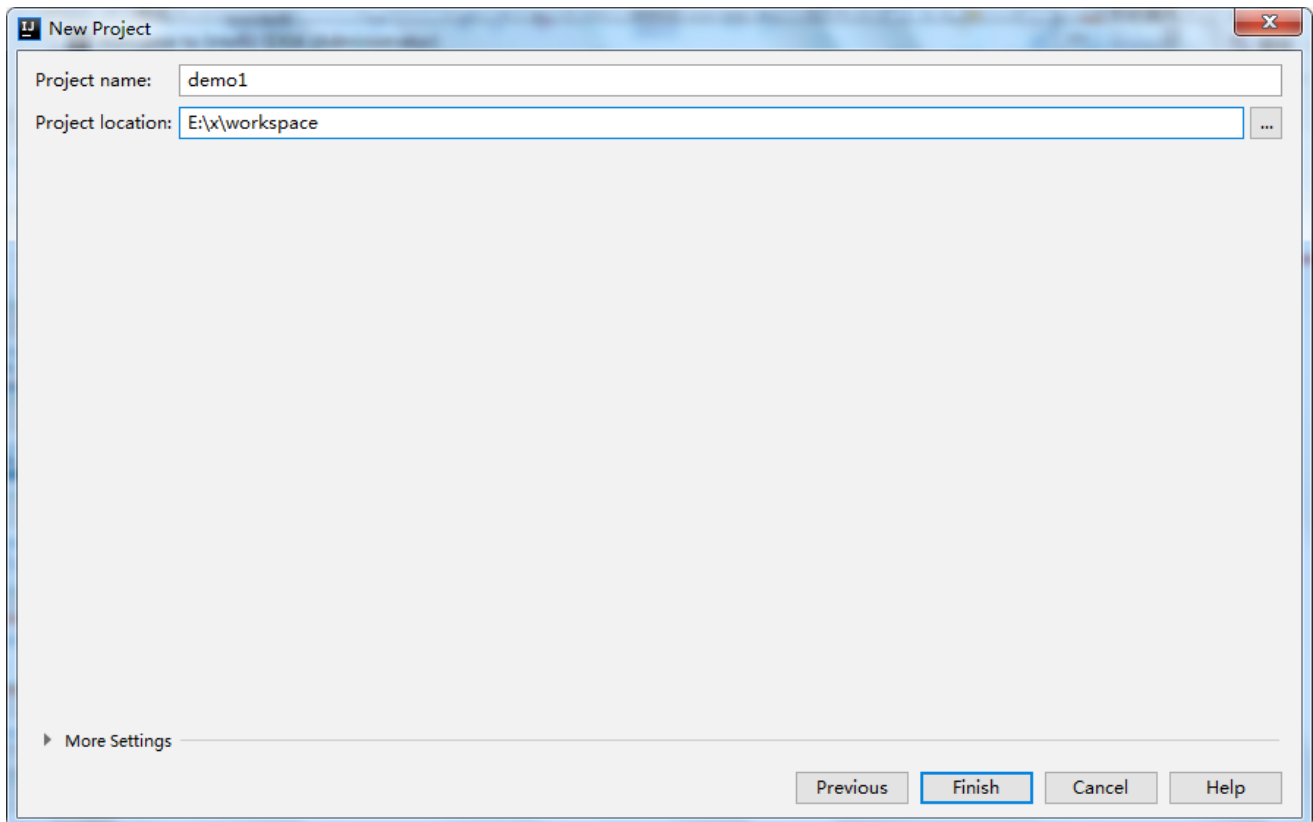
