## 数据库表的构建

数据库的构建和java代码中的实体类的构建有很大的联系。数据库中表的名称和实体类的类名对应，表中属性名和java实体类的属性名对应。

注：java的类名使用的为驼峰命名法，如UserRole转化为数据库中的表名就是user\_role

## Dao层代码书写

**BaseDao**（包括**BaseDao.xml**文件）代码书写

介绍：数据库一些基本的增删改查。这个接口是一个公共的接口其他的dao接口都可以继承这个接口。

接口：

1，public void add(@Param("tableName") String tableName, @Param("objects")Object...objects );

向数据库中添加一条数据，objects是数据表对应的value值。

对应的sql语句为：

insert into ${tableName} values(value1,value2,…)

2，public void delete(@Param("tableName") String tableName,@Param("id") Integer id);

删除对应表中的一条记录，id 为该表的主键 索引。

对应的sql语句为：

delete from ${tableName} where id=#{id}

3，public void update(@Param("tableName") String tableName, @Param("id") Integer id, @Param("objects")Object[] objects);

更新对应表中的id对应的那条记录，objects对应 的是set语句集合，如set telephone=’123’

update ${tableName} ${obj1},${obj2}… where id=#{id}

4, public Map<Object, Object> select(@Param("tableName") String tableName, @Param("id") Integer id);

查找对应表中对应id的一条数据，返回键值对。

对应sql语句为

select \* from ${tableName} where id=#{id}

5，public List<Map<Object, Object>> selectAll(@Param("tableName") String tableName);

查找对应表中的所有数据

对应sql语句为

select \* from ${tableName}

6，public void addForNotMatch(@Param("tableName") String tableName,@Param("fieldNames")Object[] fieldNames, @Param("fieldValues")Object[] fieldValues);

向对应表中添加一条记录，指定对应属性的值

insert into ${tableName} () values(value1,value2,…)

## Service层代码书写

service层的代码位于package njit.service 中，其中XxxService接口定义了对应XxxServiceImp应该实现的方法。

**BaseService<T>**代码的书写，定义的接口如下，使方法和Dao层对应，此处不在赘述,其中泛型实现类应该传入一个对应的模型类，所有其他Service层接口都应该继承这个接口，并传入对应的模型。

@Deprecate

public void add(T t);

public int addForNotMatch(Object[] fieldNames, Object[] fieldValues);

public void delete(int id);

public void update(T t);

public T select(int id);

public List<T> selectAll();

**BaseServiceImp<T>** 实现了**BaseService<T>**中的方法，并定义一个抽象的方法，这个方法的作用是返回各自模型对应Dao层操作数据库的 对象

public abstract BaseDao getBaseDao();

## Controller层代码书写

**IndexController** 定义了所有视图的访问链接

**AdminController** 定义了所有管理员操作的访问请求地址和操作 如对用户的增删改查

**BookController** 定义了所有预定相关的访问请求地址和操作，如用户的预定请求

**UserController** 定义了所有和用户相关的访问请求地址和操作，如用户的登陆注册找回密码

## 参考文档地址

mybatis中文文档 <http://www.mybatis.org/mybatis-3/zh/getting-started.html>

mybatis 分页插件的用法<https://github.com/pagehelper/Mybatis-PageHelper/blob/master/wikis/zh/HowToUse.md>

boostrap中文网 <http://www.bootcss.com/>

boostrap 日期选择插件[http://www.bootcss.com/p/bootstrap-datetimepicker/demo.htm#demo\_advanced](http://www.bootcss.com/p/bootstrap-datetimepicker/demo.htm" \l "demo_advanced)

阿里巴巴矢量标签库 <https://www.iconfont.cn/manage/index?manage_type=myprojects&projectId=1039301>

## 小bug

1，模糊查询，点击下一页显示的就不再是结果数据

2，NoticeService 中的selectNoticeByPages第2行PageInfo的构造方法不能使用this.selectAll()，否则查不出来分页的效果。（我感觉可能是这样的，这个分页插件是在Dao层的封装，即对代理对象的进一步预处理，而你传this的话它就找不到对应的代理对象）

## 需要权限控制的Mapping（未做）

未登录不能访问的mapping

/bookinformation.html

/booking.html