乐字节教育高级架构课程

正所谓“授人以鱼不如授人以渔”，你们想要的 Java 学习资料来啦！不管你是学生，还是已经步入职场的同行，希望你们都要珍惜眼前的学习机会，奋斗没有终点，知识永不过时。

扫描下方二维码即可领取

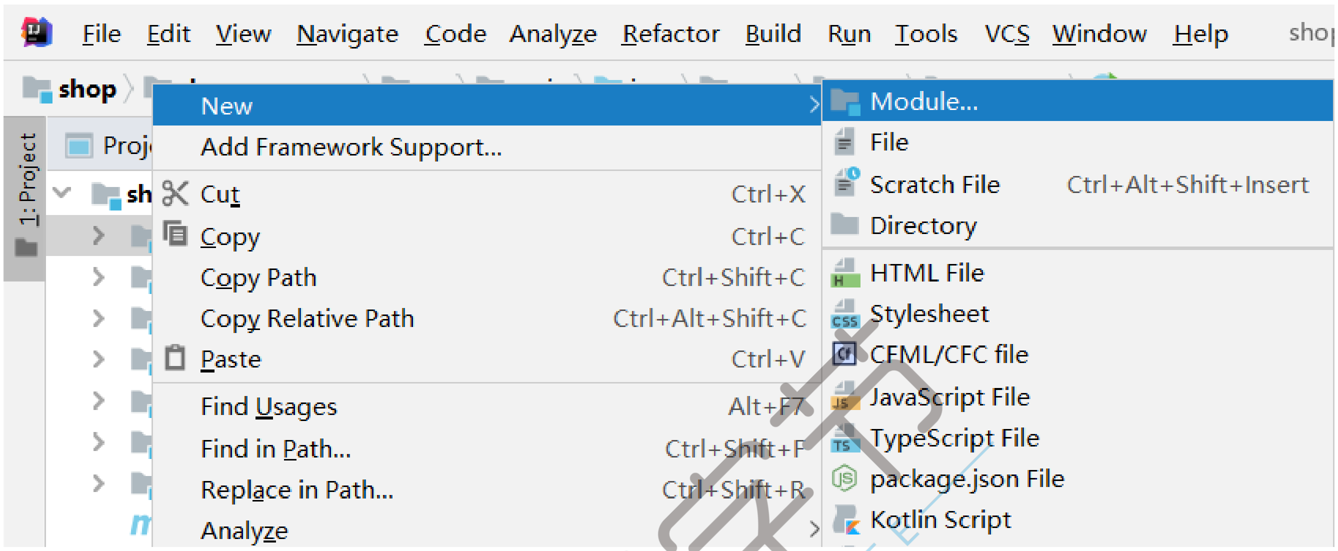
乐字节晓啡

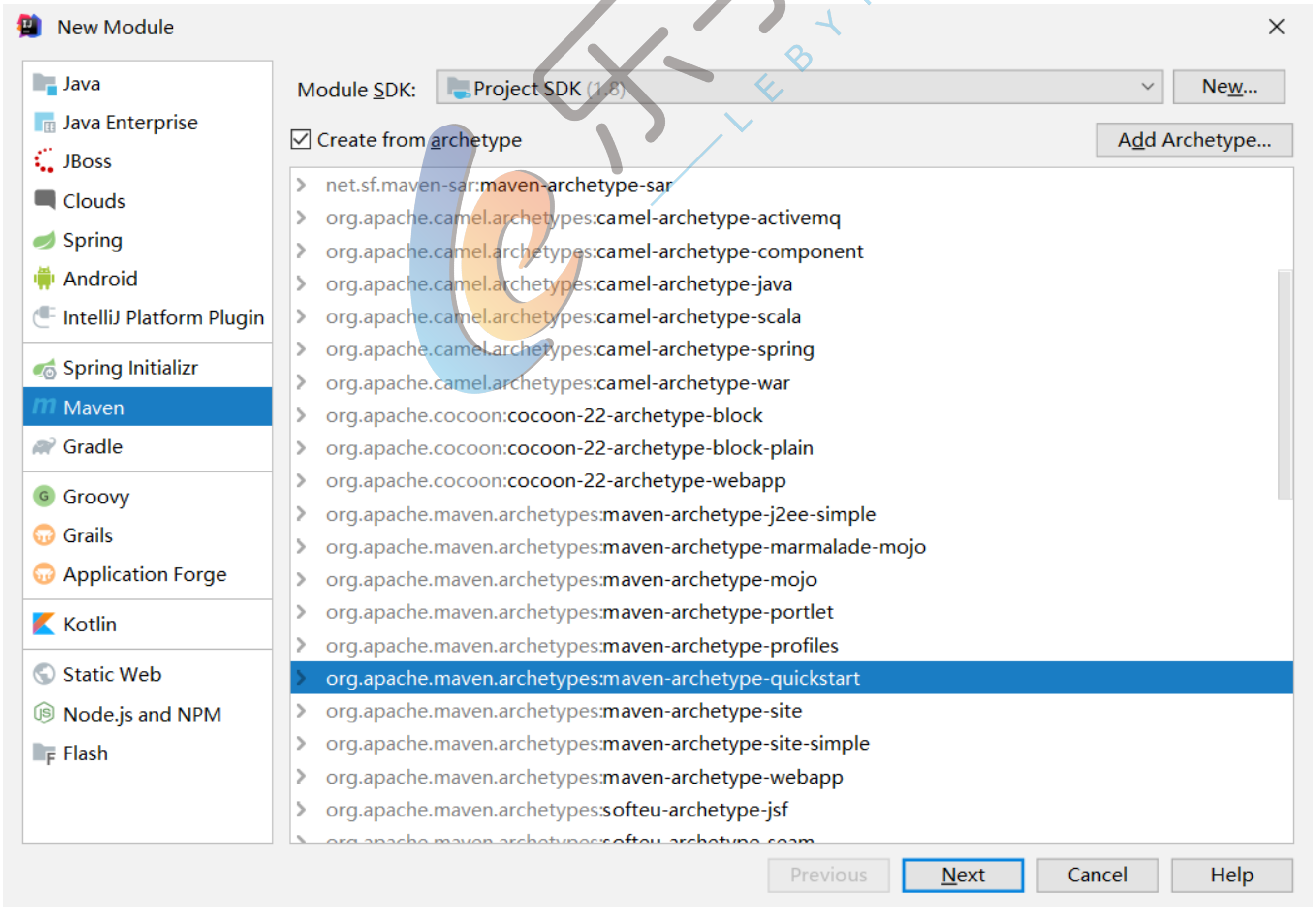
乐字节官方交流群

商城订单功能实现

商城订单系统环境搭建

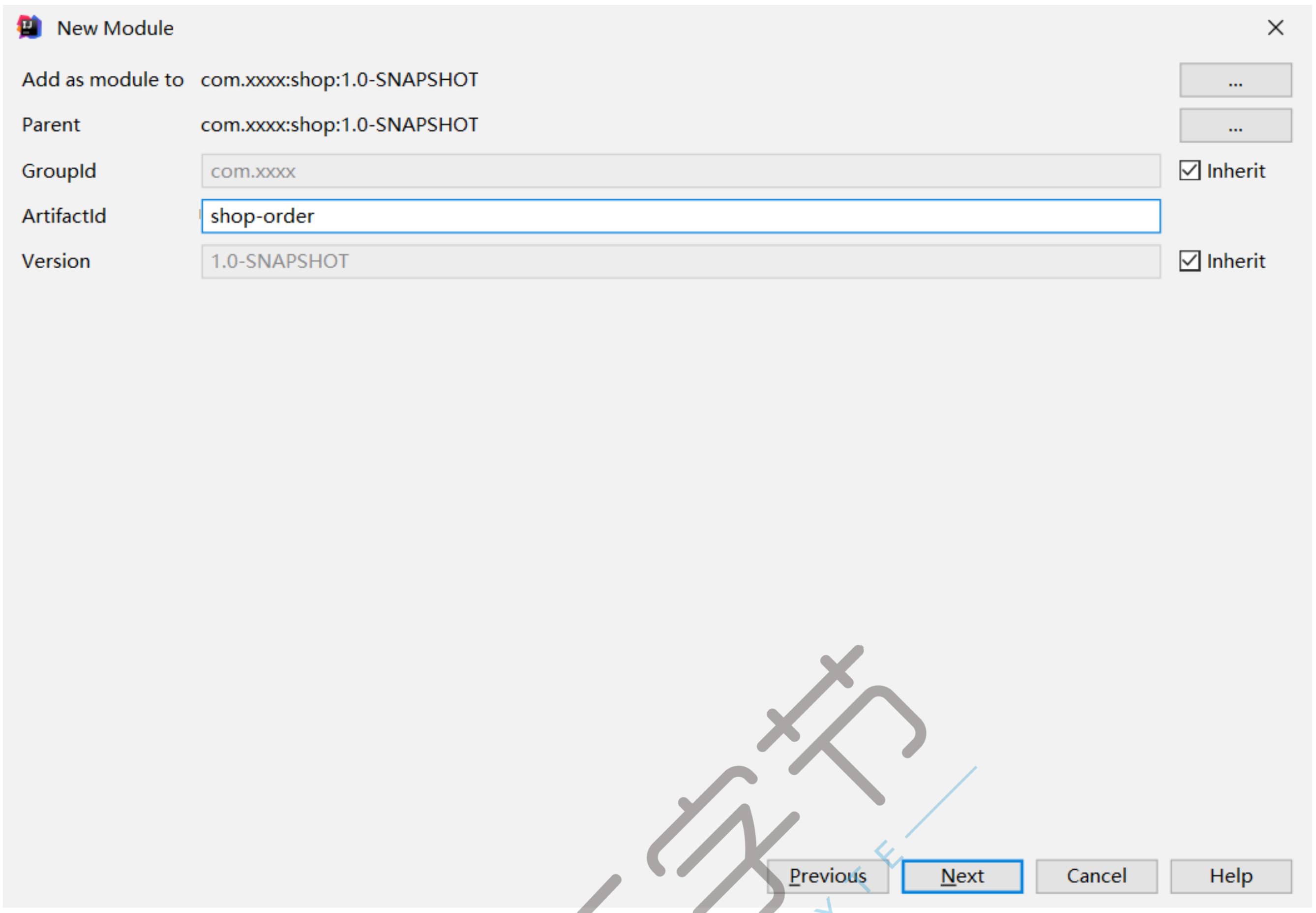
创建聚合类型项目shop-order，pom父模块




New Module
Java
Java Enterprise
JBoss
Clouds
Spring
Android
IntelliJ Platform Plugin
Spring Initializr
Maven
Gradle
Groovy
Grails
Application Forge
Kotlin
Static Web
Node.js and NPM
Flash
New...
Add Archetype...
net.sf.maven-sar:maven-archetype-sar
org.apache.camel.archetypes:camel-archetype-activemq
org.apache.camel.archetypes:camel-archetype-component
org.apache.camel.archetypes:camel-archetype-java
org.apache.camel.archetypes:camel-archetype-scala
org.apache.camel.archetypes:camel-archetype-spring
org.apache.camel.archetypes:camel-archetype-war
org.apache.cocoon:cocoon-22-archetype-block
org.apache.cocoon:cocoon-22-archetype-block-plain
org.apache.cocoon:cocoon-22-archetype-webapp
org.apache.maven.archetypes:maven-archetype-j2ee-simple
org.apache.maven.archetypes:maven-archetype-marmalade-mojo
org.apache.maven.archetypes:maven-archetype-mojo
org.apache.maven.archetypes:maven-archetype-portlet
org.apache.maven.archetypes:maven-archetype-profiles
org.apache.maven.archetypes:maven-archetype-quickstart
org.apache.maven.archetypes:maven-archetype-site
org.apache.maven.archetypes:maven-archetype-site-simple
org.apache.maven.archetypes:maven-archetype-webapp
org.apache.maven.archetypes:softeu-archetype-jsf
Previous
Next
Cancel
Help

