



Java程序高级开发

汪海龙



教师简介

汪海龙

福州大学 计算机科学与技术

中南大学 计算机技术

中国煤炭科工集团常州研究院有限公司

经历编程的学、用、教

有能力、有信心能够教好大家

联系方式



1班QQ群: 487737701

2班QQ群: 930895546

办公地点: 化工楼301

课堂规矩



理论课不说话

实验课带电脑



课程概要

- 一、课程介绍
- 二、如何学好该课程
- 三、概念解析
- 四、软件安装
- 五、SpringBoot入门



一、课程介绍

课程性质



基于Java程序设计开发

Java Web 后端开发

Java企业级应用开发

课程安排



星期 节次	一	二	三	四	五
1				Java程序高级开发(1)班 [轻工515] (01-07周)	
2		.NET框架应用开发(1)班 [化工314]		Java程序高级开发综合实训(1)班 [轻工515] (08-17周)	
3				Java程序高级开发(2)班 [轻工515] (01-07周)	
4				Java程序高级开发综合实训(2)班 [轻工515] (08-17周)	
5				.NET框架应用开发(1)班 [化工314] (01-05周)	
6		Java程序高级开发(1)班 [化工316] Java程序高级开发综合实训(1)班 [轻工518]		.NET框架应用开发(1)班 [轻工515] [5~8节] (06-10周) .NET框架应用开发实践(1)班 [轻工515] [5~8节]	
7		Java程序高级开发(2)班 [化工316]			
8		Java程序高级开发综合实训(2)班 [轻工518]			

1-18周
70课时



课程考核

✓ Java程序高级开发

- 平时成绩：平时作业30分
- 期末闭卷考试：70分

✓ Java程序高级开发实践

- 以小组为单位，完成课程实践



课程内容

- Spring Boot
- Spring
- Spring MVC
- Mybatis
- Spring Boot与其他技术整合
- 项目



课程特点

- Java后端最主流的技术
- 实践性强
- 课程相对简单
- 即学即用



二、如何学好该课程



编程是什么

- 技能，技术，技艺
- 开车，厨师
- 面向实践，面向生产，创造价值



为什么要学好

➤ 简历上的项目，面试，毕业设计，就业等

➤ 对比网络课程

➤ 对比线下课程

基于SpringBoot的仓储物流系统的设计与实现
基于若依框架的景点管理系统的设计与实现
基于SpringBoot的新闻网站系统的设计与实现
基于SpringBoot的房产中介系统的设计与实现
基于SpringBoot的家装管理系统的设计与实现
基于SrpingBoot的就业管理系统的设计与实现
基于SpringBoot的在线判题系统的设计与实现
基于SpringBoot的运动管理系统的设计与实现



怎么学好

- 师傅和学徒，教练和学员
- 老师会认真教，做好课程服务
- 学生要**多动手**，多学，多查，多问



三、概念解析

客户端/服务端



客户端（Client）是指与服务端（Server）相对应的，用于在本地为用户提供服务的程序。这些程序通常安装在客户机上，并与服务器端程序互相配合运行以实现各种功能。常用的客户端包括网页浏览器以及电子邮件客户端。

