

建立开发环境，一个简单Spring Boot应用程序的开发与运行

陶召胜

准备好开发环境

- 请在第一次上课前安装好开发环境，参见文件：建立基于IntelliJ IDEA的开发环境.txt

一、Java开发环境

二、IntelliJ IDEA安装

三、Git本地安装 (Windows、 macOS)

We're committed to giving back to our wonderful community, which is why IntelliJ IDEA Community Edition is completely free to use



IntelliJ IDEA Community Edition

The IDE for pure Java and Kotlin development

Download

.exe ▼

Free, built on open source

Git本地安装 (Windows、macOS)

1) Git官网

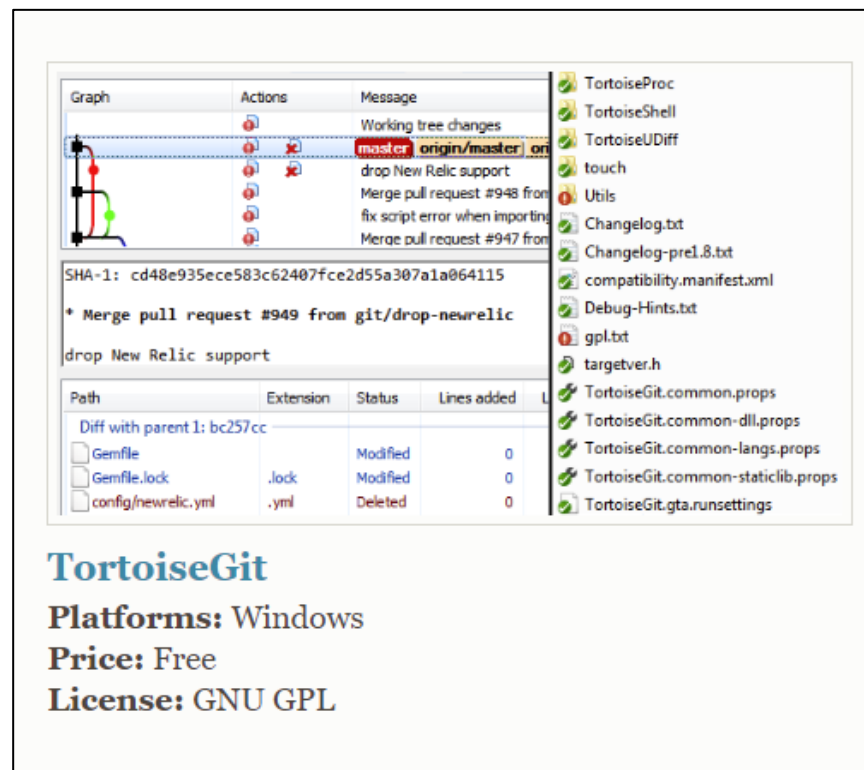
<https://git-scm.com/>

2) Git下载和安装

<https://git-scm.com/downloads>

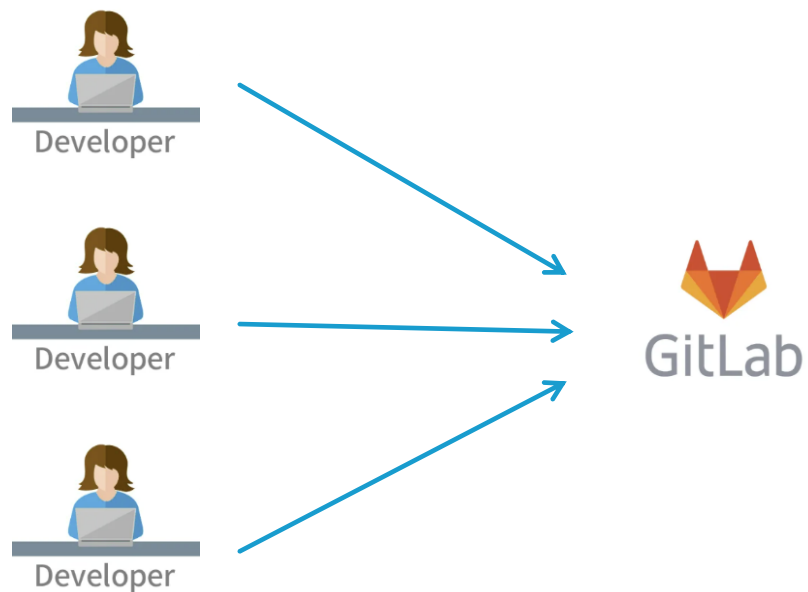
3) GUI客户端 (可选)