Module SDK: Project SDK (1.8)

[✓] Create from archetype



New Module

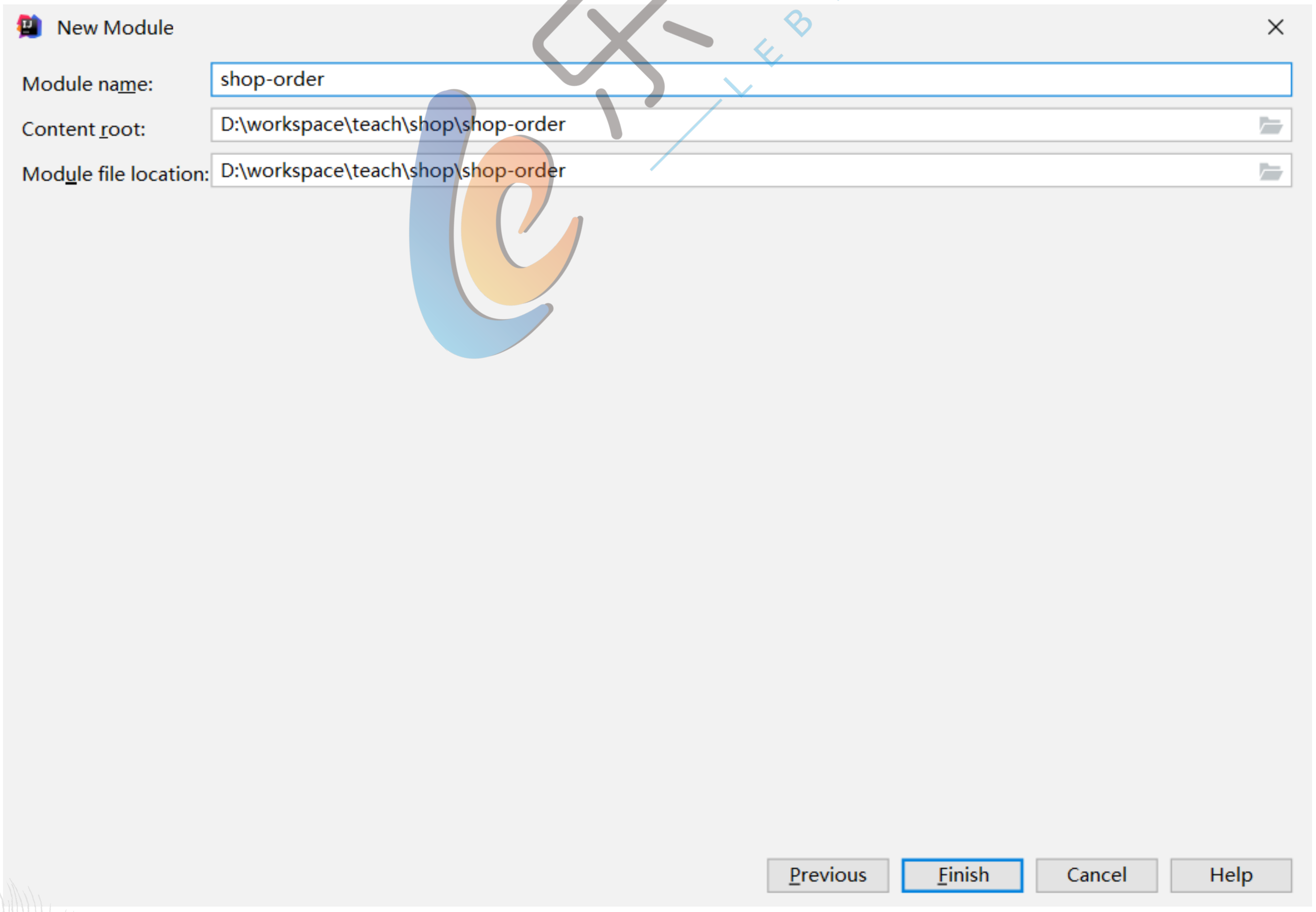
Add as module to com.xxxx:shop:1.0-SNAPSHOT