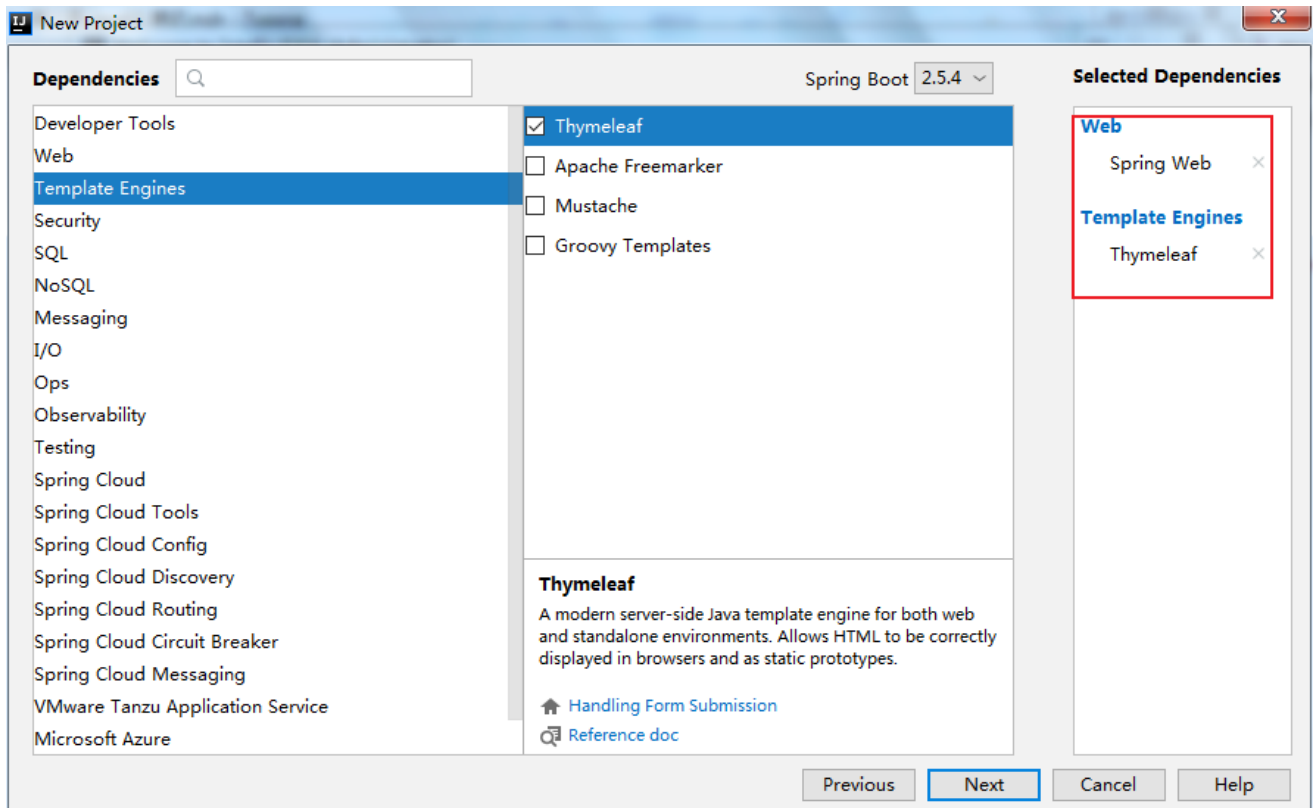
Version: 0.0.1-SNAPSHOT