<https://git-scm.com/downloads/guis>



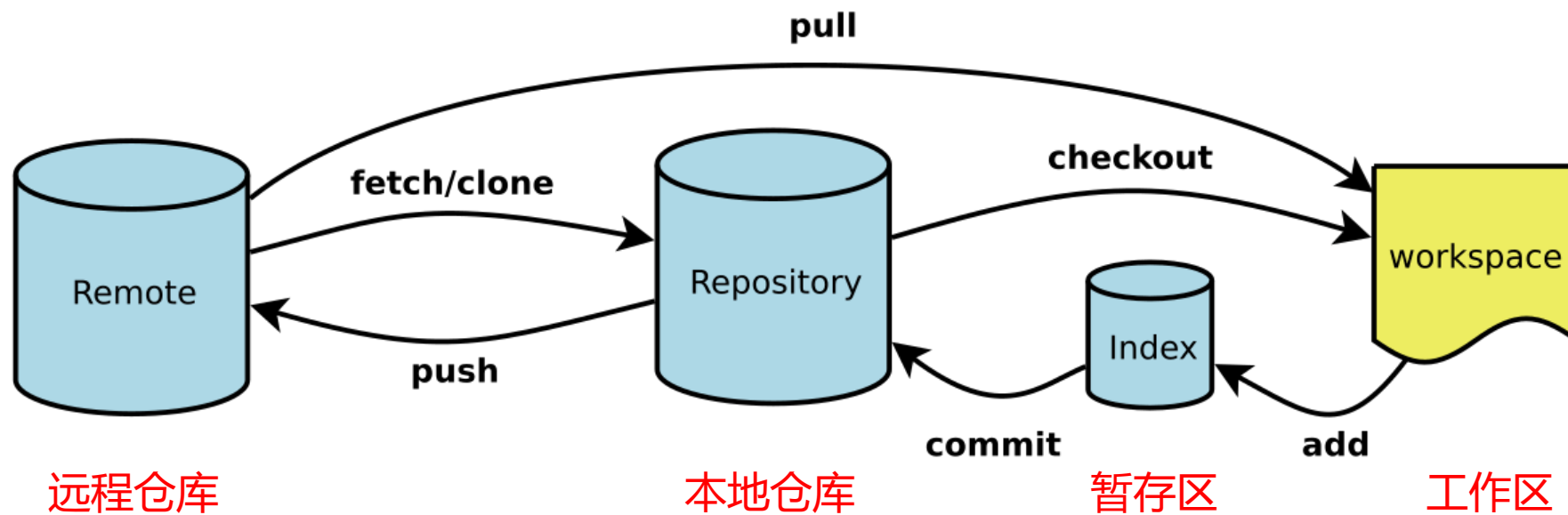
源代码仓库管理

- 也称为版本控制（version control）系统，常用工具有：GitLab、SVN（Subversion）、Bitbucket等
- 需纳入版本控制的有：功能代码、测试代码、测试脚本、构建脚本、部署脚本、配置文件等



Git关键概念

Git常用命令使用: https://blog.csdn.net/web_csdn_share/article/details/79243308



前置条件

- Java语法、JDK基础
 - Maven，构建工具
 - Git常用命令
 - 自动化测试框架：JUnit
- Maven常用的阶段：
 - ✓ compile 编译项目的源代码。
 - ✓ test-compile 编译测试源代码。
 - ✓ test 使用合适的单元测试框架运行测试。这些测试代码不会被打包或部署。
 - ✓ package 接受编译好的代码，打包成可发布的格式，如 JAR 。
 - ✓ install 将包安装至本地仓库，以让其它项目依赖。
 - ✓ deploy 将最终的包复制到远程的仓库，以让其它开发人员与项目共享

什么是开发框架？

- 开发框架(Framework)是整个或部分系统的可重用设计，表现为一组抽象构件及构件实例间交互的方法；另一种定义认为，框架是可被应用开发者定制的应用骨架
- 前端常用开发框架：Vue
- Java常用开发框架：Spring
- C++开发框架：Qt
- Python开发框架：Django