服务端（Server）是一种有针对性的服务程序。服务端是为客户端服务的，服务的内容诸如向客户端提供资源，保存客户端数据等。运行服务端的电脑称为“服务器”。

提问：两者是如何联系？需要在两台不同的电脑上吗？



C/S和B/S架构

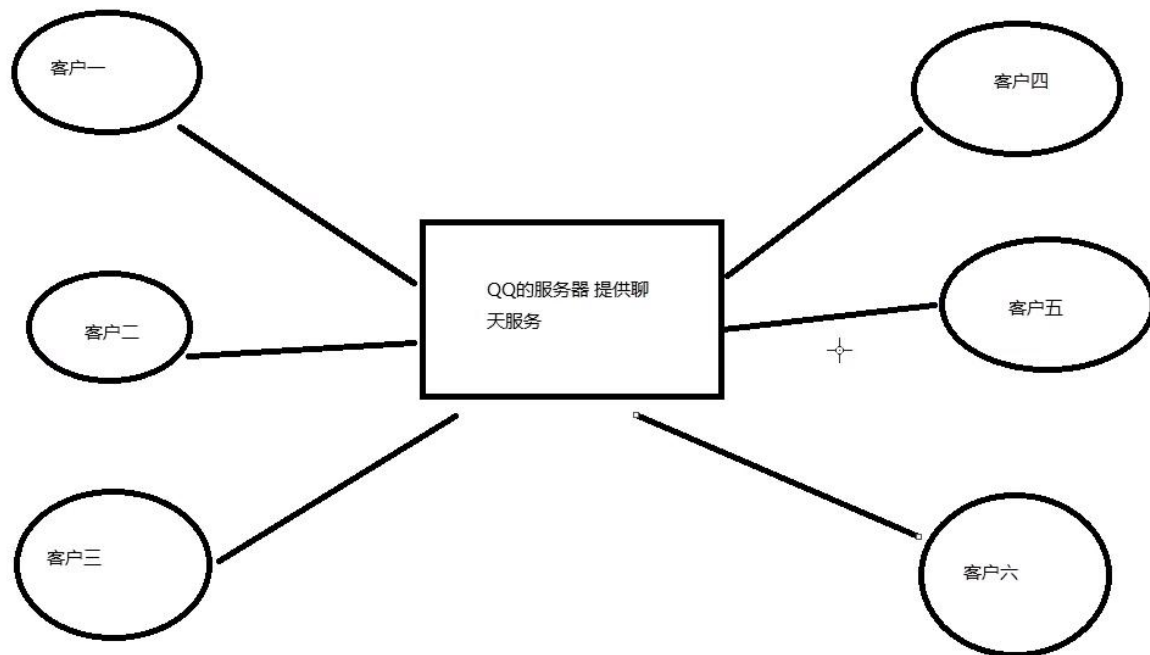
Client-Server，客户机和服务器架构。这种结构最显著的特征是需要安装客户端，通过客户端程序来访问应用系统。常见的C/S架构有应用商店下载的各类软件。

Browser-Server，浏览器和服务器架构，包含客户端浏览器、web应用服务器、数据库服务器的软件系统。用户只需要一个浏览器就可以访问服务。常见的B/S架构有各类网站。



客户端/服务端举例

1.QQ是如何运行的？



客户端/服务端举例



2.学校官网





C/S和B/S架构特点

C/S架构特点

- 1.响应速度快：能充分发挥客户端PC的处理能力，很多工作可以在客户端处理后再提交给服务器。
- 2.维护成本较高：系统更新时，既要更新服务端，也要更新客户端。



C/S和B/S架构特点

B/S架构特点

- 1.分布性强，客户端零维护：只要有网络、浏览器，可以随时随地进行查询、浏览等业务处理。
- 2.维护成本较低：系统更新时候，只需要更新服务端，所有用户就能看到最新功能。