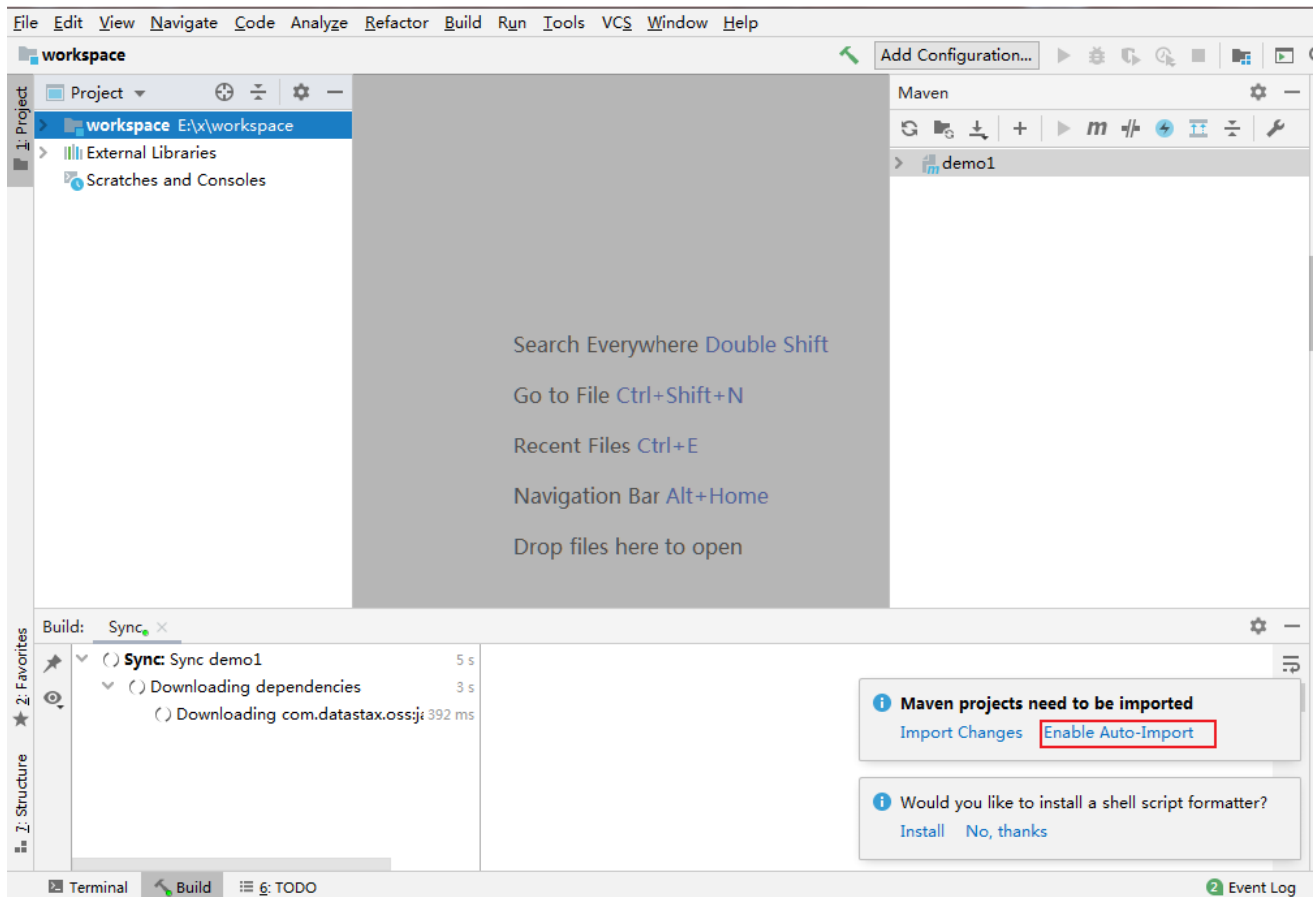
Name: demo1

Description: Demo project for Spring Boot

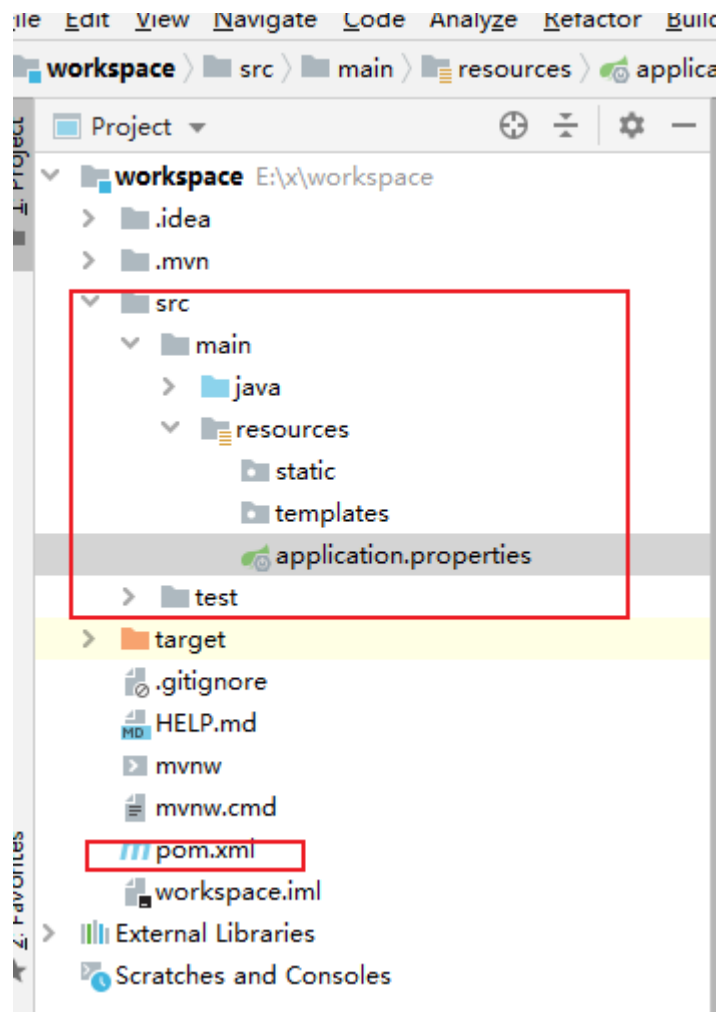