可被应用开发者定制的应用骨架：RuoYi-Cloud

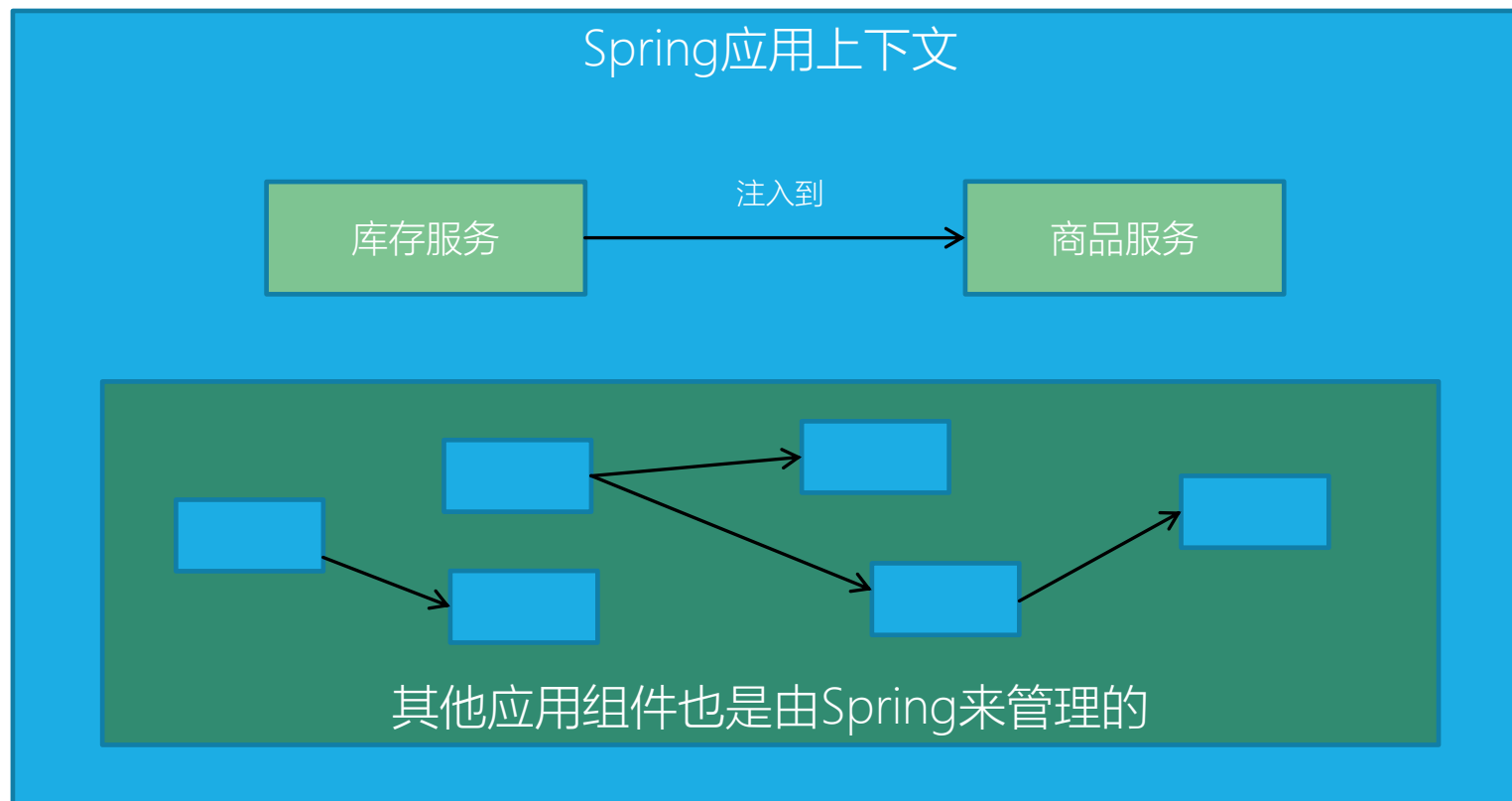
- 源代码： https://gitee.com/y_project/RuoYi-Cloud
- 文档： <http://doc.ruoyi.vip/ruoyi-cloud/>
- 架构图： <https://www.processon.com/view/5ec290195653bb6f2a18504e>

