# **签到管理**

1. 建立数据库Activiting Students



Activiting

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段 | 描述 | 类型和长度 | 主键 | 可空 | 默认值 |
| 1 | A\_R\_id | 活动进行id | int | 是 | 否 | 无（自增） |
| 2 | A\_S\_id | 学生学号 | varchar(12) | 是 | 否 | 无 |
| 3 | A\_status | 活动预约状态（1.预约成功 2.扫码签到 3.后台确认学生签到） | int | 否 | 是 | 无 |
| 4 | A\_grade | 分数（1:优秀 2:通过 3:未通过） | int | 否 | 是 | 无 |
| 5 | A\_devicecode | 设备识别码 | varchar(255) | 否 | 是 | 无 |
| 6 | A\_level | 级别 | int | 否 | 是 | 无 |
| 7 | A\_firsttime | 第一次时间 | datetime | 否 | 是 | 无 |
| 8 | A\_secdevicecode | 第二次设备识别码 | varchar(255) | 否 | 是 | 无 |
| 9 | A\_secondtime | 第二次时间 | datetime | 否 | 是 | 无 |
| 10 | A\_thirddevicecode | 第三次设备识别码 | varchar(255) | 否 | 是 | 无 |
| 11 | A\_thirdtime | 第三次时间 | datetime | 否 | 是 | 无 |
| 12 | A\_fourdevicecode | 第四次设备识别码 | varchar(255) | 否 | 是 | 无 |
| 13 | A\_fourtime | 第四次时间 | datetime | 否 | 是 | 无 |

学生表 Students

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段 | 描述 | 类型和长度 | 主键 | 可空 | 默认值 |
| 1 | s\_id | 学生id | int | 是 | 否 | 无（自增） |
| 2 | s\_studentId | 学生学号 | varchar(12) | 是 | 否 | 无 |
| 3 | s\_name | 学生姓名 | varchar(255) | 否 | 是 | 无 |
| 4 | s\_sex | 性别（0位男生，1位女生） | tinyint | 否 | 是 | 无 |
| 5 | s\_school | 学校 | varchar(255) | 否 | 是 | 无 |
| 6 | s\_college | 所在学院 | varchar(255) | 否 | 是 | 无 |
| 7 | s\_profession | 所在专业 | varchar(255) | 否 | 是 | 无 |
| 8 | s\_class | 所在班级 | varchar(255) | 否 | 是 | 无 |
| 9 | s\_classtype | 所在班级类型 | varchar(255) | 否 | 是 | 无 |
| 10 | s\_jobstatus | 班级内担任职务 | varchar(255) | 否 | 是 | 无 |
| 11 | s\_tutor | 学生导师 | varchar(255) | 否 | 是 | 无 |
| 12 | s\_phone | 学生手机号 | varchar(15) | 否 | 是 | 无 |
| 13 | s\_idcard | 学生身份证号 | varchar(18) | 否 | 是 | 无 |
| 14 | s\_password | 密码 | varchar(255) | 否 | 是 | 无 |
| 15 | s\_createTime | 创建时间 | datetime | 否 | 是 | 无 |
| 16 | s\_changeTime | 最近一次修改信息时间 | datetime | 否 | 是 | 无 |
| 17 | s\_picture | 学生照片 | varchar(255) | 否 | 是 | 无 |
| 18 | s\_token | token号 | varchar(255) | 否 | 是 | 无 |
| 19 | s\_tokenendtime | token保存到期时间 | datetime | 否 | 是 | 无 |

1. 在web工程里反向工程





1. 然后在applicationContext.xml添加映射

<value>org/model/Activiting.hbm.xml</value>

<value>org/model/Students.hbm.xml</value>

1. 在informationlist.jsp页面添加签到管理按钮

<button onClick=*"qiandao(*${res.RId}*)"* type=*"button"* class=*"btn btn-primary btn-xs"*> <span>签到管理</span>

</button>



1. 给button添加点击是事件 onclick(qiandao($res.Rid))

**function** qiandao(id) {

**var** url = "stuQiandao.jsp?actid=" + id;

window.location.href = url;

}

1. 由于要知道点击的活动编号，所以通过res.rid获取到了相应的活动信息
2. 在webAction.java申明一个int变量actid用来保存临时变量，并生成get set方法