Parent com.xxxx:shop:1.0-SNAPSHOT

GroupId com.xxxx ☑ Inherit

ArtifactId shop-order

Version 1.0-SNAPSHOT ☑ Inherit



New Module

Module name: shop-order

Content root: D:\workspace\teach\shop\shop-order

Module file location: D:\workspace\teach\shop\shop-order

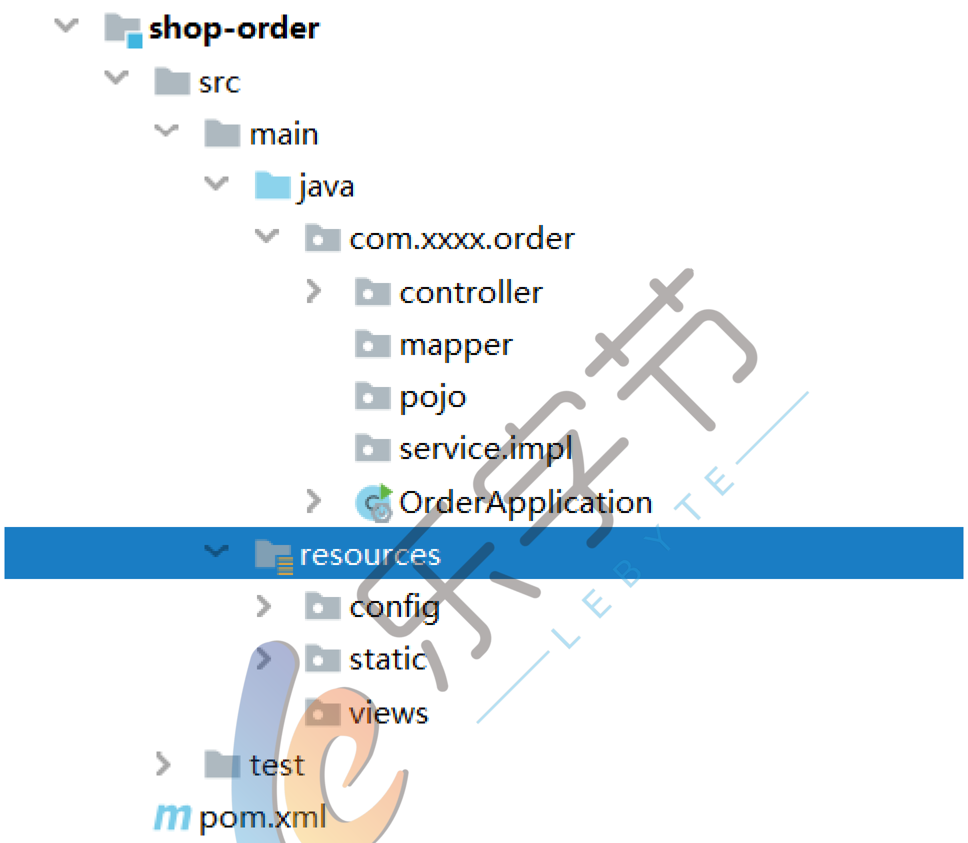
pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-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.xxxx</groupId>
   <artifactId>shop-order</artifactId>
   <version>1.0-SNAPSHOT</version>