Spring是Java生态圈的主流开发框架

- 轻量级 (Lightweight)
- 非侵入性 (No intrusive)
- 容器 (Container)
- Dependency Injection
- AOP (Aspect-oriented programming)
- 持久层 (JDBC封装、事务管理、ORM工具整合)
- Spring-MVC Web框架
- 其他企业服务的封装



Spring的核心是提供了一个容器（container）



图摘自《Spring 实战》（第6版）

Spring的衍生

- Spring：是核心、基础框架
- Spring Boot：简化基于Spring的开发，自动配置
- Spring Cloud：基于云的、分布式系统开发，相关技术：容器、微服务、DevOps

• 2022年云基础设施应用范围在中国企业中进一步拓展

30%

中国企业整体上云率

云上基础设施再夯实，降低企业用云门槛

80%

中国企业已经使用容器、微服务、DevOps等云原生技术

Severless升级为行业带来研发模式变革的可能，组装式研发提效50%

95%

央国企正在使用或者考虑使用云原生技术

云上应用深入，新应用备受关注

数据来源：公开数据和专家访谈，《第四期（2021-2022）传统行业云原生技术落地调研——央国企篇》

参考书目

- 书目1: 《Spring实战》(第4版), 作者Craig Walls, 人民邮电出版社, 2016年, ISBN 978-7-115-41730-5
- 书目2: 《Spring实战》(第6版), 作者Craig Walls, 人民邮电出版社, 2022年, ISBN 978-7-115-59869-1
- 书目3: 《Spring微服务实战》, 作者John Carnell, 人民邮电出版社, 2018年, ISBN 978-7-115-48118-4



教学内容安排

周次	教学内容	授课方式	备注
1	建立开发环境，一个简单Spring Boot应用程序的开发与运行	课堂教学	
2	依赖注入	课堂教学	
3	面向切面编程	课堂教学	
4	Web开发框架	课堂教学	
5	Spring Data JDBC/JPA	课堂教学	
6	NoSQL数据库使用和编程	课堂教学	
7	Spring Security	课堂教学	
8	属性配置	课堂教学	..
9	Rest服务开发	课堂教学	
10	OAuth2	课堂教学	
11	发送异步消息	课堂教学	
12	Spring集成	课堂教学	
13	反应式编程基础	课堂教学	
14	开发反应式API	课堂教学	
15	Docker使用	课堂教学	
16	微服务开发与部署	课堂教学	
17	k8s使用与部署	课堂教学	

Spring Boot开发框架

- Spring Boot是由Pivotal团队提供的全新框架，其设计目的是用来简化Spring应用的初始搭建以及开发过程
- Spring Boot的特征有：
 - (1) 可以创建独立的Spring应用程序，并且基于其Maven或Gradle插件，可以创建可执行的JARs和WARs
 - (2) 内嵌Tomcat或Jetty等Servlet容器
 - (3) 提供自动配置的“starter”项目对象模型（POMS）以简化Maven配置
 - (4) 尽可能自动配置Spring容器
 - (5) 提供准备好的特性，如指标、健康检查和外部化配置
 - (6) 绝对没有代码生成，不需要XML配置



一个简单Spring Boot应用程序的开发与运行

- 代码下载路径: <https://github.com/tzs919/taco-cloud-01.git>

Spring initializr: 生成初始框架代码

- 访问: <https://start.spring.io/>

The screenshot shows the Spring Initializr web application interface. The header includes the Spring logo and the text "spring initializr". The main content area is divided into several sections:

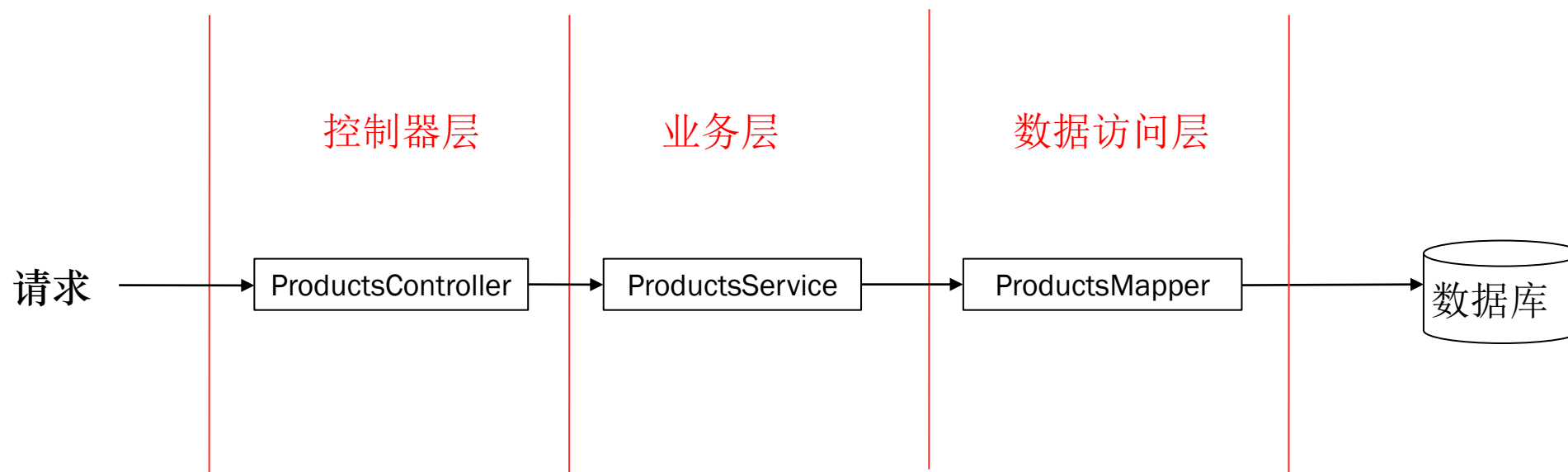
- Project:** Radio buttons for "Gradle - Groovy", "Gradle - Kotlin", and "Maven" (selected).
- Language:** Radio buttons for "Java" (selected), "Kotlin", and "Groovy".
- Spring Boot:** Radio buttons for "3.2.0 (SNAPSHOT)", "3.2.0 (M2)", "3.1.4 (SNAPSHOT)", "3.1.3", "3.0.11 (SNAPSHOT)", "3.0.10", "2.7.16 (SNAPSHOT)", and "2.7.15" (selected).
- Project Metadata:** Text input fields for "Group" (tacos), "Artifact" (taco-cloud), "Name" (taco-cloud), "Description" (Taco Cloud Example), and "Package name" (tacos). A "Packaging" section with radio buttons for "Jar" (selected) and "War".
- Dependencies:** A section with a button "ADD DEPENDENCIES... CTRL + B". It lists "Spring Boot DevTools" (DEVELOPER TOOLS) and "Spring Web" (WEB).
- Thymeleaf:** A section with a button "TEMPLATE ENGINES". It describes Thymeleaf as a modern server-side Java template engine.

At the bottom, there are three buttons: "GENERATE CTRL + G", "EXPLORE CTRL + SPACE", and "SHARE...".