Package: com.ljxy.demo1

Previous Next Cancel Help

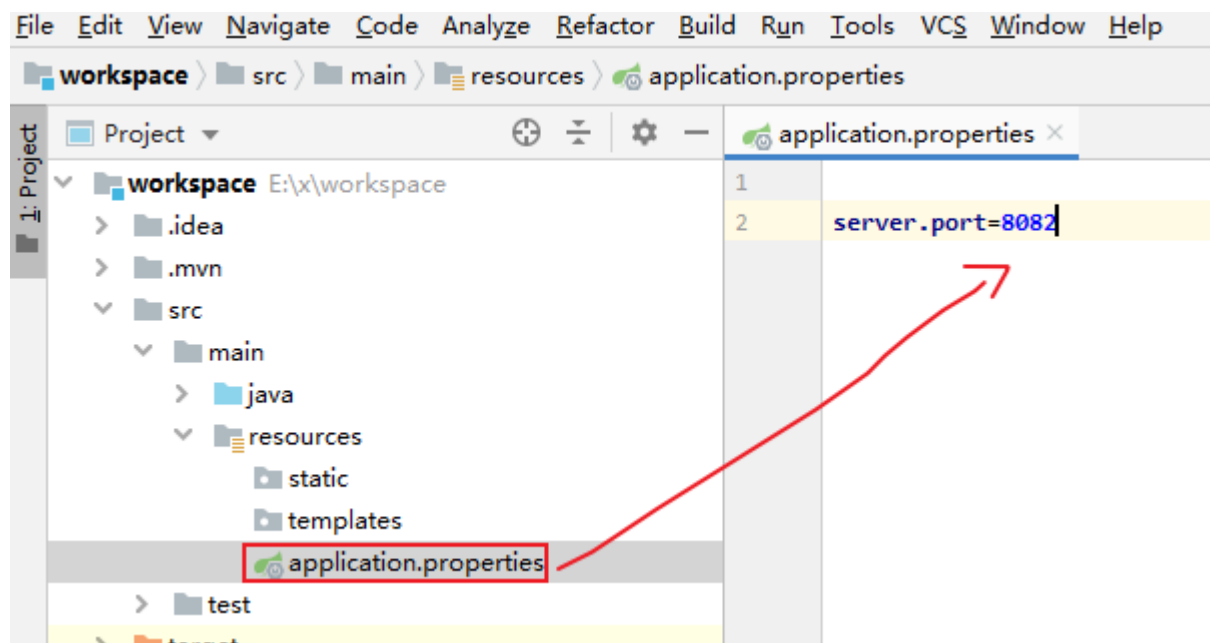




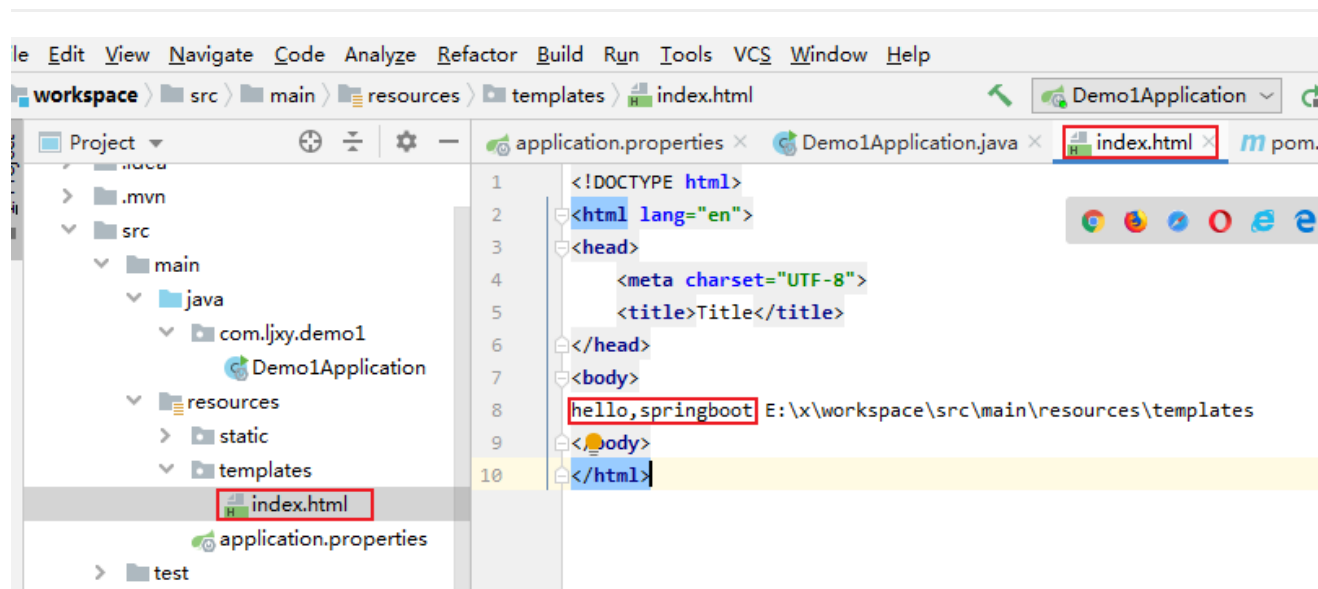
创建好的工程基本结构