   <!-- 继承 shop-parent 依赖 -->
   <parent>
      <groupId>com.xxxx</groupId>
      <artifactId>shop</artifactId>
      <version>1.0-SNAPSHOT</version>
   </parent>

   <properties>
      <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
      <maven.compiler.source>1.8</maven.compiler.source>
      <maven.compiler.target>1.8</maven.compiler.target>
   </properties>

   <dependencies>
      <!-- shop rpc 依赖 -->
      <dependency>
         <groupId>com.xxxx</groupId>
         <artifactId>shop-rpc</artifactId>
         <version>1.0-SNAPSHOT</version>
      </dependency>
      <!-- shop common 依赖 -->
      <dependency>
         <groupId>com.xxxx</groupId>
         <artifactId>shop-common</artifactId>
         <version>1.0-SNAPSHOT</version>
      </dependency>
      <!-- shop sso 依赖 -->
      <dependency>
         <groupId>com.xxxx</groupId>
         <artifactId>shop-sso</artifactId>
         <version>1.0-SNAPSHOT</version>
      </dependency>
      <!-- spring boot web 依赖 -->
      <dependency>
         <groupId>org.springframework.boot</groupId>
         <artifactId>spring-boot-starter-web</artifactId>
      </dependency>
      <!-- spring boot freemarker 依赖 -->
      <dependency>
         <groupId>org.springframework.boot</groupId>
         <artifactId>spring-boot-starter-freemarker</artifactId>
```

```
        </dependency>
        <!-- mybatis 依赖 -->
        <dependency>
            <groupId>org.mybatis.spring.boot</groupId>
            <artifactId>mybatis-spring-boot-starter</artifactId>
        </dependency>
        <!-- pagehelper 分页依赖 -->
        <dependency>
            <groupId>com.github.pagehelper</groupId>
            <artifactId>pagehelper-spring-boot-starter</artifactId>
        </dependency>
        <!-- mysql 数据库依赖 -->
        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
        </dependency>
        <!-- druid 连接池依赖 -->
        <dependency>
            <groupId>com.alibaba</groupId>
            <artifactId>druid</artifactId>
        </dependency>
        <!-- spring data redis 依赖 -->
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-data-redis</artifactId>
        </dependency>
        <!-- commons-pool2 对象池依赖 -->
        <dependency>
            <groupId>org.apache.commons</groupId>
            <artifactId>commons-pool2</artifactId>
        </dependency>
        <!--dubbo 依赖-->
        <dependency>
            <groupId>com.alibaba.spring.boot</groupId>
            <artifactId>dubbo-spring-boot-starter</artifactId>
        </dependency>
        <!-- zkClient 依赖 -->
        <dependency>
            <groupId>com.101tec</groupId>
            <artifactId>zkclient</artifactId>
            <exclusions>
                <exclusion>
                    <artifactId>slf4j-log4j12</artifactId>
                    <groupId>org.slf4j</groupId>
                </exclusion>
            </exclusions>
        </dependency>
        <!-- spring boot test 依赖 -->
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
            <exclusions>
```

```xml
                <exclusion>
                    <groupId>org.junit.vintage</groupId>
                    <artifactId>junit-vintage-engine</artifactId>
                </exclusion>
            </exclusions>
        </dependency>
    </dependencies>
</project>
```

目录结构



```
shop-order
  src
    main
      java
        com.xxxx.order
          controller
          mapper
          pojo
          service.impl
          OrderApplication
      resources
        config
        static
        views
    test
  pom.xml
```

application-dev.yml

