### Mybatis+SpringBoot 环境搭建

1. 引入MySQL的驱动

|  |
| --- |
| <dependency>  <groupId>mysql</groupId>  <artifactId>mysql-connector-java</artifactId>  </dependency> |

1. 引入MyBatis和SpringBoot集成的包

|  |
| --- |
| <!-- https://mvnrepository.com/artifact/org.mybatis.spring.boot/mybatis-spring-boot-starter -->  <dependency>  <groupId>org.mybatis.spring.boot</groupId>  <artifactId>mybatis-spring-boot-starter</artifactId>  <version>2.1.4</version>  </dependency> |

1. 配置application.properties

|  |
| --- |
| server.port=8080  spring.datasource.url=jdbc:mysql://localhost:3306/sales?serverTimezone=GMT%2B8  spring.datasource.username=root  spring.datasource.password=123456  spring.datasource.driverClassName=com.mysql.cj.jdbc.Driver  mybatis.type-aliases-package=org.softcits.cn.model  mybatis.mapper-locations=classpath:mapper/\*Mapper.xml |

1. 創建mapper接口

|  |
| --- |
| package org.softcits.cn.mapper;  import org.apache.ibatis.annotations.Mapper;  import org.softcits.cn.model.Occupation;  @Mapper  public interface OccupationMapper {  Occupation getFirstOccupation();    } |

1. 创建mapper xml

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <!DOCTYPE mapper  PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"  "http://mybatis.org/dtd/mybatis-3-mapper.dtd">  <mapper namespace="org.softcits.cn.mapper.OccupationMapper">  <select id="getFirstOccupation" resultType="org.softcits.cn.model.Occupation">  select Name, Occupation from occupations limit 0,1  </select>  </mapper> |

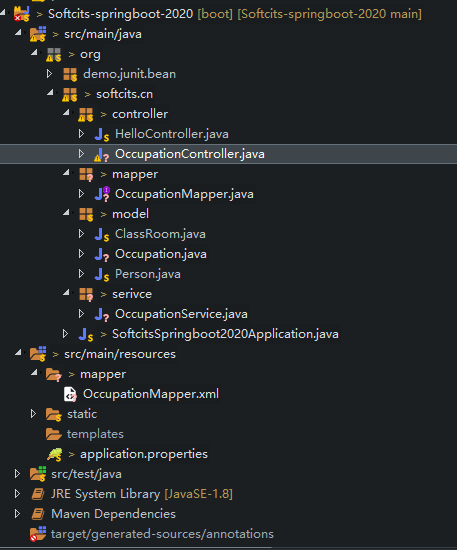
1. 创建Service

|  |
| --- |
| package org.softcits.cn.serivce;  import org.softcits.cn.mapper.OccupationMapper;  import org.softcits.cn.model.Occupation;  import org.springframework.beans.factory.annotation.Autowired;  import org.springframework.stereotype.Service;  @Service  public class OccupationService {    @Autowired  private OccupationMapper occupationMapper;    public Occupation getFirstOccupation() {  Occupation occupation = occupationMapper.getFirstOccupation();  return occupation;  }  } |

1. 创建Controller

|  |
| --- |
| package org.softcits.cn.controller;  import org.softcits.cn.mapper.OccupationMapper;  import org.softcits.cn.model.Occupation;  import org.softcits.cn.serivce.OccupationService;  import org.springframework.beans.factory.annotation.Autowired;  import org.springframework.web.bind.annotation.GetMapping;  import org.springframework.web.bind.annotation.RestController;  @RestController  public class OccupationController {    @Autowired  private OccupationService occupationService;    @GetMapping("/occupation/getfirst")  public Occupation getFirstOccupation() {  return occupationService.getFirstOccupation();  }  } |

最终项目结构：



集合查询，resultType也要写集合内存储的类型

|  |
| --- |
| <!-- List<Occupation> getAllOccupations(); -->  <select id="getAllOccupations" resultType="org.softcits.cn.model.Occupation">  SELECT NAME, Occupation FROM occupations  </select> |

条件查询

|  |
| --- |
| <!-- Occupation getOccupationByName(String name); -->  <select id="getOccupationByName" resultType="org.softcits.cn.model.Occupation"  parameterType="java.lang.String">  SELECT NAME, Occupation FROM occupations WHERE NAME = #{name, jdbcType=VARCHAR}  </select> |

多参数情况：

1. 接口函数里面的参数和xml参数同名

|  |
| --- |
| <!-- int updateOccupationByName(String name, String occp); -->  <update id="updateOccupationByName">  UPDATE occupations SET Occupation=#{occp, jdbcType=VARCHAR} WHERE NAME = #{name, jdbcType=VARCHAR}  </update> |

1. 使用@Param(value=”xxxxx”)给参数起别名

|  |
| --- |
| <!-- int updateOccupationByName(@Param(value="\_name")String name, @Param(value="\_occp")String occp); -->  <update id="updateOccupationByName">  UPDATE occupations SET Occupation=#{\_occp, jdbcType=VARCHAR} WHERE NAME = #{\_name, jdbcType=VARCHAR}  </update> |

1. #{param1}, #{param2} .....

|  |
| --- |
| <!-- int insertOccupation(String name, String occp); -->  <insert id="insertOccupation">  INSERT INTO occupations(NAME, Occupation) VALUES ( #{param1}, #{param2});  </insert> |

1. 使用map集合

|  |
| --- |
| <!-- int deleteOccupation(Map<String, Object> map); -->  <delete id="deleteOccupation" parameterType="java.util.Map">  DELETE FROM occupations WHERE NAME=#{name} AND Occupation = #{occp}  </delete> |

1. 传入java bean

|  |
| --- |
| <!-- Occupation getByNameAndOccupation(Occupation occp); -->  <select id="getByNameAndOccupation" resultType="org.softcits.cn.model.Occupation"  parameterType="org.softcits.cn.model.Occupation">  select name, Occupation from occupations where Name=#{Name, jdbcType=VARCHAR} and Occupation=#{Occupation, jdbcType=VARCHAR}  </select> |