
微简开发指引

微简公司
2022年8月

目录

| | |
|-----------------------------|----|
| 1 总述 | 4 |
| 2 开发流程 | 4 |
| 3 vjpage 画界面工具 | 5 |
| 3.1 批量生成界面 | 5 |
| 3.2 常用功能 | 5 |
| 3.2.1 新建项目 | 5 |
| 3.2.2 新建、插入文件夹 | 6 |
| 3.2.3 新建、插入页面 | 6 |
| 3.2.4 设置组件属性 | 7 |
| 3.2.5 打开页面 | 8 |
| 3.2.6 打开窗口 | 9 |
| 3.2.7 页面布局 | 11 |
| 3.2.8 样式属性 | 12 |
| 3.2.9 边距微调 | 13 |
| 3.2.10 设计器中页面刷新 | 14 |
| 3.2.11 生成页面 | 14 |
| 3.3 vjpage 详细功能请参考帮助 | 16 |
| 4 详细沟通需求 | 16 |
| 4.1 大场景走通 | 16 |
| 4.2 页面详细内容 | 17 |
| 4.3 其他需求 | 17 |
| 5 字典 | 17 |
| 5.1 字典定义 | 17 |
| 5.2 Vjproject 查看对应的字典 | 19 |
| 5.3 字典入库、更新 | 21 |
| 6 数据库设计 | 23 |
| 6.1 添加数据库 | 23 |
| 6.2 添加表 | 24 |
| 6.3 从父项目导入 | 25 |
| 6.4 设计表结构 | 25 |
| 7 配置数据库 | 27 |
| 7.1 批量配置数据表 | 27 |
| 7.2 手工配置 | 28 |
| 7.3 检查 | 28 |
| 8 接口设计 | 29 |
| 8.1 接口定义的几个来源 | 29 |
| 8.2 添加文件夹、子文件夹 | 30 |
| 8.3 添加接口 | 30 |
| 8.4 复制接口定义 | 31 |
| 8.5 接口调试 | 32 |
| 9 前端页面命名 | 35 |

| | |
|----------------------------|----|
| 9.1 附加样式命名 | 35 |
| 9.2 字段命名 | 36 |
| 9.3 其他注意的: | 37 |
| 9.4 检查 | 37 |
| 9.5 页面命名 | 39 |
| 9.6 页面命名规则 | 39 |
| 10 生成 java 代码 | 39 |
| 10.1 生成代码 | 39 |
| 10.2 使用方法 | 41 |
| 11 生成前端代码 | 42 |
| 11.1 前置条件 | 43 |
| 11.2 设置配置文件 | 43 |
| 11.3 使用方法 | 45 |
| 12 生成后前后端代码修改 | 46 |
| 13 vjmanage 项目管理 | 47 |
| 13.1 登录入口 | 47 |
| 13.2 编辑项目信息 (配置接口地址) | 47 |
| 13.3 添加项目成员、设置权限 | 47 |
| 13.4 项目通讯录 | 48 |
| 13.5 vjpage 项目关联 | 48 |
| 13.6 指定父项目 | 50 |
| 13.7 文档备份 | 51 |
| 13.8 代码备份 | 51 |
| 13.9 Bug 管理 | 51 |
| 14 vjframe 框架介绍 | 52 |
| 15 vjframe 框架开发 | 52 |

1 总述

微筒全家桶包括三个工具、一套应用框架。

■ 三个工具

vjpage: 画界面工具，代码可用的原型工具，既有原型工具的简单易用，又能生成可用的前端代码。pc web 界面、app 界面都可以。

vjproject: 项目开发管理工具。具体功能包括：需求管理、数据库设计、接口管理、字典、错误码、笔记。生成 java 代码。

vjmanage: 项目管理工具。vjpage 项目关联、父项目管理、代码管理、文档管理、项目成员、项目计划、bug 管理。

■ 一个应用框架

vjframe 是微筒自研的 java 前后端分离框架，全部采用原生+函数为主的写法。**vjframe** 是一个动态 web 项目，采用 eclipse 开发。后端采用 servlet 基本的原生架构。前端采用 bootstrap、adminlte、echarts、jqgrid、ztree 等，并做了大量美化。

基本功能: 包含了登录、注册、用户管理、角色管理、组管理、权限管理，和一个自研的小型工作流引擎。

vjpage 生成的前端开发代码必须复制到 **vjframe** 中才能用。**vjproject** 生成的 java 代码必须复制到 **vjframe** 中才能用。

2 开发流程

使用全家桶开发流程和常规流程不太一样，和其他开发工具也不一样。微筒开发更强调界面，以界面为主，引导用户的详细需求。

■ 具体的开发步骤:

(一) **vjpage** 画界面。可以自己画，也可以向微筒申请可复用的业务界面（具体申请方法见微筒官网：使用方法），微筒有大量业务界面积累，也可以批量生成业务界面。（**vjpage**）

(二) 详细沟通需求。以界面为主和用户沟通需求，沟通需求过程中，把详细的界面画出来、需求文档写出来。（**vjpage**、其他常规工具）

(三) 需求过程中把字典捋出来。（**vjproject**）

(四) 设计数据库表，详细的字段，字段取值。（**vjproject**）

(五) 配置数据库——生成接口、生成 java 代码——Java 代码复制到 vjframe 中。

(vjproject)

(六) 设计更详细的接口。(vjproject)

(七) 前端界面配置，关联数据库，前端开发命名——生成前端开发代码——前端代码复制到 vjframe 中。

(八) 代码部分脱离微筒工具，到常规工具中手工开发。vjproject 持续使用。

(九) 如果前端或后端发生改变，需要重新生成，不能逆回去。

3 vjpage 画界面工具

3.1 批量生成界面

该功能为微筒内部功能，不对外开放。如果需要现成的业务界面，请向微筒申请，具体见微筒官网（使用方法）。微筒可提供成套的高质量业务界面。

3.2 常用功能

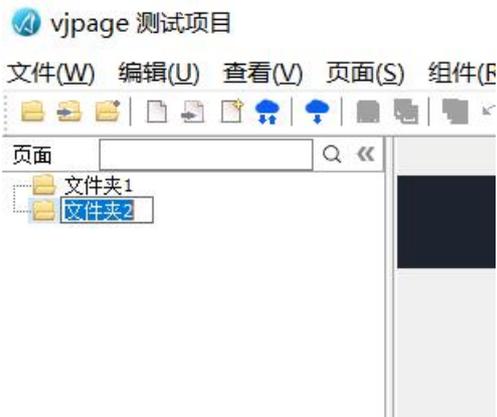
3.2.1 新建项目

选择 pc 前后端分离。新建 pc 项目。输入项目名称，单击确定。



3.2.2 新建、插入文件夹

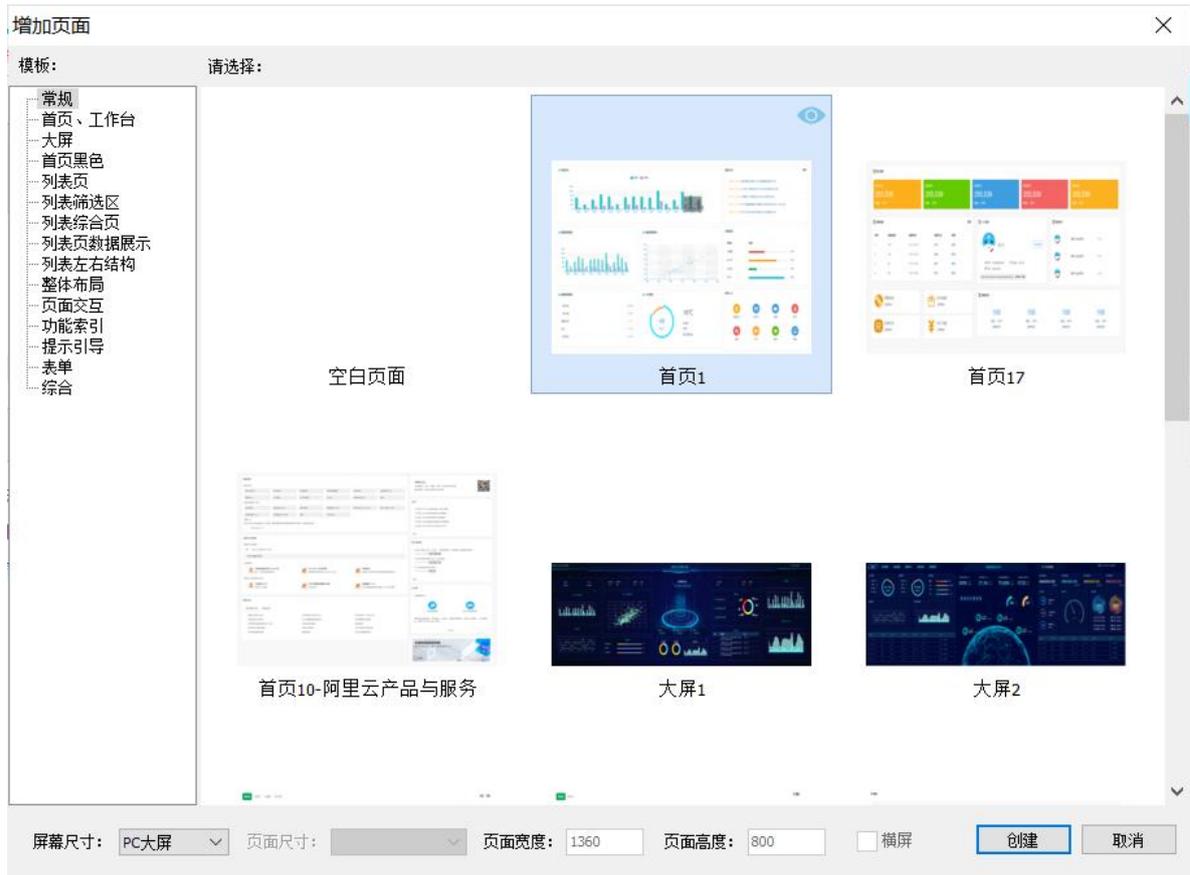
新建或插入文件夹，作为页面的分类。



3.2.3 新建、插入页面

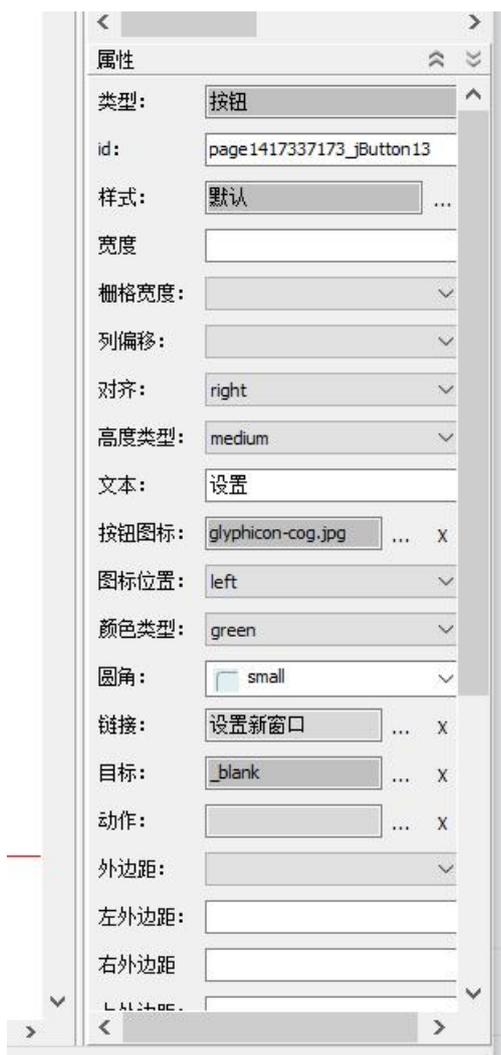
新建页面，选择一个模板。注意，尽量用模板，或者在几个模板页面的基础上做修改、拼凑。

如果用空页面，需要一定的积累才可以，代码可用的原型工具不同于 axure，完全固定定位，需要对 bootstrap 的布局有一定了解，对 vjpage 界面元素的属性有一定了解。如果技术模板中没有，可以直接向微简申请界面模板服务。



