

# 软件设计与系统架构

---

## 什么是Spring?

---

### 什么是微服务?

SSM: Spring IOC框架、SpringMVC Web框架、MyBatis 数据层框架

Spring Boot:

Maven: 依赖库管理工具, 编译、打包、发布

头疼问题: 改使用哪个类库、哪个版本?

问题: .gitignore文件的作用?

Git工具: 版本控制: 管理你的代码的编写历史

动手: 完成第一个SpringBoot程序

## 第二课

---

IoC: 控制反转, 将使用对象和创建对象分离, 解耦

MVC: Model、View、Controller 模型、视图、控制器

ORM: 对象关系映射 Object Relation Mapping 数据库的记录和程序的类对象的映射

热部署: 在程序运行期间修改了代码, 不需要再重新启动程序

单元测试: spring-boot-start-test

```
package com.example.demo;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.RestController;

@SpringBootApplication//springboot启动类
@RestController
public class DemoApplication {

    public static void main(String[] args) { //入口程序
        SpringApplication.run(DemoApplication.class, args);
    }

    @GetMapping("/hello")//访问地址, 表示使用Get方法访问, 并设置访问路径
```

```
public String hello(@RequestParam(value = "name", defaultValue = "world") String name) {  
    return String.format("Hello %s!", name);  
}
```

## lombok

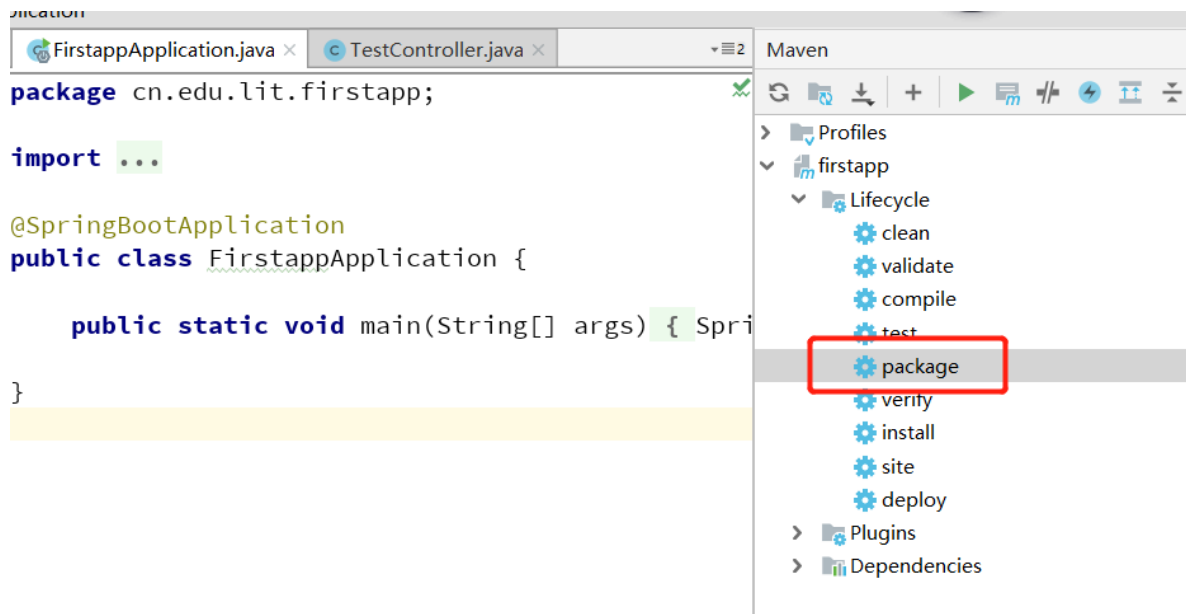
简化数据类的代码，提高程序的可读性，开发效率

## @RestController

表示提供Restful Web Service

什么是Restful?

## 如何打包运行项目



运行jar包: D:\study\springboot-bucket\firstapp\target> java -jar .\firstapp-0.0.1-SNAPSHOT.jar

## 做一做

创建一个新的springboot程序，提供RestFul服务，打包，运行。

<https://spring.io/guides/gs/rest-service/>

## 总结:

lombok 简化数据类

RestController 提供RestFul服务

几个注解

打包、运行

# 作业

在你的项目里如何消费别人提供的Rest服务？

<https://spring.io/guides/gs/consuming-rest/>

## 第三课

### 数据库操作

```
CREATE DATABASE springbootdata;
CREATE TABLE `t_article` (
  `id` int(20) NOT NULL AUTO_INCREMENT COMMENT '文章id',
  `title` varchar(200) DEFAULT NULL COMMENT '文章标题',
  `content` longtext COMMENT '文章内容',
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8;
INSERT INTO `t_article` VALUES ('1', 'Spring Boot基础入门', '从入门到精通讲解...');
INSERT INTO `t_article` VALUES ('2', 'Spring Cloud基础入门', '从入门到精通讲解...');
```

1.浏览器的请求，怎么到了指定的函数？怎么返回了JSON格式？

```
@RestController//指明此类提供RestFul风格的服务
public class BookController {
    @Autowired
    BookService bookService;
    @GetMapping("/books")//指明响应请求的url
    public List<Book> getAll(String name){
        return bookService.list(name);
    }
}
```

2.声明的BookService变量，如何被赋值了BookServiceImpl对象？

```
public interface BookService{}

@Service//注解：注明此类是一个服务类，自动生成服务类对象
public class BookServiceImpl implements BookService

@Autowired//自动注入此对象
BookService bookService;
```

3.如何使用MyBatisPlus从数据库获取数据？

```
@Data
@NoArgsConstructor
@AllArgsConstructor
@TableName("t_article")//注解，注明了此数据类对应的数据表
public class Book {
    int id;
    String title;
```

```
String content;
}

public interface BookMapper extends BaseMapper<Book> {
} // 给数据类写一个对应的Mapper接口

@MapperScan(basePackages = {"com.example.demo.data.mapper"}) // 注解：扫描Mapper对象
public class DemoApplication {
```

#### 4. 如何配置数据库连接信息和服务接口

```
// application.properties
spring.datasource.url=jdbc:mysql://localhost/springbootdata?
characterEncoding=UTF-8&serverTimezone=Asia/Shanghai&allowMultiQueries=true
spring.datasource.username=root
spring.datasource.password=root
spring.datasource.driver=com.mysql.cj.jdbc.Driver
server.port=8088
```

### Question: 配置的作用

证明一点：通过约定简化编程过程

### 引入MyBatisPlus依赖

```
<dependency>
  <groupId>com.baomidou</groupId>
  <artifactId>mybatis-plus-boot-starter</artifactId>
  <version>3.3.1</version>
</dependency>
```