**private** **int** actid;//活动的编号

1. 添加stuQiandao.jsp页面

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1. ）这个页面用到了灵活的jequery.datatable插件这个插件是用来画表格的 2. 用到这个插件需要用到4个js文件 2个css文件，分别是：   Js：①jquery，②bootstrap3，③datatables的js，④datatables对应bootstrap样式  Css:①bootstrap3，②datatables对应bootstrap样式  我把这些样式都放在了 jules的文件夹里了  <link href=*"jules/css/bootstrap.min14ed.css?v=3.3.6"* rel=*"stylesheet"*>  <link href=*"jules/css/dataTables.bootstrap.css"* rel=*"stylesheet"*>  <script src=*"jules/js/jquery.min.js?v=2.1.4"*></script>  <script src=*"jules/js/bootstrap.min.js?v=3.3.6"*></script>  <script src=*"jules/js/jquery.dataTables.js"*></script>  <script src=*"jules/js/dataTables.bootstrap.js"*></script>  这些是使用datable这个插件需要引用到的文件   1. ）写一个方法用来匹配需要查找的类容   **var** pattern = **new** RegExp("[?&]" + name + "\=([^&]+)", "g");//用了正则表达式   |  |  | | --- | --- | | [i](http://www.w3school.com.cn/jsref/jsref_regexp_i.asp) | 执行对大小写不敏感的匹配。 | | [g](http://www.w3school.com.cn/jsref/jsref_regexp_g.asp) | 执行全局匹配（查找所有匹配而非在找到第一个匹配后停止）。 | | m | 执行多行匹配。 |     **var** matcher = pattern.exec(url);//用来匹配对应的 **RegExp 对象方法**  |  |  |  |  | | --- | --- | --- | --- | | [compile](http://www.w3school.com.cn/jsref/jsref_regexp_compile.asp) | 编译正则表达式。 | 1 | 4 | | [exec](http://www.w3school.com.cn/jsref/jsref_exec_regexp.asp) | 检索字符串中指定的值。返回找到的值，并确定其位置。 | 1 | 4 | | [test](http://www.w3school.com.cn/jsref/jsref_test_regexp.asp) | 检索字符串中指定的值。返回 true 或 false。 | 1 | 4 |   decodeURIComponent() 对编码后的 URI 进行解码  组合在一起：  **function** getUrlParam(url, name) {  **var** pattern = **new** RegExp("[?&]" + name + "\=([^&]+)", "g");  **var** matcher = pattern.exec(url);  **var** items = **null**;  **if** (matcher != **null**) {  **try** {  items = decodeURIComponent(decodeURIComponent(matcher[1]));  } **catch** (e) {  **try** {  items = decodeURIComponent(matcher[1]);  } **catch** (e) {  items = matcher[1];  }  }  }  **return** items;  }   1. ）调用getUrlParam方法   **var** url = window.location;  **var** deleteid = getUrlParam(url, 'actid');   1. ）绘制表格   这个#datatable1是这个table的id    **var** left1 = $('#datatable1').DataTable({  destory : **true**,//重新布置  searching : **true**,//启动搜索功能  bLengthChange : **true**,//是否允许用户，在下拉列表自定义选择分页大小(10, 25, 50 and 100),默认就是true  ordering : **false**,//不启用排序  bScrollInfinite : **true**,//开启内置滚动条，并且显示所有数据  bScrollCollapse : **true**,//当设置sScrolly时  如果数据没那么高  表格是否自适应高度  ajax : "toActqiandao.action?actid=" + deleteid,  aoColumns : [  {  "data" : "id"  },  {  "data" : "name"  },  {  "data" : "college"  },  {  "data" : "clasz"  },  {  "data" : "iphone"  }  ],  }  );  这里的id name college clasz iphone是从ajax里获取到的数据：   1. 在steuts.xml写入toActqiandao.action   <!-- 查看活动签到信息-->  <action name=*"toActqiandao"* class=*"webAction"* method=*"toActqiandao"*>  <result type=*"json"* name=*"success"*>  <param name=*"root"*>s</param>  </result>  </action>   1. 在webAction.java写入toActqiandao方法 2. 申明变量 ：   **private** List<Map<String, Object>> activitingslist = **new** ArrayList<Map<String, Object>>();  ActRecordDao actRecordDao;并生成get set方法   1. 在applicationContext.xml配置ActRecordDao   <bean id=*"actRecordDaoImp"* class=*"web.dao.imp.ActRecordDaoImp"*>  <propertyname=*"sessionFactory"* ref=*"mysessionFactory"*></property></bean>  <property name=*"teacherDao"* ref=*"teacherDaoImp"*></property>  <property name=*"relationDao"* ref=*"relationDaoImp"*></property>  <property name=*"addressDao"* ref=*"addressDaoImp"*></property>  <property name=*"activityDao"* ref=*"activityDaoImp"*></property>  <property name=*"actRecordDao"* ref=*"actRecordDaoImp"*></property>  </bean>   1. 新建接口ActRecordDao   **public** **interface** ActRecordDao {  List<Map<String, Object>> findActqiandao(**int** actid);  }   1. 实现接口ActRecordDaoImp   **public** **class** ActRecordDaoImp **extends** HibernateDaoSupport **implements** ActRecordDao {  @SuppressWarnings({ "unchecked", "unused" })  **public** List<Map<String, Object>> findActqiandao(**int** id) {  Session ssn = getSession();  Query query = ssn.createQuery(  "from Activiting a where A\_devicecode=null and A\_secdevicecode=null and A\_thirddevicecode=null"  + " and A\_fourdevicecode=null and A\_status=1 and a.id.ARId=" + id);  List<Activiting> l = query.list();  ssn.close();  List<Map<String, Object>> listg = **new** ArrayList<>();  **if** (l != **null** && l.size() > 0) {  **int** len = l.size();  Map<String, Object> map;  List<Students> list = **null**;  Object[] values = **new** Object[len];  StringBuilder sb = **new** StringBuilder();  **for** (**int** j = 0; j < len; j++) {  values[j] = l.get(j).getId().getASId();  **if** (j != len - 1) {  sb.append("?,");  } **else** {  sb.append("?");  }}  list = getHibernateTemplate().find("FROM Students t WHERE t.id.SStudentId IN (" + sb.toString() + ")",  values);  **if** (list != **null** && list.size() > 0) {  **for** (Activiting a : l) {  **for** (Students stu : list) {  **if** (stu.getId().getSStudentId().equals(a.getId().getASId())) {  map = **new** HashMap<String, Object>();  map.put("id", stu.getId().getSStudentId());  map.put("name", stu.getSName());  map.put("college", stu.getSCollege());  map.put("clasz", stu.getSClass());  map.put("iphone", stu.getSPhone());  listg.add(map);  }}}}}  **return** listg;}}  **public** String toActqiandao() {  activitingslist = actRecordDao.findActqiandao(actid);  s = JSONTools.*createJsonObject*("data", activitingslist);  **return** "success";  } |

效果：