3.2.4 设置组件属性

选中组件之后，可修改相关属性。

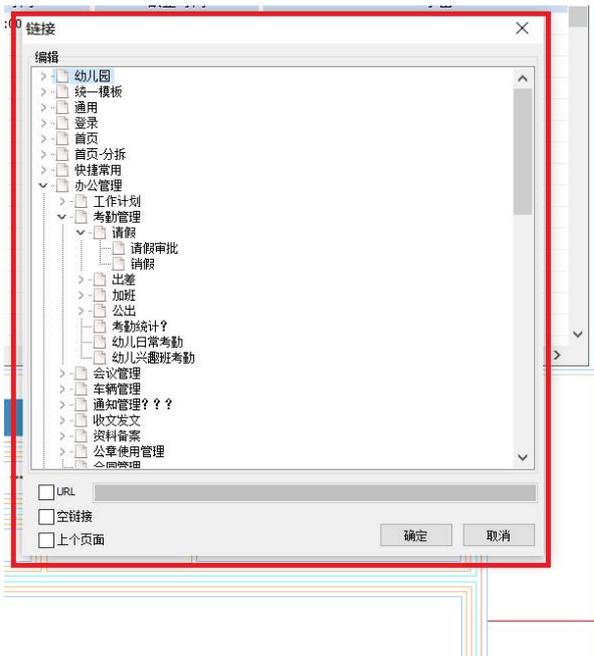


3.2.5 打开页面

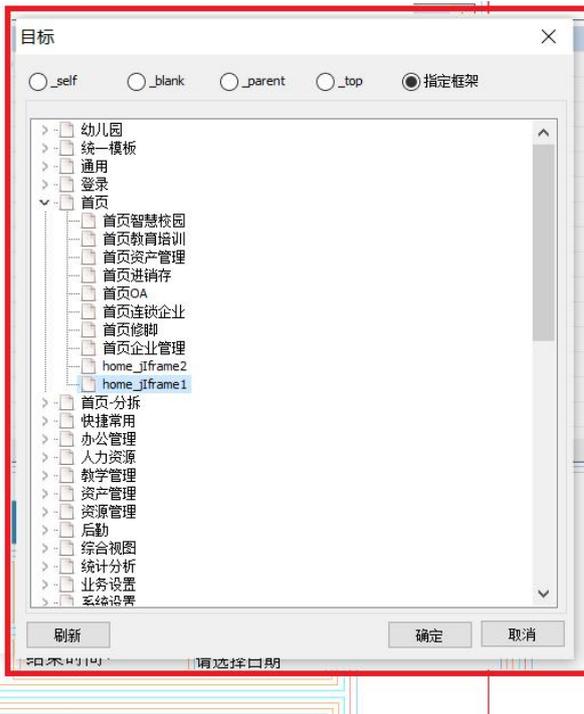
选中组件，比如按钮（单击按钮打开页面）。设置链接、目标属性即可。



单击链接属性，选中要链接打开的页面。