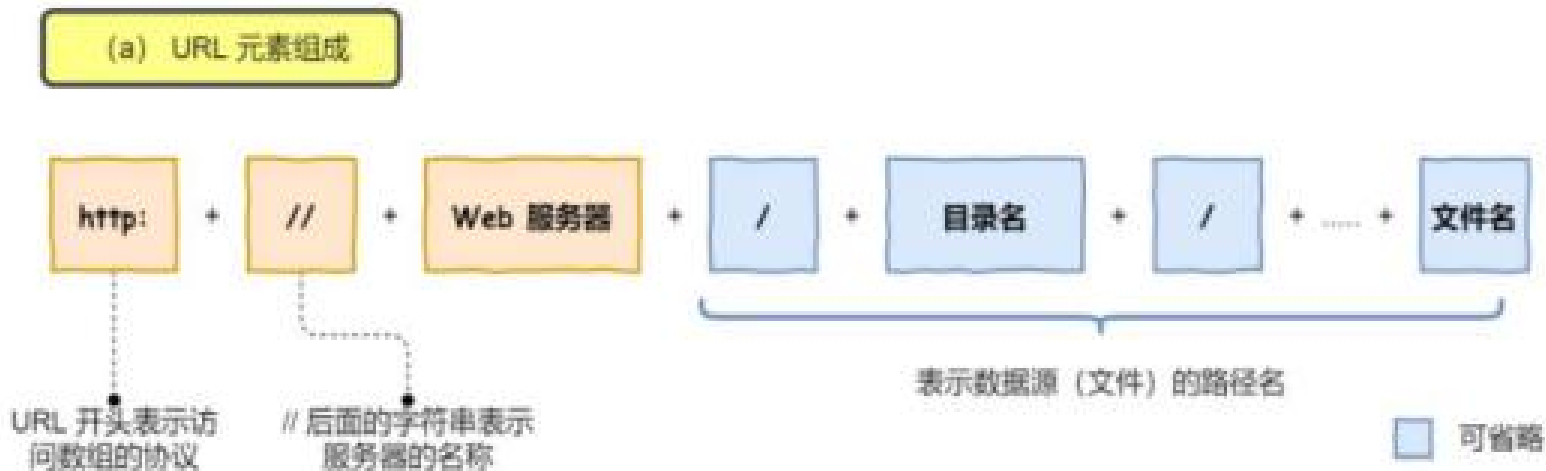


提问

- 1.我们安装的MySQL是客户端还是服务端？
- 2.这门课是做客户端/服务端开发？ C/S、 B/S架构？

客户端服务端交互

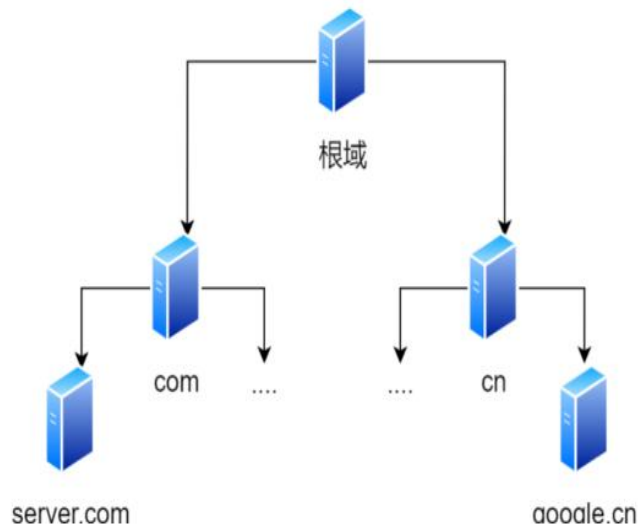
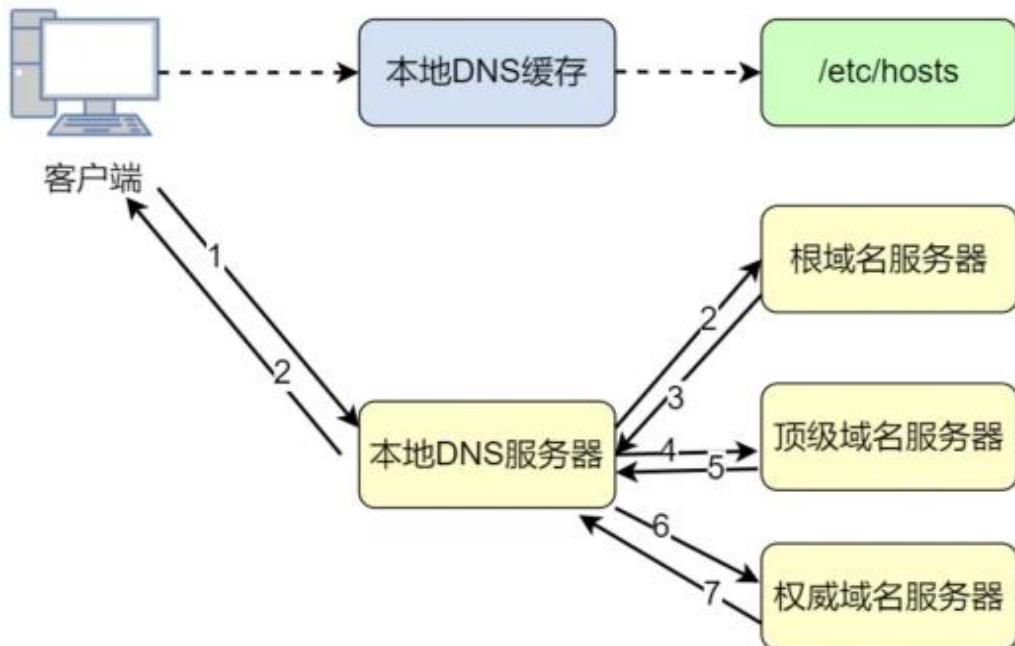
1. 浏览器对输入的url进行拆分解析