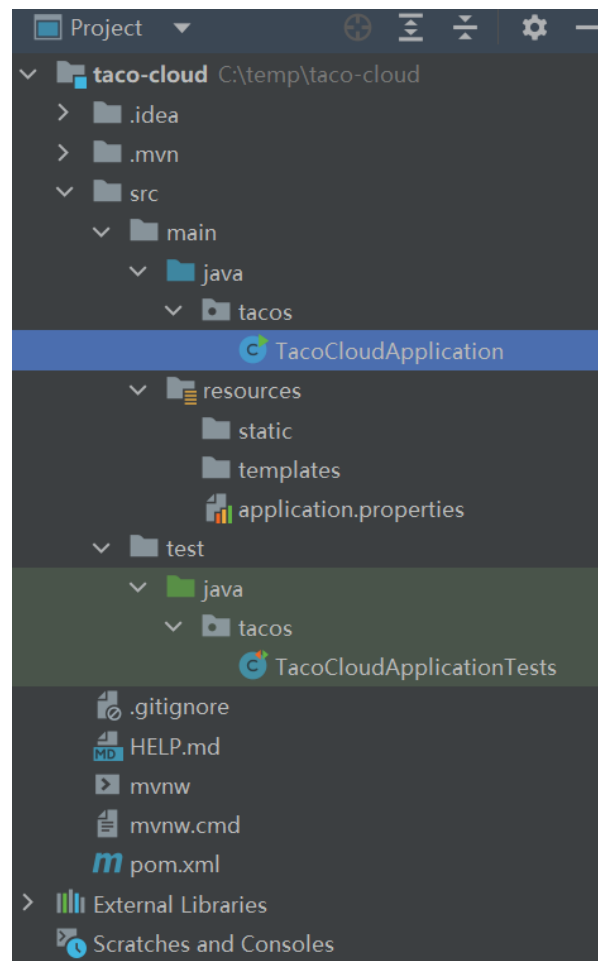
依赖

- spring-boot-devtools
- spring-boot-starter-web
- spring-boot-starter-thymeleaf

Spring Web开发框架的分层



项目文件夹结构



自动化测试框架：JUnit

- JUnit是一个Java语言的自动化测试框架。它由Kent Beck和Erich Gamma建立，xUnit家族中的一员
- JUnit有它自己的JUnit扩展生态圈。多数Java的开发环境都已经集成了JUnit作为自动化测试的工具

所有的测试

Class	花的时间	失败 (区别)	跳过 (区别)	Pass (区别)	总数 (区别)
HomeControllerTest	3 秒	0	0	1	1
SpitterControllerTest	1.4 秒	0	0	3	3
SpittleControllerTest	0.4 秒	0	0	4	4

测试用例与测试套

- 测试用例 (Test Case) 是为某个特殊目标而编制的一组测试输入、执行条件以及预期结果, 以便测试某个程序路径或核实是否满足某个特定需求
- 通常把一组相关的测试称为一个测试套 (Test Suite), 通过测试套, 将服务于同一个测试目的或同一运行环境下的一系列测试用例有机的组合起来

```
✓ <default package>
  ✓ HomeControllerTest
    ✓ testHomePage
  ✓ SpitterControllerTest
    ✓ shouldShowRegistration
    ✓ shouldFailValidationWithNoData
    ✓ shouldProcessRegistration
  ✓ SpittleControllerTest
    ✓ saveSpittle
    ✓ shouldShowPagedSpittles
    ✓ houldShowRecentSpittles
    ✓ testSpittle
```

使用JUnit写测试用例

- 测试方法上必须使用@Test
- 测试方法必须使用（public可选）void进行修饰
- 新建一个源代码目录来存放测试代码
- 测试类的包应该和被测试类的包一样
- 测试单元中的每个方法要能够独立测试，其方法不能有依赖

```
@Test
public void testAdd() {
    int result = calculate.add(7, 8);
    assertEquals(15, result);
}
```

```
@Test
public void multiply() {
    int result = calculate.multiply(7, 8);
    assertThat(result, is(56));
}
```

断言

- 将程序预期的结果与程序运行的最终结果进行比对，确保对结果的可预知性

```
@Test
public void testAdd() {
    int result = calculate.add(7, 8);
    assertEquals(15, result);
}
```

```
@Test
public void multiply() {
    int result = calculate.multiply(7, 8);
    assertThat(result, is(56));
}
```