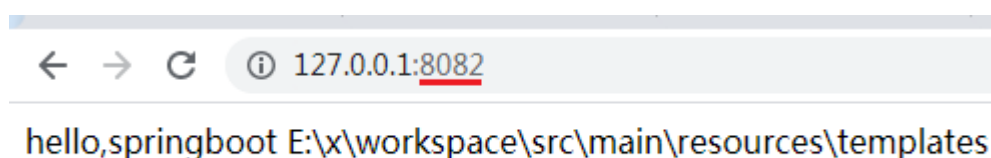
8080端口被占用了，修改默认端口



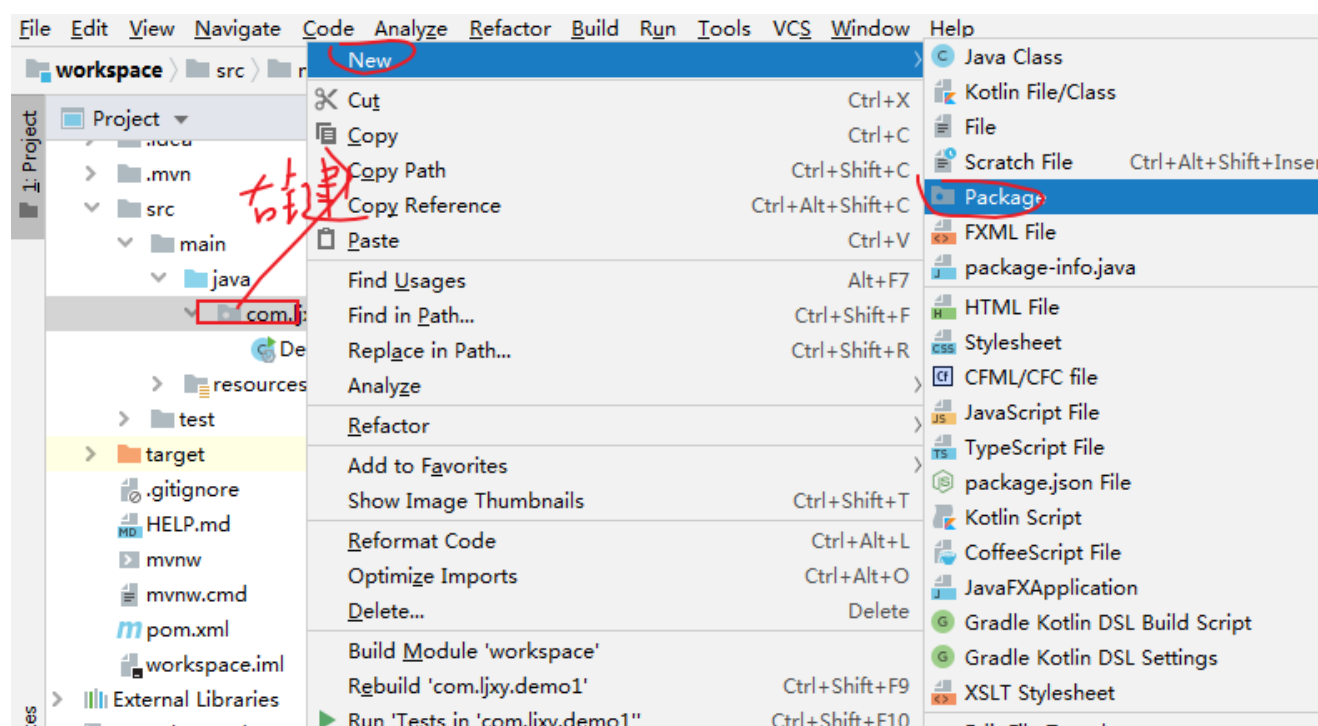
创建index.html文件

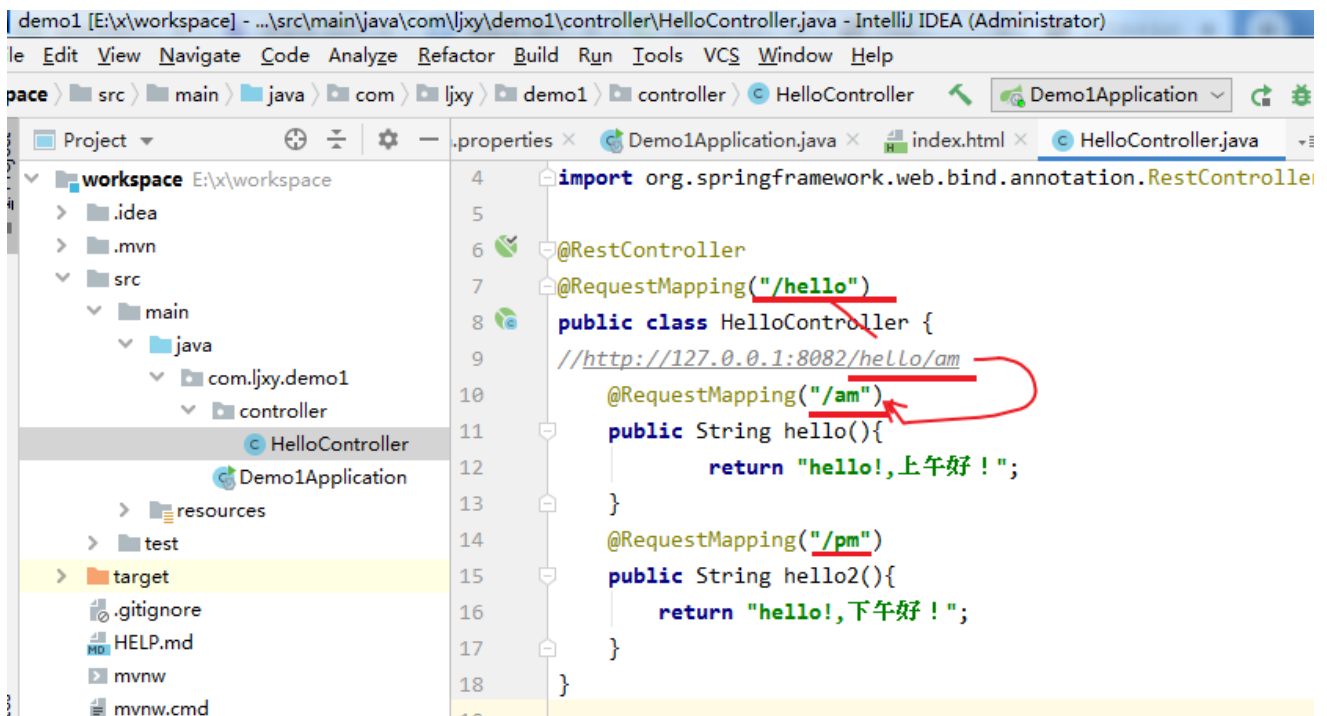