- 路径是相对路径，是以web服务器根目录为起始目录的
- URL对应Web服务器里的一个文件资源

客户端服务端交互

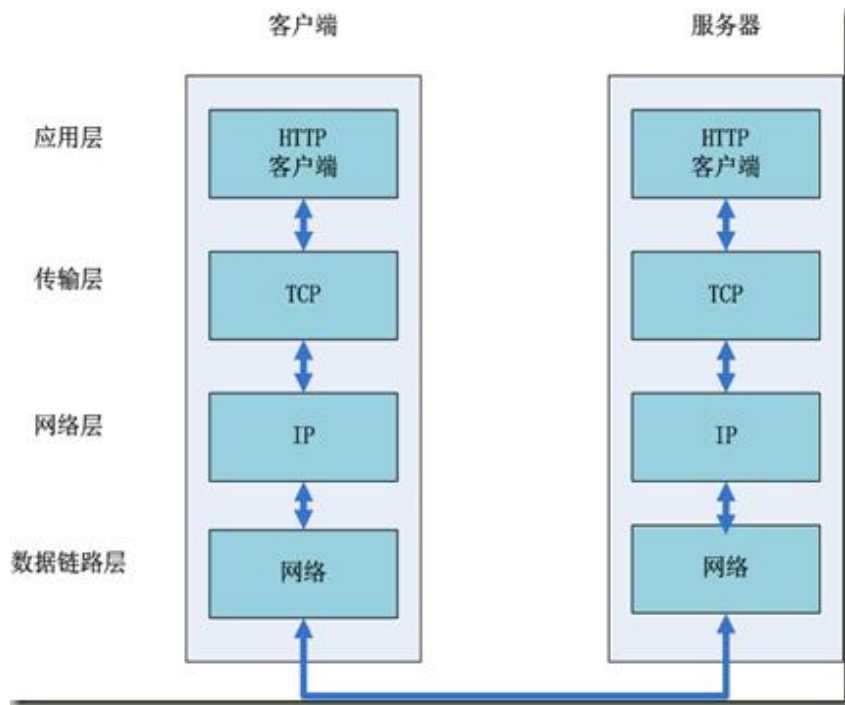
2. 查询域名对应的 IP 地址：DNS服务器





客户端服务端交互

3. 客户端构造 HTTP 请求报文，向服务端发送HTTP请求





客户端服务端交互

4. 服务端收到客户端发送的HTTP请求后，查找客户端请求的资源，并返回响应报文

5. 浏览器收到响应报文，渲染页面

详细可见博客

<https://blog.csdn.net/V13807970340/article/details/129753277>



客户端服务端交互举例

1.注册

2.登录



四、软件安装



软件安装

- IDEA

下载地址: <https://www.jetbrains.com/idea/download/other.html>

激活教程: <https://www.docs.qq.com/doc/DRGxwbEJrU3hrcVNr>

建议下载IDEA 2021.2.3版本, 并安装在D:\JavaEE路径下

- Maven

安装教程: https://blog.csdn.net/i_for/article/details/131138898

建议下载Maven 3.6.3版本, 并安装在D:\JavaEE路径下