添加

- 处理Web请求，控制器：HomeController.java
- 定义视图，视图模板：home.html
- static 图片
- 测试

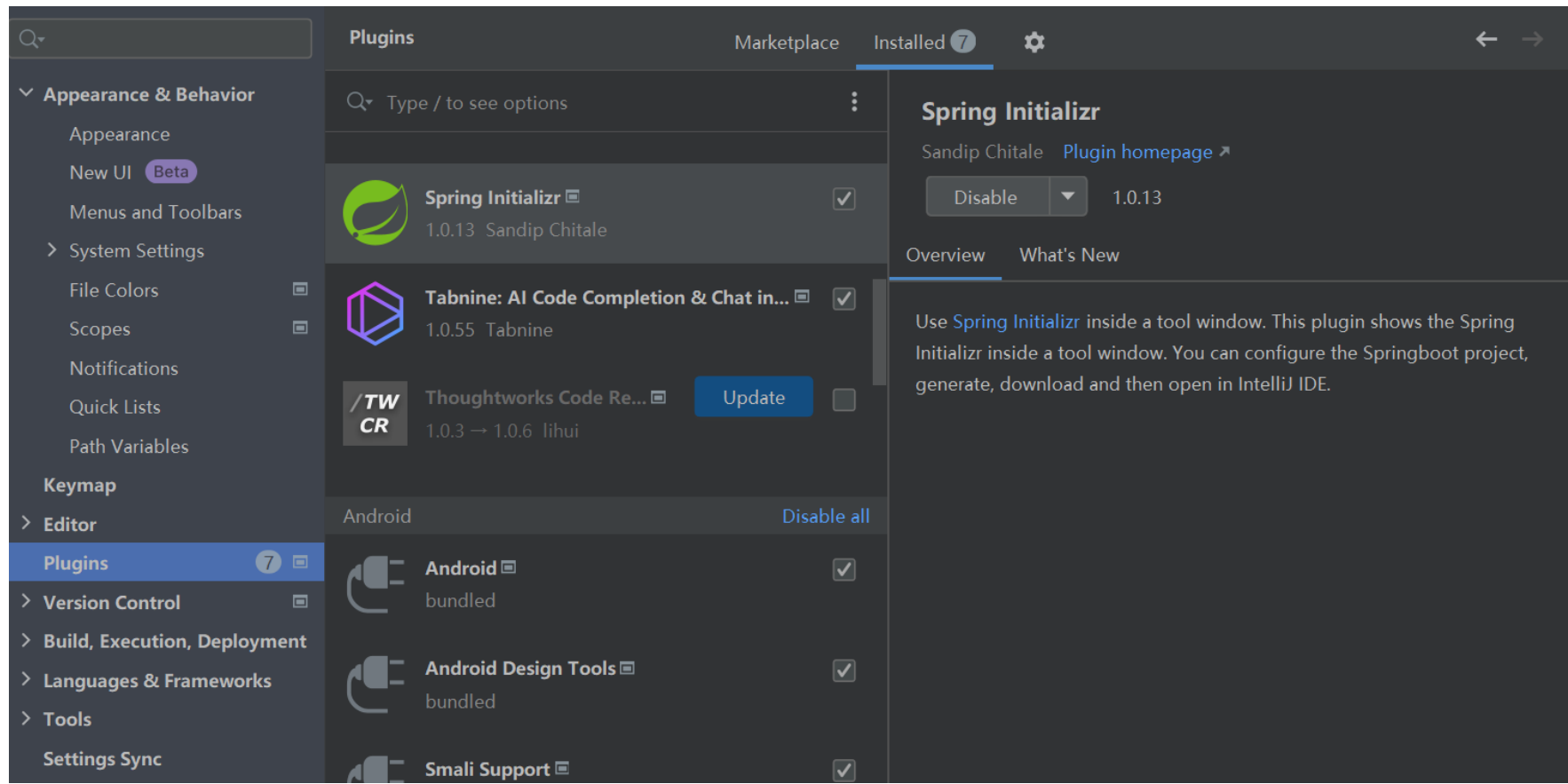
构建和运行应用

- 插件: `spring-boot-maven-plugin`
- `mvn spring-boot:run`

访问

- <http://localhost:8080/>

Spring Initializr插件



开发期工具：Spring Boot DevTools

- 代码变更后应用会自动重启（需要借助IDE的自动编译）
- 当面向浏览器的资源（如模板、JavaScript、样式表）等发生变化时，会自动刷新浏览器
 - ✓ 应用会暴露LiveReload端口，日志如：LiveReload server is running on port 35729
 - ✓ IntelliJ IDEA要做的配置见下页ppt
 - ✓ 需要安装浏览器插件：LiveReload，并打开
- 自动禁用模板缓存
- 如果使用H2数据库，则内置了H2控制台
 - ✓ 访问：<http://localhost:8080/h2-console>

使用Live Reload在IntelliJ IDEA和Chrome实时编译部署和调试

- pom.xml加依赖: spring-boot-devtools
- IntelliJ Idea项目
 - ✓ File->Settings->Build, Execution, Deployment -> Compiler, 在自动构建项目的复选框上 Build project automatically做勾选。
 - ✓ File->Settings->Advanced Settings 中勾选: Allow auto-make to start even if developed application is currently running