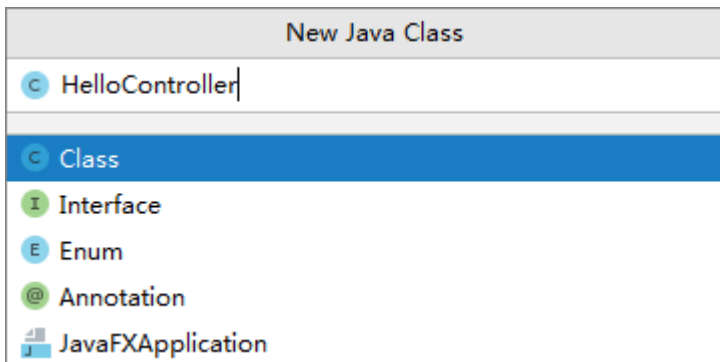
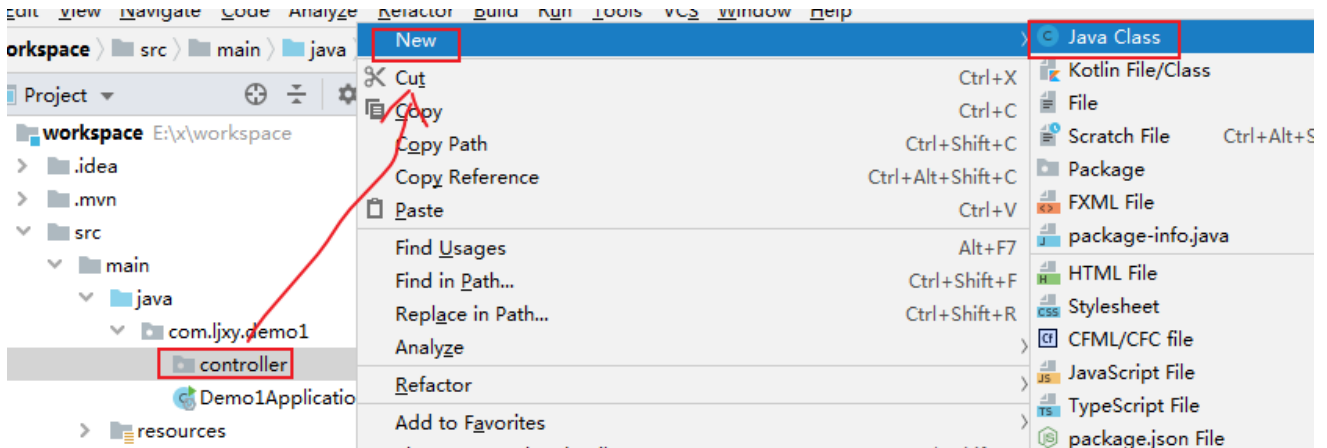
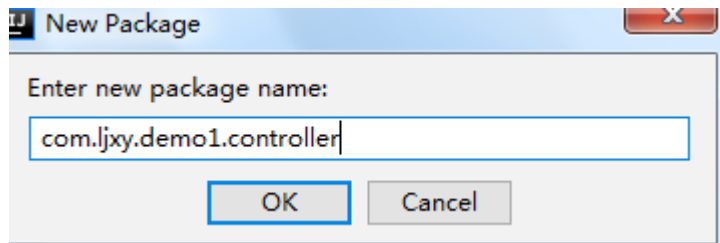