软件安装

- Chrome
- Postman
- `http://localhost:8899/users/findUserDept`

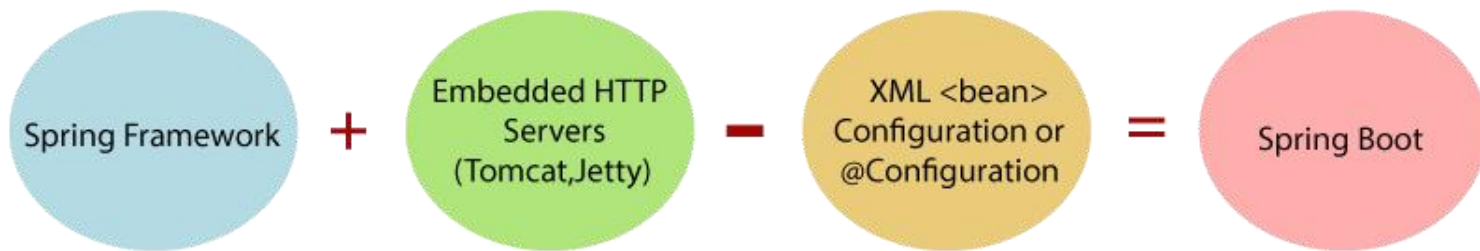


五、Spring Boot入门



Spring Boot

Spring Boot是基于Spring框架之上的，提供了快速应用程序开发功能，可以用简便、快捷的方式来配置和运行基于Web的应用程序。





Spring Boot优点

1. 独立运行

Spring Boot 项目可以以 jar 包的形式独立运行，只需通过命令
“ java-jar xx.jar” 即可运行。

2. 内嵌 Servlet 容器

Spring Boot 使用嵌入式的 Servlet 容器（例如 Tomcat、Jetty 等），
可以轻松测试Web应用程序，无需部署WAR文件。



Spring Boot优点

3. 提供 starter 简化 Maven 配置

Spring Boot 提供了一系列的 “starter” 项目对象模型 (POMS) 来简化 Maven 配置。

4. 提供了大量的自动配置

Spring Boot 提供了大量的默认自动配置，来简化项目的开发，开发人员也通过配置文件修改默认配置。



IDEA创建Spring Boot项目

Spring Boot 版本及其环境配置要求如下