```yaml
server:
  port: 9092                                          # 项目访问端口，默认 8080
  servlet:                                           # 项目访问路径，默认 /
    context-path: /shop-order
# Spring
spring:
  # 数据源
  datasource:
    driver-class-name: com.mysql.cj.jdbc.Driver
    url: jdbc:mysql://localhost:3306/shop?useUnicode=true&characterEncoding=UTF-
8&serverTimezone=Asia/Shanghai
    username: root
    password: root
    # 指定 druid 连接池以及 druid 连接池配置
    type: com.alibaba.druid.pool.DruidDataSource
    druid:
      initial-size: 1                       # 初始连接数
      max-active: 20                        # 最大连接数
      max-idle: 20                          # 最大空闲
      min-idle: 1                           # 最小空闲
      max-wait: 60000                       # 最长等待时间
  # freemarker 模板引擎
  freemarker:
    cache: false
    charset: UTF-8
    content-type: text/html;charset=UTF-8
    enabled: true
    suffix: .ftl
    template-loader-path: classpath:/views/
    # 配置模板里是否可以直接取request的属性 request是别名
    request-context-attribute: request
    # 配置将request和session中的键值添加到
    # AbstractTemplateView类的renderMergedOutputModel方法中的model这个Map参数中
    expose-request-attributes: true
    expose-spring-macro-helpers: true
    # 配置模板里是否可以直接取session的属性 true 是允许
    expose-session-attributes: true
    settings:
      tag_syntax: auto_detect               # 配置标签语法为自动，页面可以将 <> 改为 []，为了区别
html 标签
      template_update_delay: 0              # 模板更新时间，单位秒
      default_encoding: UTF-8               # 默认编码字符集
      output_encoding: UTF-8                # 模板输出编码字符集
      locale: zh_CN                         # 本地化配置
      date_format: yyyy-MM-dd               # 日期格式化
      time_format: HH:mm:ss                 # 时间格式化
      datetime_format: yyyy-MM-dd HH:mm:ss  # 日期时间格式化
      number_format: #.##                   # 数字格式化
      boolean_format: true,false            # boolean格式化
      # ignore,debug,html_debug,rethrow
      # 1.TemplateExceptionHandler.IGNORE_HANDLER简单地压制所有异常
      # 它对处理异常没有任何作用，也不会重新抛出异常，页面可以正常渲染，后台抛异常
      # 2.TemplateExceptionHandler.DEBUG_HANDLER打印堆栈信息和重新抛出异常。这是默认的异常控制器
      # 3.TemplateExceptionHandler.HTML_DEBUG_HANDLER和DEBUG_HANDLER相同
      # 但是可以格式化堆栈跟踪信息，HTML页面，建议使用它而不是DEBUG_HANDLER
      # 4.TemplateExceptionHandler.RETHROW_HANDLER简单重新抛出所有异常而不会做其他的事情
      # 5.使用自定义异常类实现TemplateExceptionHandler重写handleTemplateException方法
      template_exception_handler: html_debug
  # 文件上传
  servlet:
    multipart:
      max-file-size: 100MB                  # 设置单个上传文件的大小
      max-request-size: 1000MB               # 设置一次请求上传文件的总容量
  # redis 缓存
  redis:
    timeout: 10000ms                        # 连接超时时间
    host: 192.168.10.100                   # Redis服务器地址
    port: 6379                              # Redis服务器端口
    database: 0                             # 选择哪个库，默认0库
    lettuce:
      pool:
        max-active: 1024                      # 最大连接数，默认 8
        max-wait: 10000ms                     # 最大连接阻塞等待时间，单位毫秒，默认 -1
        max-idle: 200                         # 最大空闲连接，默认 8
        min-idle: 5                           # 最小空闲连接，默认 0
  # Dubbo
  dubbo:
    #开启dubbo服务
    server: true
    # 提供方应用信息，用于计算依赖关系
    application:
      name: rpc-consumer
    # 使用 zookeeper 注册中心暴露服务地址
    registry:
      address: zookeeper://192.168.10.100:2181?
backup=192.168.10.100:2182,192.168.10.100:2183
    # 用 dubbo 协议在 20880 端口暴露服务
#    protocol:
#      name: dubbo
#      port: 20880
    # 扫描需要暴露的服务接口包
    scan:
      base-packages: com.xxxx.order