重启项目

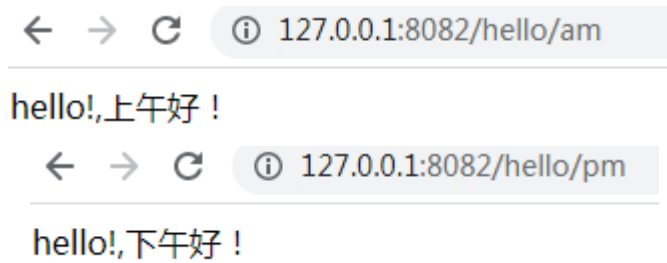


创建一个用于Web访问的Controller





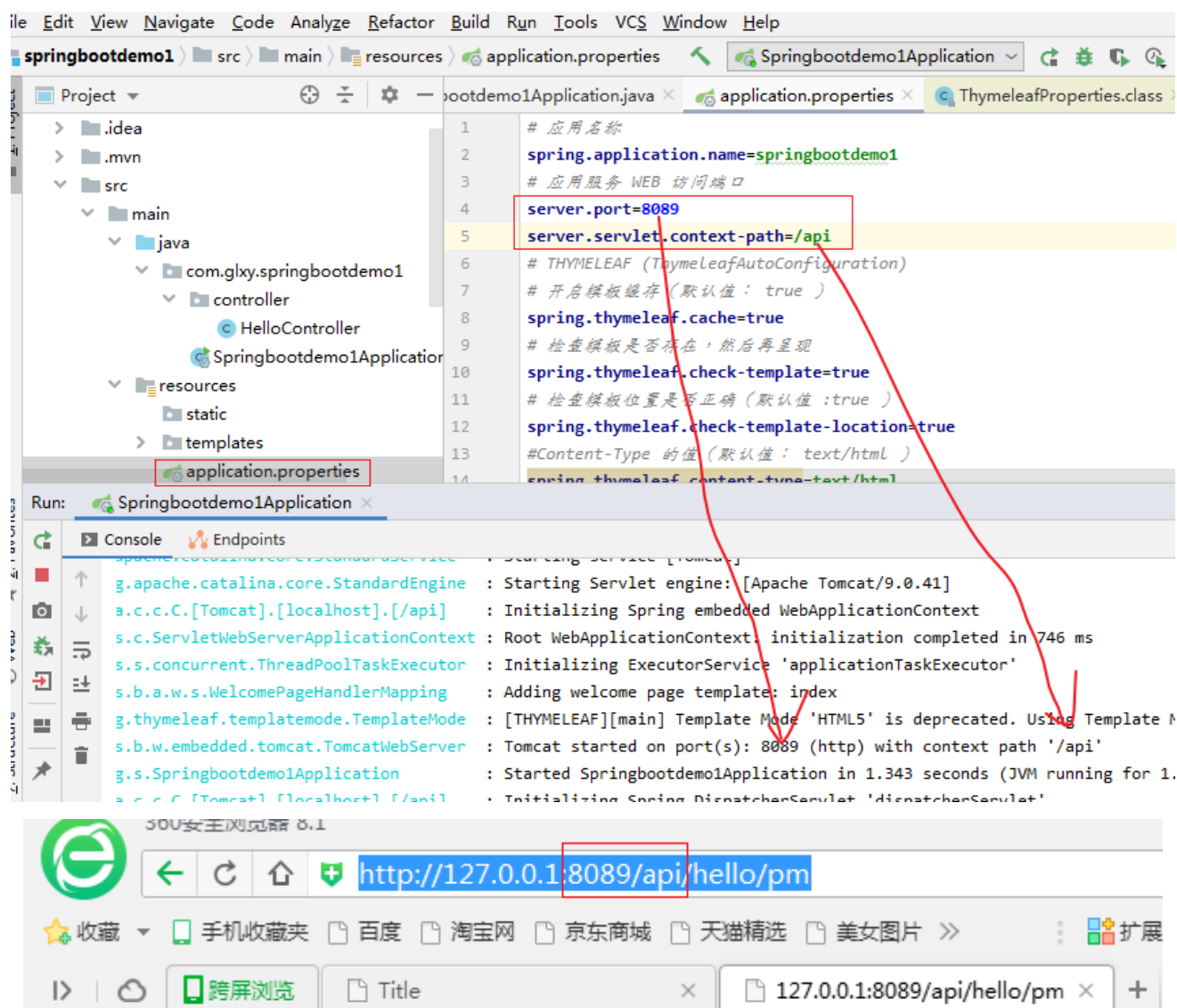
运行项目



热部署

全局配置文件

```
#修改端口 http://127.0.0.1:8089/api/hello/pm
# 应用名称
spring.application.name=springbootdemo1
# 应用服务 WEB 访问端口
server.port=8089
server.servlet.context-path=/api
# THYMELEAF (ThymeleafAutoConfiguration)
# 开启模板缓存 (默认值: true)
spring.thymeleaf.cache=true
# 检查模板是否存在, 然后再呈现
spring.thymeleaf.check-template=true
# 检查模板位置是否正确 (默认值: true)
spring.thymeleaf.check-template-location=true
#Content-Type 的值 (默认值: text/html)
spring.thymeleaf.content-type=text/html
# 开启 MVC Thymeleaf 视图解析 (默认值: true)
spring.thymeleaf.enabled=true
# 模板编码
spring.thymeleaf.encoding=UTF-8
# 要被排除在解析之外的视图名称列表, 逗号分隔
spring.thymeleaf.excluded-view-names=
# 要运行于模板之上的模板模式。另见 StandardTemplate-ModeHandlers (默认值: HTML5)
spring.thymeleaf.mode=HTML5
# 在构建 URL 时添加到视图名称前的前缀 (默认值: classpath:/templates/)
spring.thymeleaf.prefix=classpath:/templates/
# 在构建 URL 时添加到视图名称后的后缀 (默认值: .html)
spring.thymeleaf.suffix=.html
```

