

服务器端搭建

SSM框架: Spring+SpringMVC+MyBatis SpringBoot工具

创建所有项目都是基于maven创建。 maven项目管理(清理-编译-测试-打包-部署)和依赖(jar)管理的工具。

解压maven

```
D:\tools\apache-maven-3.8.8
```

配置maven

打开D:\tools\apache-maven-3.8.8\conf\settings.xml

```
-- 本地仓库 首先创建一个本地仓库文件夹 D:\tools\repository2
<!-- 配置maven的本地仓库的地址 -->
<localRepository>D:\tools\repository2</localRepository>

-- 下载jar服务器镜像改为国内
<mirror>
  <id>aliyunmaven</id>
  <mirrorOf>*</mirrorOf>
  <name>阿里云公共仓库</name>
  <url>https://maven.aliyun.com/repository/public</url>
</mirror>
```

maven集成Idea

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
  https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <!-- 当前项目坐标 -->
  <groupId>com.haiyang</groupId>
  <artifactId>mall-server-021</artifactId>
  <version>1.0.0</version>

  <packaging>jar</packaging>

  <!-- 运行参数 -->
  <properties>
    <java.version>1.8</java.version>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
    <!-- springboot版本 -->
    <spring-boot.version>2.3.7.RELEASE</spring-boot.version>
  </properties>

  <!-- 依赖 -->
  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
```

```

        <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
    <!-- MySQL数据库驱动-->
    <dependency>
        <groupId>mysql</groupId>
        <artifactId>mysql-connector-java</artifactId>
        <scope>runtime</scope>
    </dependency>
    <!-- Lombok插件-->
    <dependency>
        <groupId>org.projectlombok</groupId>
        <artifactId>lombok</artifactId>
        <optional>true</optional>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-test</artifactId>
        <scope>test</scope>
        <exclusions>
            <exclusion>
                <groupId>org.junit.vintage</groupId>
                <artifactId>junit-vintage-engine</artifactId>
            </exclusion>
        </exclusions>
    </dependency>
</dependencies>
<dependencyManagement>
    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-dependencies</artifactId>
            <version>${spring-boot.version}</version>
            <type>pom</type>
            <scope>import</scope>
        </dependency>
    </dependencies>
</dependencyManagement>
<build>
    <plugins>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-compiler-plugin</artifactId>
            <version>3.8.1</version>
            <configuration>
                <source>1.8</source>
                <target>1.8</target>
                <encoding>UTF-8</encoding>
            </configuration>
        </plugin>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
            <version>2.3.7.RELEASE</version>
        </plugin>
    </plugins>
</build>
</project>

```

```
server:
  port: 10001
  servlet:
    context-path: /
```

注意:

yml文件配置属性节点层次必须对其。
属性值前面必须有一个空格

项目启动，浏览器访问:

<http://localhost:10001/index>

如果设置 context-path: /api

<http://localhost:10001/api/index>

启动类

包: 包名全部小写。包是分层级。

例如: com.haiyang

对应物理磁盘上就是文件夹。

包名--域名倒置 cctv.com baidu.com

分层开发:

```
vue界面
|
控制层:  处理前端发来的请求. vue请求( 登录 ).      com.haiyang.controller
|
业务逻辑层  项目具体功能和流程  登录方法      com.haiyang.service
|
数据访问层:  对数据库CRUD操作      com.haiyang.mapper
|
MySQL
```

```
package com.haiyang;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class MallServer021Application {
    public static void main(String[] args) {
        SpringApplication.run(MallServer021Application.class, args);
    }
}
```

创建IndexController控制器

```
package com.haiyang.controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController    //当前类是一个控制器
@RequestMapping("/index")
public class IndexController {
```

```
//处理请求方法
//方法返回数据 就是响应给前端的数据
@RequestMapping("/test")
public String test(){
    return "Hello SpringBoot";
}
}
```