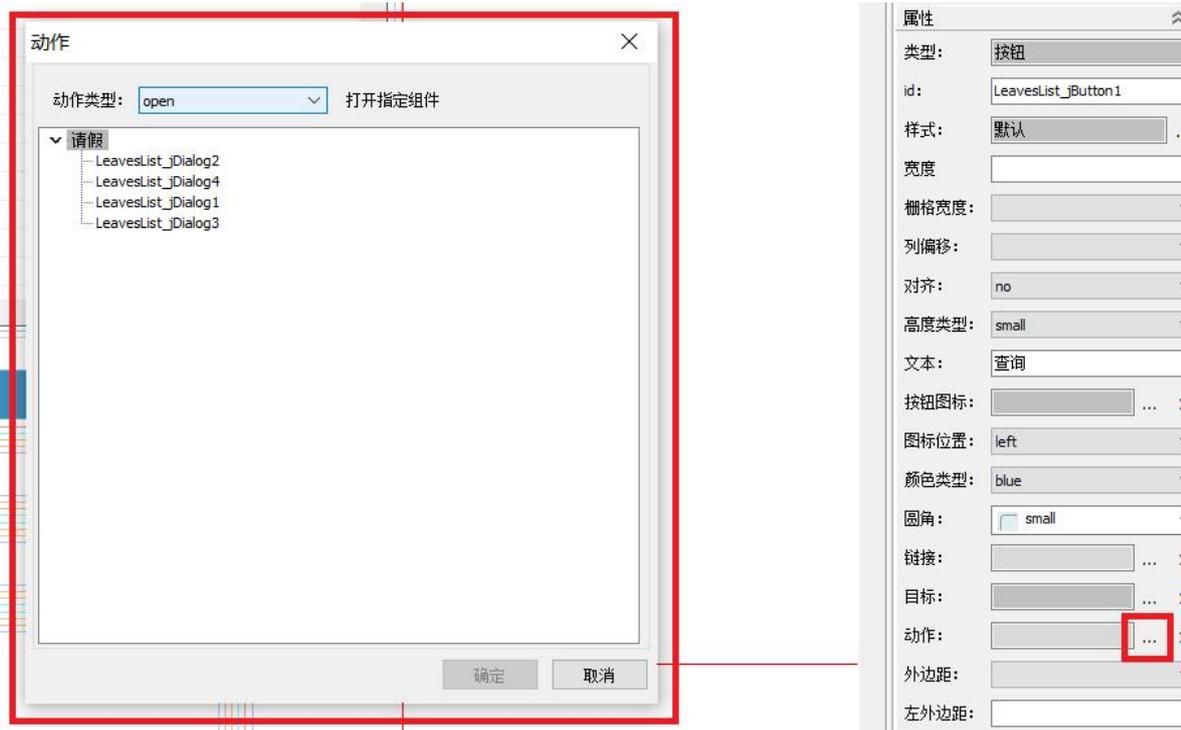


选择目标属性。_self 为本页面，_blank 为新页面，_parent 为父页面，指定框架可以设置在某个 iframe 中打开。

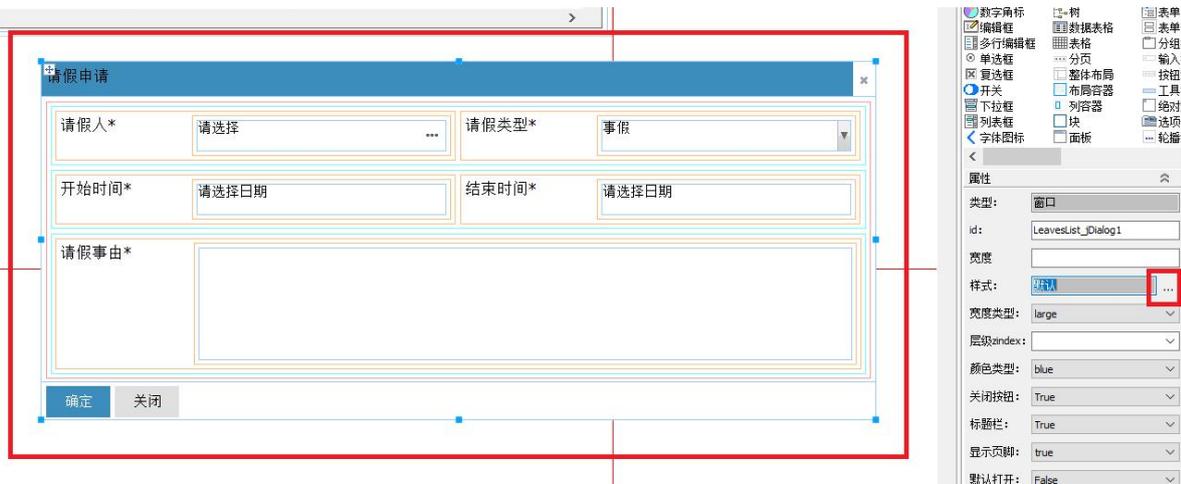


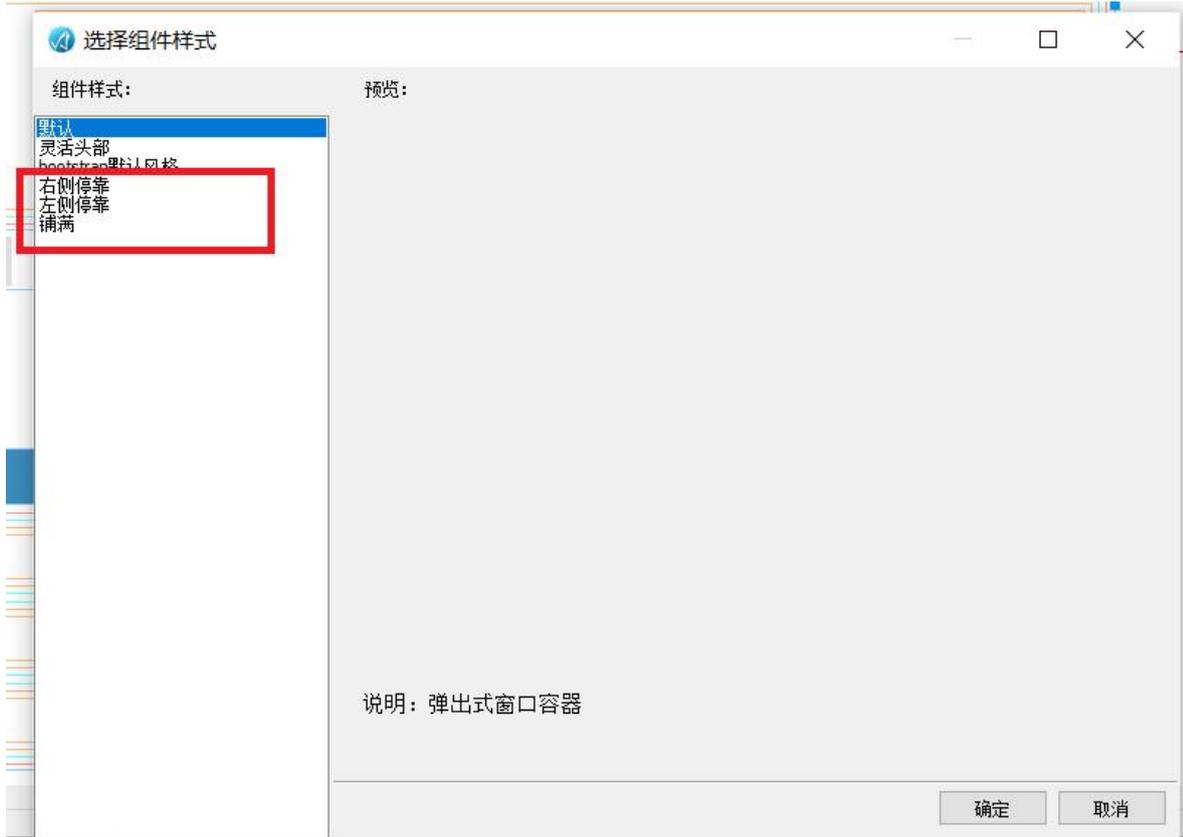