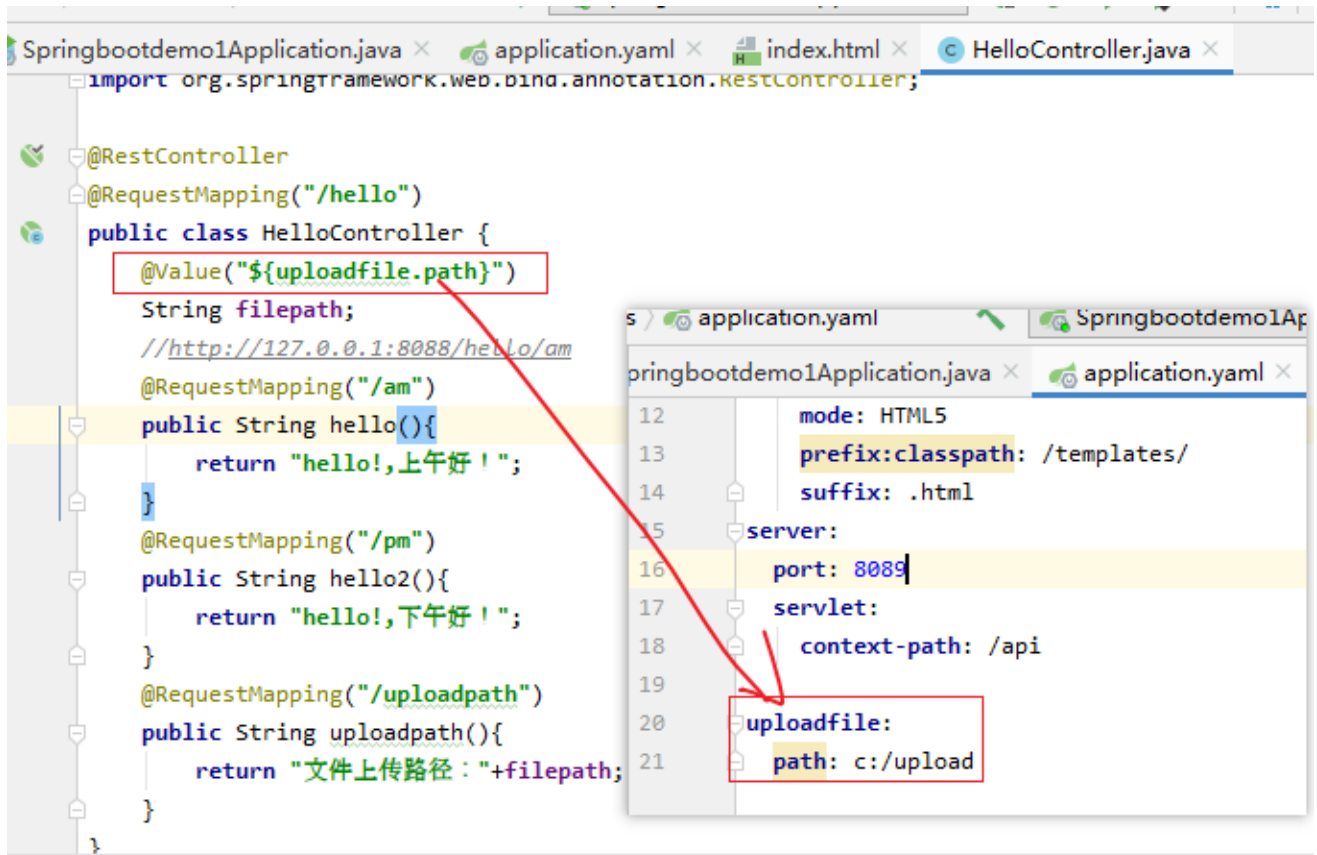
hello!, 下午好!

改为yaml格式

```
spring:
  application:
    name: springbootdemo1
  thymeleaf:
    cache: true
    check-template: true
    check-template-location: true
    content-type: text/html
    enabled: true
    encoding: UTF-8
    excluded-view-names: ''
    mode: HTML5
    prefix:classpath: /templates/
    suffix: .html
  server:
    port: 8089
    servlet:
```

```
context-path: /api
```

读取自定义属性



自定义属性提示功能

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-configuration-processor</artifactId>
  <optional>true</optional>
</dependency>
```

使用自定义属性文件

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

@RestController
@RequestMapping("/hello")
@PropertySource("classpath:my.properties")
public class HelloController {
    @Value("${uploadfile.path}")
    String filepath;
    //http://127.0.0.1:8088/hello/am
}
```

application.yml × my.properties × pom.xml × index.html × HelloController.java ×

1 uploadfile.path=c:/upload2

Profile多环境配置

💡 多环境配置文件格式



application-{profile}.properties

注：{profile}对应具体的环境标识



激活指定环境的方式



通过命令行方式激活指定环境的配置文件



在全局配置文件设置spring.profiles.active属性激活

The screenshot shows an IDE with the following components:

- Project Explorer:** Shows the project structure with `resources/application-dev.yml` selected. A red arrow points from this file to the `active: dev` line in the `application.yml` snippet.
- Main Editor:** Displays `application-dev.yml` with the following content:

```
enabled: true
encoding: UTF-8
excluded-view-names: ''
mode: HTML5
prefix: classpath: /templates/
suffix: .html

server:
  port: 8089
  servlet:
    context-path: /dev/api
```

A red arrow points from the `context-path: /dev/api` line to the console output.
- Snippet:** A small window shows the `application.yml` structure:

```
1 spring:
2   profiles:
3     active: dev
```
- Console:** Displays the application startup logs:

```
s.concurrent.ThreadPoolTaskExecutor : Initializing ExecutorService 'applicationTaskExecutor'
b.a.w.s.WelcomePageHandlerMapping    : Adding welcome page template: index
thymeleaf.templateengine.TemplateMode : [THYMELEAF][main] Template Mode 'HTML5' is deprecated. Using Template Mode 'HTML'
b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port(s): 8089 (http) with context path '/dev/api'
s.Springbootdemo1Application         : Started Springbootdemo1Application in 1.267 seconds (JVM running for 1.752)
```