# MyBatis
mybatis:
  # 配置 MyBatis数据返回类型别名(默认别名是类名)
  type-aliases-package: com.xxxx.order.pojo
  # 配置 MyBatis Mapper 映射文件
  mapper-locations: classpath:mapper/*.xml

# Mybatis SQL 打印(方法接口所在的包，不是 Mapper.xml 所在的包)
logging:
  level:
    com.xxxx.manager.mapper: debug

# Redis Key
#用户票据key
user.ticket: user:userTicket

# shop-portal系统url
shop.portal.url: http://localhost:9091/shop-portal/

#关闭Elasticsearch健康检查
management:
  health:
    elasticsearch:
      enabled: false
```

OrderApplication.java

```
package com.xxxx.order;
```

```java
import org.mybatis.spring.annotation.MapperScan;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

/**
 * 订单启动类
 *
 * @author zhoubin
 * @since 1.0.0
 */
@SpringBootApplication
//开启dubbo
@EnableDubboConfiguration
// 扫描 mapper 接口
@MapperScan("com.xxxx.order.mapper")
public class OrderApplication {
    public static void main(String[] args) {
        SpringApplication.run(OrderApplication.class,args);
    }
}
```

项目运行测试

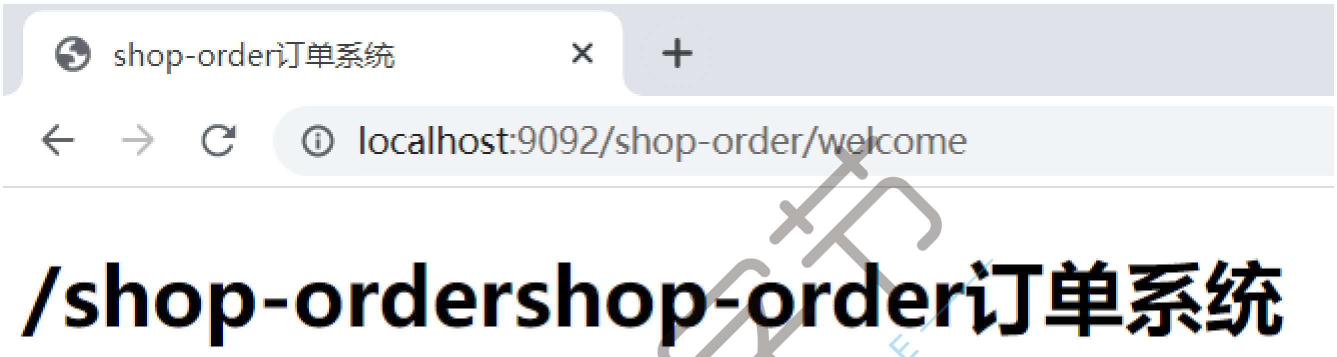
PageController.java

```java
package com.xxxx.order.controller;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;

/**
 * 跳转页面
 *
 * @author zhoubin
 * @since 1.0.0
 */
@Controller
public class PageController {
    /**
     * 页面跳转
     * @param page
     * @return
     */
    @RequestMapping("/{page}")
    public String page(@PathVariable String page) {
        return page;
    }
}
```

welcome.ftl

```
<!-- 设置项目根路径全局变量 -->
<#assign ctx=request.contextPath/>
<!DOCTYPE html>
<html>
<head>
    <title>shop-order订单系统</title>
</head>
<body>
<h1>${ctx}shop-order订单系统</h1>
</body>
</html>
```

测试结果如下：


shop-order订单系统

localhost:9092/shop-order/welcome

/shop-ordershop-order订单系统