3.2.6 打开窗口

拖入一个窗口组件。以单击按钮打开 dialog 页面为例：选中按钮，单击动作属性，可以选择要打开的 dialog 窗口。



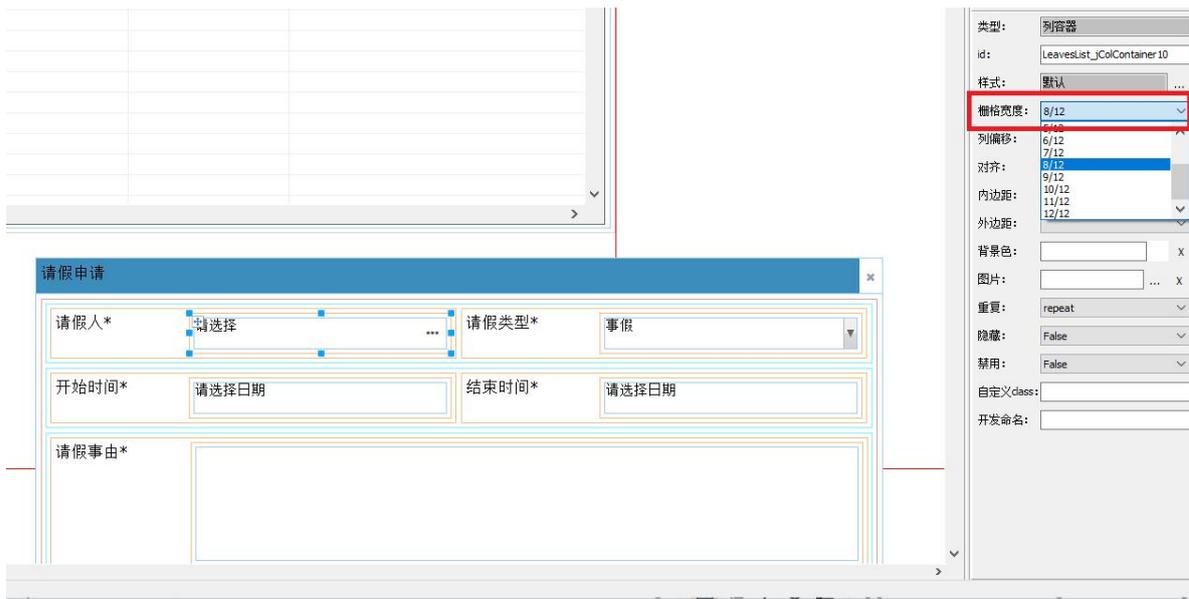
选中 dialog，单击样式属性。可设置 dialog 窗口为左侧停靠、右侧停靠、铺满等模式。