- Spring Boot 2.7.15
- JDK 8.0
- Maven 3.6.3
- IntelliJ IDEA 2021.2.3

IDEA创建Spring Boot项目



New Project

Server URL: start.aliyun.com

Name: SpringBoot01 项目名称

Location: D:\JavaProject\SpringBoot01 项目存放路径

Language: Java Kotlin Groovy

Type: Maven Gradle

Group: edu.fdzcx 一般是公司域名的倒写

Artifact: SpringBoot01

Package name: edu.fdzcx.springboot01

Project SDK: 1.8 version 1.8.0_221

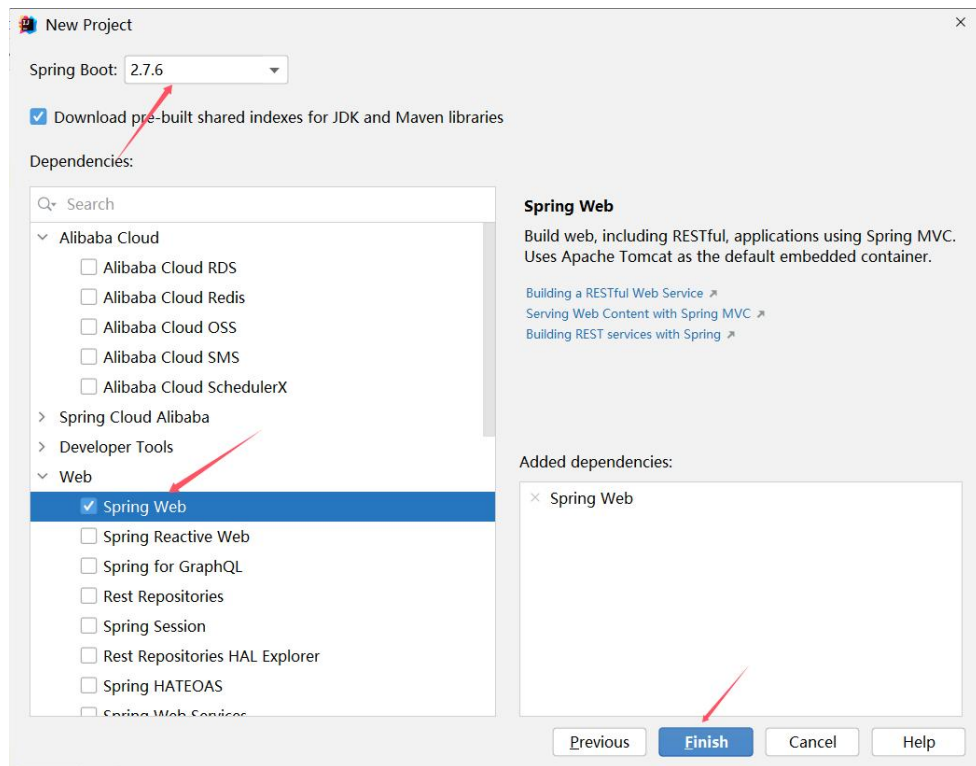
Java: 8

Packaging: Jar War

Previous Next Cancel Help

如果选不了Java 8,
将Server URL改为
<https://start.aliyun.com/>

IDEA创建Spring Boot项目

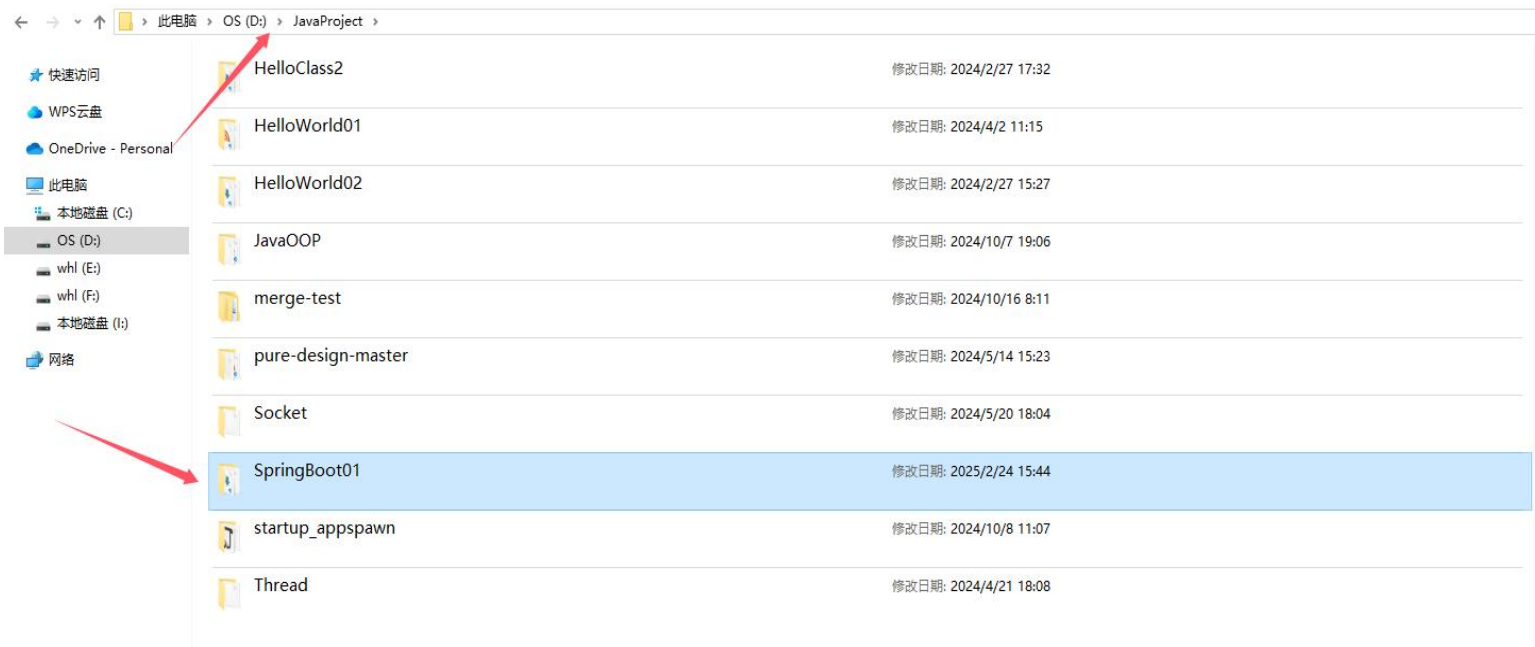