用spring boot编译出支持不同版本的jdk

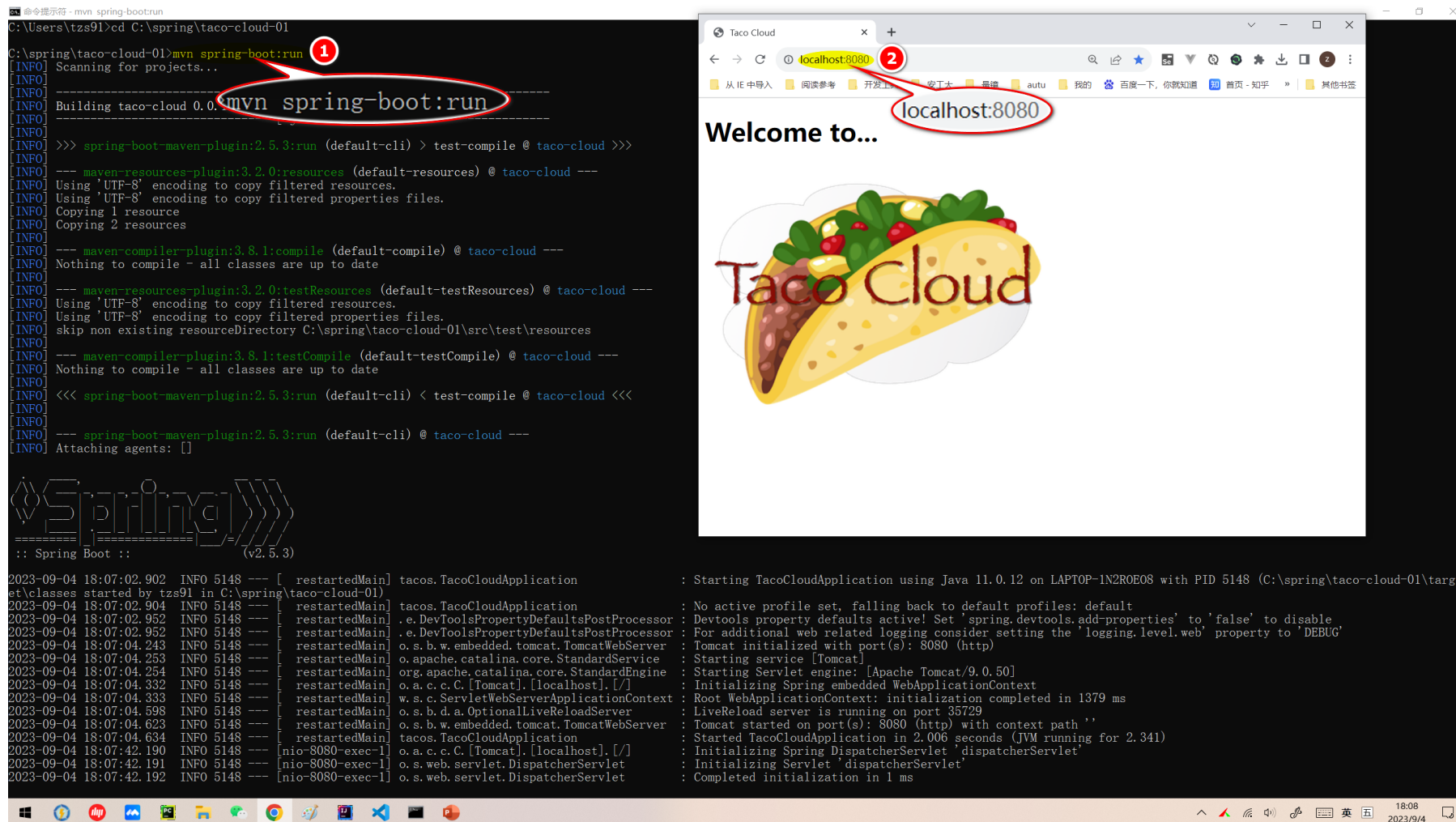
在pom.xml文件添加以下属性

```
<properties>  
    <java.version>11</java.version>  
</properties>
```

如果当前安装的jdk是17，则编译出的版本支持：1.8、11、17

作业

- 把老师今天课堂上演示的例子重做一遍，截图提交



谢谢观看！

