

**Java方向**

**综合实训**

**实验报告书**

专业： 网络工程（NIIT项目）

班级： 2016030203

学号： 201603020325

姓名： 唐永海

2023年 9 月 14 日

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Major  专业 | 网络工程(NIIT) | | Course name课程名称 | | Java方向综合实训 | | |
| Guide Teacher 指导老师 | 廖昌旺 | Place 实验地点 | | 11207 | | Date 实验日期 | 2023.9.14 |
| Student Name 学生姓名 | 唐永海 | Batch 班　级 | | Java1 | | Student Id 学　号 | 201603020325 |
| Activity  &  Exercises 活动与练习 | Sprint03-歌手管理模块实训报告：Challenge,完成歌手管理中的歌曲管理功能，可以正常查看歌手的歌曲列表页面。 | | | | | | |
| Details for  Activity  活动相关细节 | Activity活动: Sprint03-歌手管理模块实训报告：Challenge,完成歌手管理中的歌曲管理功能，可以正常查看歌手的歌曲列表页面。 | | | | | | |
| Request要求:Challenge,完成歌手管理中的歌曲管理功能，可以正常查看歌手的歌曲列表页面。  (Problem Statement) 关于问题的描述 | | | | | | |
| Solution解决:   1. type 建Song.java   @TableName(value = "song") @Data public class Song {   @TableId(type = IdType.*AUTO*)  private Integer id;   private Integer singerId;   private String name;   private String introduction;   @TableField(fill = FieldFill.*INSERT*)  private Date createTime;   @TableField(fill = FieldFill.*INSERT\_UPDATE*)  private Date updateTime;   private String pic;   private String lyric;   private String url;   @Override  public String toString() {  return ToStringBuilder.*reflectionToString*(this);  } }   1. type 建SongMapper 接口   */\*\*  \* admin-歌手管理-歌曲管理（mapper层接口）  \*/* public interface SongMapper extends BaseMapper<Song> { }   1. type 在resources/mapper下建SongMapper.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="com.example.onlinemusicservice.mapper.SongMapper">  <resultMap id="BaseResultMap" type="com.example.onlinemusicservice.model.domain.Song">  <id column="id" property="id" jdbcType="INTEGER" />  <result column="singer\_id" property="singerId" jdbcType="INTEGER" />  <result column="name" property="name" jdbcType="VARCHAR" />  <result column="introduction" property="introduction" jdbcType="VARCHAR" />  <result column="create\_time" property="createTime" jdbcType="TIMESTAMP" />  <result column="update\_time" property="updateTime" jdbcType="TIMESTAMP" />  <result column="pic" property="pic" jdbcType="VARCHAR" />  <result column="url" property="url" jdbcType="VARCHAR" />  </resultMap>   <resultMap id="ResultMapWithBLOBs" type="com.example.onlinemusicservice.model.domain.Song" extends="BaseResultMap">  <result column="lyric" property="lyric" jdbcType="LONGVARCHAR" />  </resultMap>   <sql id="Base\_Column\_List">  id, singer\_id, name, introduction, create\_time, update\_time, pic  </sql>  <sql id="Blob\_Column\_List">  lyric  </sql>   <select id="selectAllSongIds" resultType="java.lang.Integer">  select id from song  </select>  </mapper>  4.type 在service层建SongService接口  public interface SongService extends IService<Song> {   */\*\*  \* 根据歌手ID查询歌曲列表  \* @param singerId  \* @return  \*/* public R selectSongsBySingerId(int singerId); }   1. type在service/Impl下建SongServiceImpl接口实现类   @Service public class SongServiceImpl extends ServiceImpl<SongMapper, Song> implements SongService {  @Autowired  private SongMapper songMapper;  @Override  public R selectSongsBySingerId(int singerId) {  *// 根据歌手ID查询歌曲 select \* from song where singer\_id = 1  // 定义查询条件  //调用Mapper去完成基础的查询功能  //1.使用QueryWrapper设置查询条件* QueryWrapper<Song> queryWrapper = new QueryWrapper<>();  queryWrapper.eq("singer\_id",singerId);  *// 调用mybatis提供的查询方法  // 2.调用mybatis提供的查询列表的方法* return R.*success*("查询成功",songMapper.selectList(queryWrapper));  } }  5.type 在Controller层建SongController.java  @RestController public class SongController {   @Autowired  private SongService songService;   */\*\*  \* 根据歌手Id查询歌曲列表  \* @param  \* @return  \*/* @GetMapping("/song/singer/detail")  public R selectSongsBySingerId(@RequestParam int singerId) {  return songService.selectSongsBySingerId(singerId);  } } | | | | | | |
| Source code源代码(代码截图): | | | | | | |
| Running Result运行结果: | | | | | | |
| Experimental  Experience  实验心得 | Problems encountered遇到的问题:  无 | | | | | | |
| Knowledge mastered掌握的知识: | | | | | | |
| Comments  评语 |  | | | | | | |
| Score 成绩 |  | | | | | | |