可见 https://blog.csdn.net/weixin_51309915/article/details/123349773

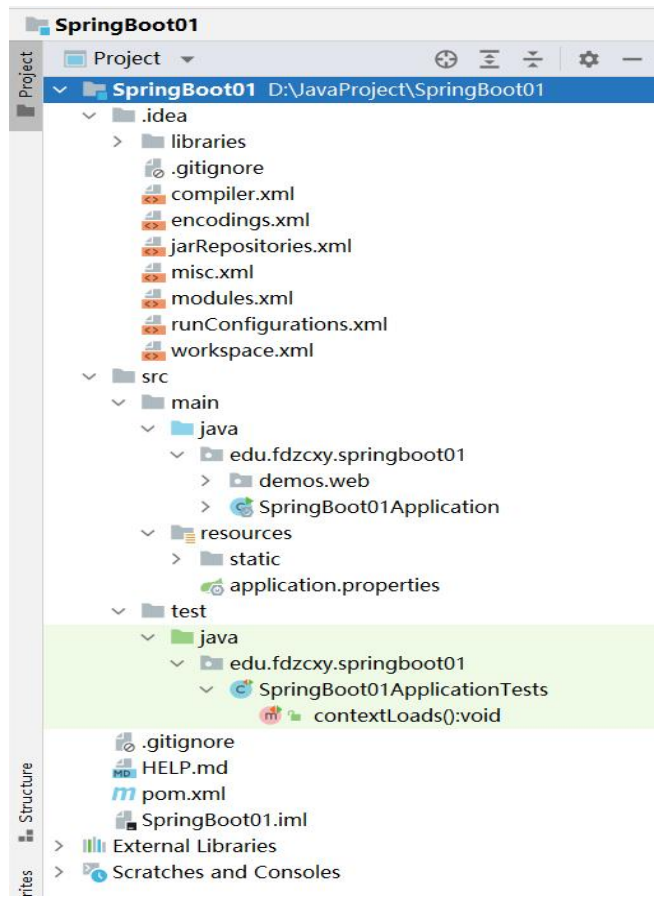


IDEA创建Spring Boot项目

创建完项目，就会在相应路径下生成对应的文件



Spring Boot项目目录结构



project

— src

#程序根目录

— main

#程序源文件目录

— java

#Java源文件目录

— __Application.java

#程序启动类

— resources

#资源文件夹目录

— static

#静态资源目录

— templates

#页面文件夹目录

— application.properties

#程序配置文件

— test

#单元测试文件夹

— __ApplicationTests.java

#测试类

— pom.xml

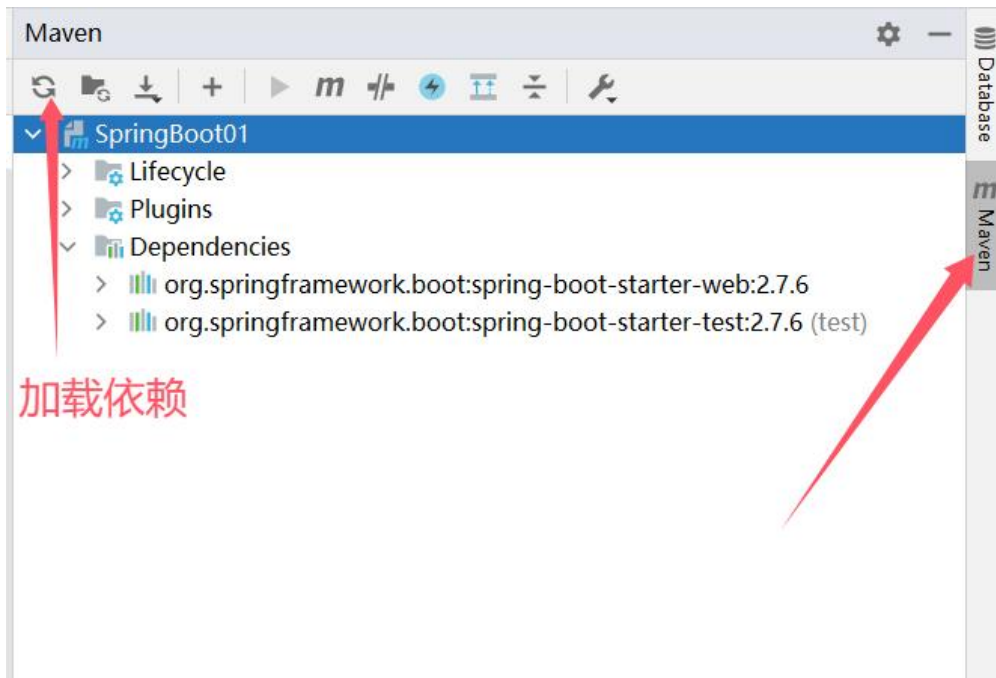
#Maven配置文件

https://blog.csdn.net/qq_34115869/article/details/107107107

Spring Boot项目目录结构



加载依赖

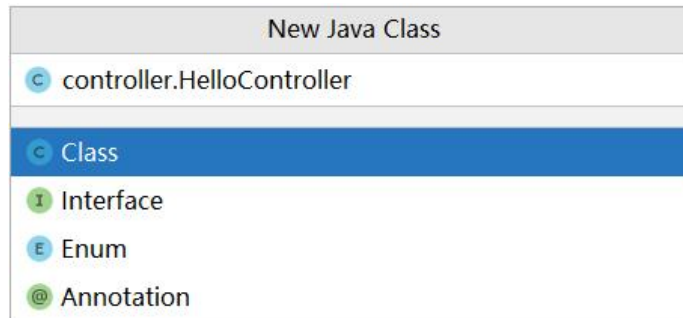
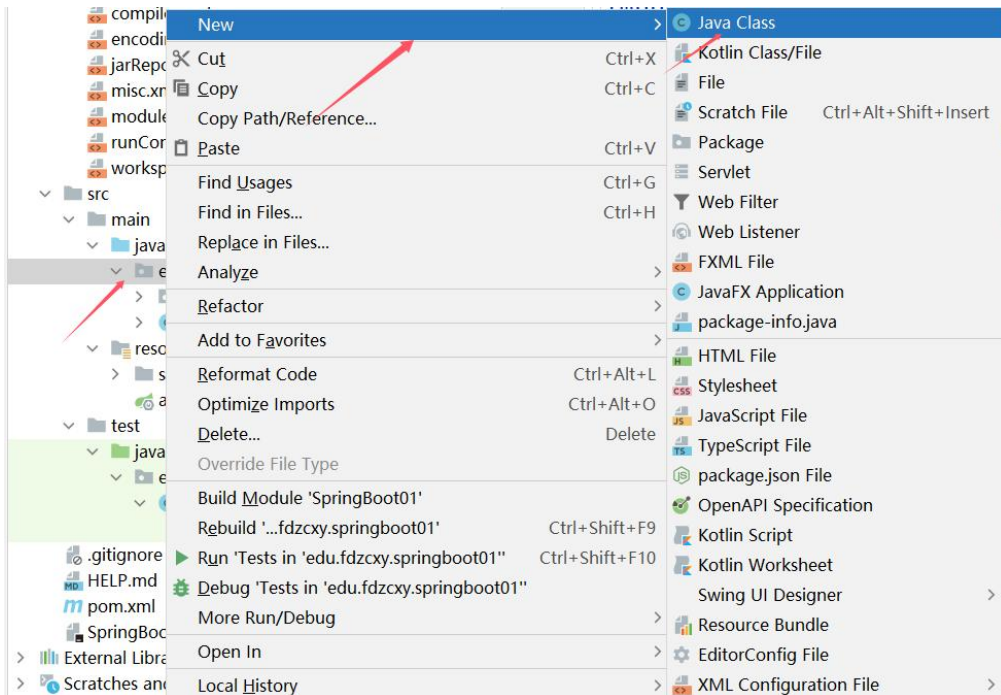


