

6-3 查询会议列表分页数据（后端）

140.143.132.225:8000/project-1/doc-101

一、业务分析

每个员工只能看到自己参与的会议记录，自己不参与的会议是看不到的。而且会议必须是未开始状态，或者进行中状态才能被看到，已经结束的会议、审批中的会议、审批不通过的会议都是无法被看到的。



输入会议标题

日期

2021-02-07

开始时间

23:36

结束时间

00:36

会议类型

在线会议

会议内容

参会者 (0人)



保存

二、编写持久层代码

在 `TbMeetingDao.xml` 文件中编写分页查询的SQL语句

```

1. <select id="searchMyMeetingListByPage" parameterType="HashMap" resultType="HashMap">
2.     SELECT
3.         m.id,
4.         m.uuid,
5.         m.title,
6.         u2.name,
7.         DATE_FORMAT(m.date,'%Y年%m月%d日') AS date,
8.         m.place,
9.         DATE_FORMAT(m.start,'%H:%i') AS start,
10.        DATE_FORMAT(m.end,'%H:%i') AS end,
11.        m.type,
12.        m.status,
13.        m.desc,
14.        u2.photo,
15.        TIMESTAMPDIFF(HOUR,CONCAT(m.date," ",m.start),CONCAT(m.date," ",m.end)) AS hour
16.    FROM tb_meeting m
17.    JOIN tb_user u1 ON JSON_CONTAINS ( m.members, CAST( u1.id AS CHAR ) )
18.    JOIN tb_user u2 ON m.creator_id = u2.id
19.    WHERE u1.id = #{userId} AND m.status IN(3,4)
20.        AND u1.status = 1 AND u2.status = 1
21.    ORDER BY m.date, m.start, m.id
22.    LIMIT #{start},#{length}
23. </select>

```

在 `TbMeetingDao.java` 接口中定义Dao方法

```

1. @Mapper
2. public interface TbMeetingDao {
3.     .....
4.     public ArrayList<HashMap> searchMyMeetingListByPage(HashMap param);
5. }

```

三、编辑业务层代码

在 `MeetingService.java` 中添加抽象分页方法

```

1. public interface MeetingService {
2.     .....
3.     public ArrayList<HashMap> searchMyMeetingListByPage(HashMap param);
4. }

```

在 `MeetingServiceImpl.java` 中添加分页方法

```

1. public class MeetingServiceImpl implements MeetingService {
2.     .....
3.     @Override
4.     public ArrayList<HashMap> searchMyMeetingListByPage(HashMap param) {
5.         ArrayList<HashMap> list = meetingDao.searchMyMeetingListByPage(param);
6.         String date = null;
7.         ArrayList resultList = new ArrayList();
8.         HashMap resultMap = null;
9.         JSONArray array = null;
10.        for (HashMap map : list) {
11.            String temp = map.get("date").toString();
12.            if (!temp.equals(date)) {
13.                date = temp;
14.                resultMap = new HashMap();
15.                resultList.add(resultMap);
16.                resultMap.put("date", date);
17.                array = new JSONArray();
18.                resultMap.put("list", array);
19.            }
20.            array.put(map);
21.        }
22.        return resultList;
23.    }
24. }

```

四、编写Web层代码

创建 `SearchMyMeetingListByPageForm.java` 封装分页请求数据

```

1. @ApiModel
2. @Data
3. public class SearchMyMeetingListByPageForm {
4.     @NotNull
5.     @Min(1)
6.     private Integer page;
7.     @NotNull
8.     @Range(min = 1,max = 40)
9.     private Integer length;
10. }

```

创建 `MeetingController.java` 类，添加分页查询Web方法

```
1. @RestController
2. @RequestMapping("/meeting")
3. @Api("会议模块网络接口")
4. public class MeetingController {
5.     @Autowired
6.     private JwtUtil jwtUtil;
7.     @Autowired
8.     private MeetingService meetingService;
9.     @PostMapping("/searchMyMeetingListByPage")
10.    @ApiOperation("查询会议列表分页数据")
11.    public R searchMyMeetingListByPage(@Valid @RequestBody SearchMyMeetingListByPageForm
    form, @RequestHeader("token") String token) {
12.        int userId = jwtUtil.getUserId(token);
13.        int page = form.getPage();
14.        int length = form.getLength();
15.        long start = (page - 1) * length;
16.        HashMap map = new HashMap();
17.        map.put("userId", userId);
18.        map.put("start", start);
19.        map.put("length", length);
20.        ArrayList list = meetingService.searchMyMeetingListByPage(map);
21.        return R.ok().put("result", list);
22.    }
23. }
```