加载依赖



运行Spring Boot项目

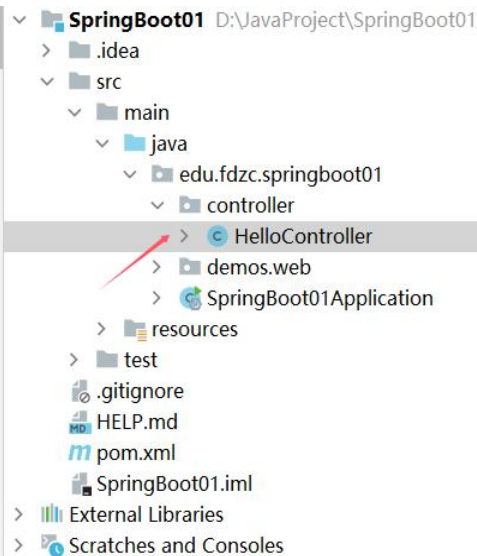
新建一个Java类HelloController



运行Spring Boot项目



编写HelloController



```
package edu.fdzc.springboot01.controller;

import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class HelloController {

    @GetMapping("/hello")
    public String Hello(){
        return "Hello SpringBoot";
    }
}
```

运行Spring Boot项目



运行main函数

The screenshot shows an IDE with the following components:

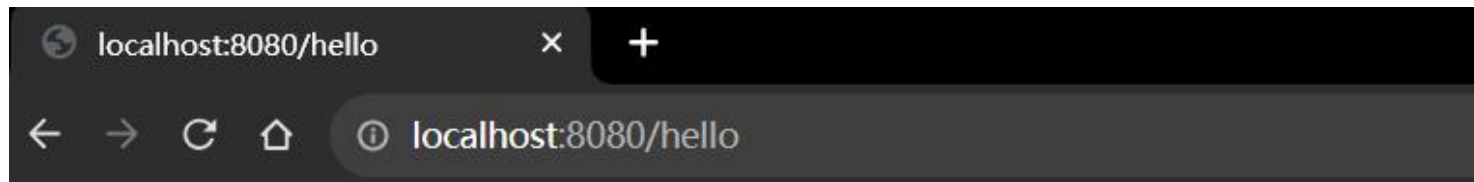
- Project Explorer:** Displays the project structure. The `SpringBoot01Application` class is selected under `edu.fdzc.springboot01 > controller`. A red arrow points to this class.
- Code Editor:** Shows the source code of `SpringBoot01Application`. The `main` method is highlighted in yellow. A red arrow points to the `Run` button (a green play icon) next to the `main` method.
- Run Configuration Dialog:** A dropdown menu is open, showing options for running the application. The first option, `Run 'SpringBoot01Applicat...' Ctrl+Shift+F10`, is selected.
- Run Console:** Displays the output of the application. The log shows the following messages:

```
main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplica
main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initializat
main] o.s.b.a.w.s.WelcomePageHandlerMapping : Adding welcome page: class path resource
main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) v
main] e.f.s.SpringBoot01Application : Started SpringBoot01Application in 1.55s
```



运行Spring Boot项目

在浏览器输入 `http://localhost:8080/hello`



Hello SpringBoot01!



Spring Boot项目讲解

pom.xml 项目依赖管理，主要分为四部分

- 项目的描述信息
- 项目依赖配置信息
- 构建时需要的公共变量
- 构建配置

详细见

https://blog.csdn.net/Black_Customer/article/details/124247032



Spring Boot项目讲解

application.properties 配置文件采用key=value的格式，提供了一种灵活的方式，使开发人员可以根据需要来配置应用程序的各种属性，而无需修改代码。示例如下。

```
1  #配置项目端口号
2  server.port=8080
3
4  #数据库 url
5  spring.datasource.url=jdbc:mysql://127.0.0.1:3306/testdb?characterEncoding=u
6
7  #数据库用户名
8  spring.datasource.username=root
9
10 #数据库密码
11 spring.datasource.password=123
```



Spring Boot项目讲解

注解

@SpringBootApplication

@RestController



作业

1.客户端服务端交互

2.软件安装，运行程序，在页面上显示自己的名字