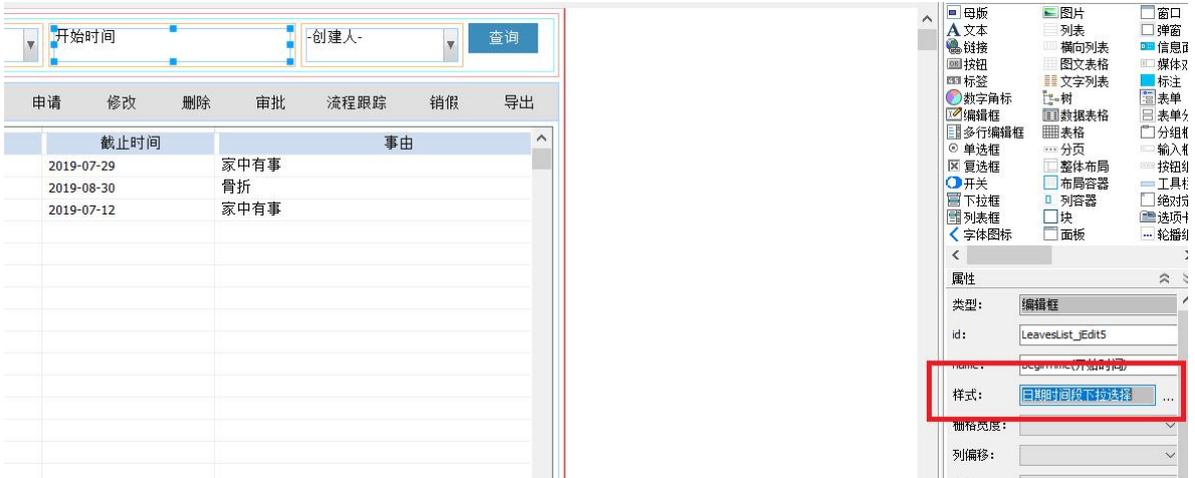
3.2.7 页面布局

Bootstrap 布局是栅格化流式布局（不懂的可百度学习一下）。宽度分为 12 等份，组件可以占到 $n/12$ 。可通过栅格宽度来设置。

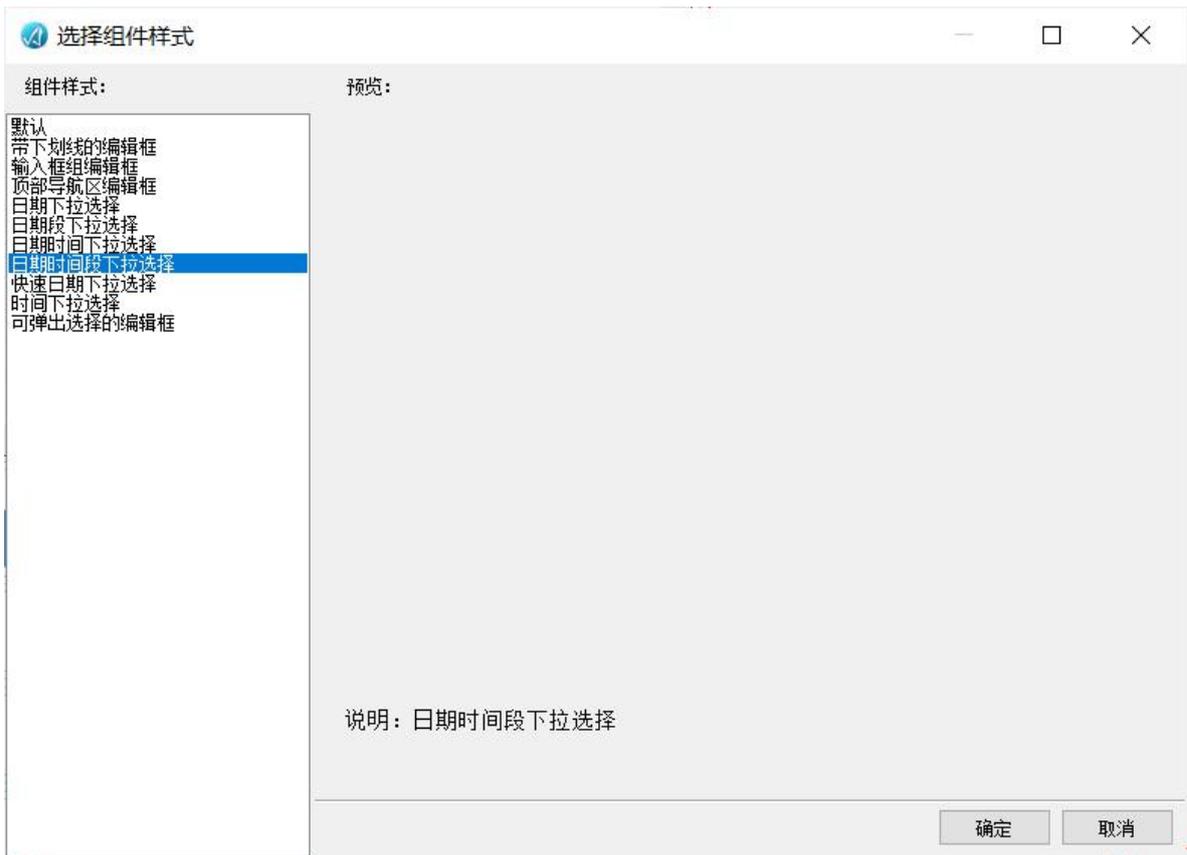


3.2.8 样式属性

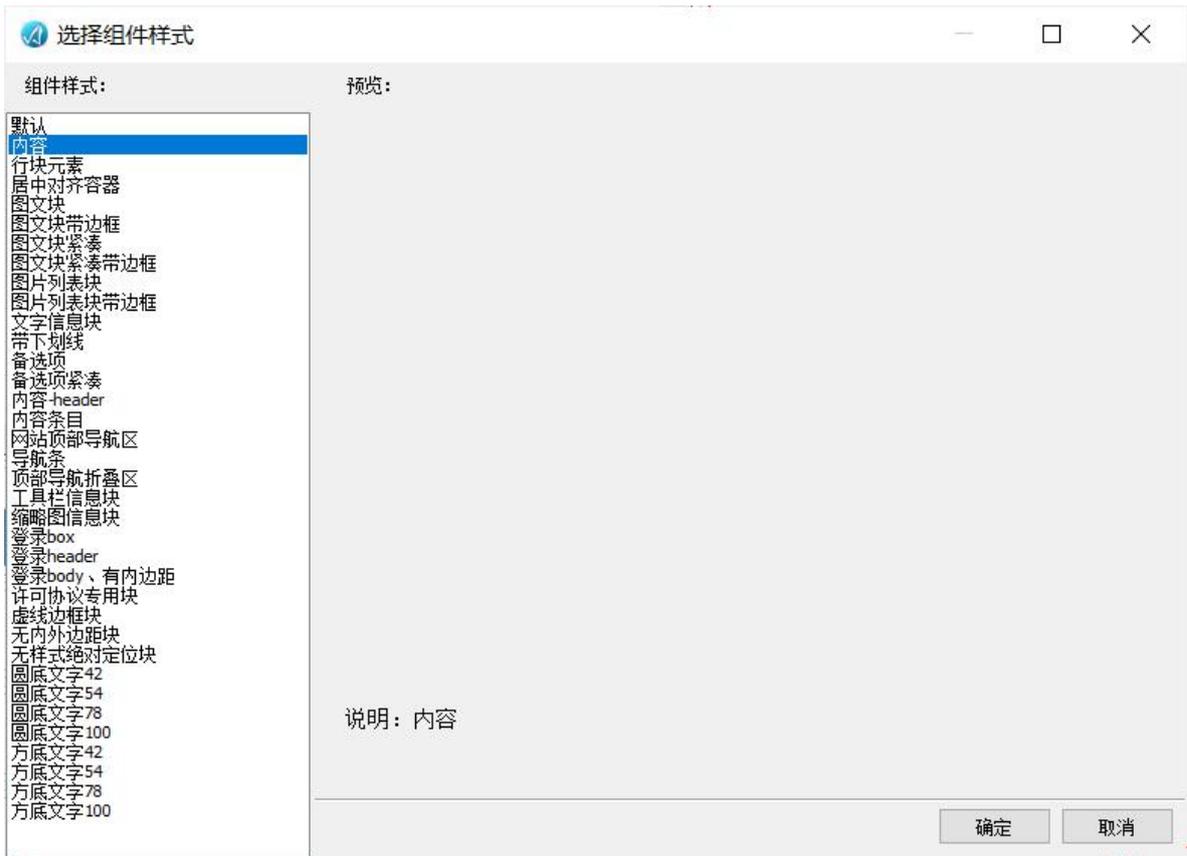
样式属性是 vjpage 的核心，对应了一组不同的 css 样式。



编辑框的样式:



块的样式:

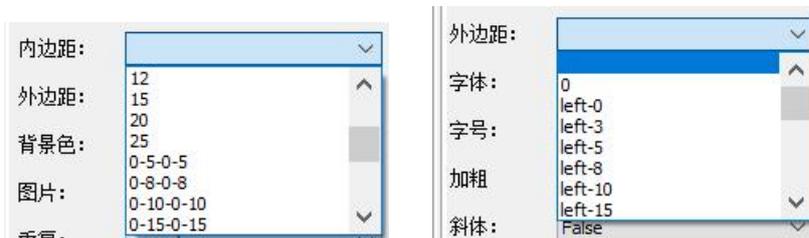


3.2.9 边距微调

页面设计时，对齐是比较麻烦的过程。我们提供了几个属性。内边距、外边距。将常用的边距样式写成了固定样式。可以直接设置。

也比可以自己指定边距。

考虑到代码的可用性，代码的冗余，尽量用现成的内边距、外边距属性。如果不够，再手工指定。



| | |
|-------|----------------------|
| 左外边距: | <input type="text"/> |
| 右外边距: | <input type="text"/> |
| 上外边距: | <input type="text"/> |
| 下外边距: | <input type="text"/> |
| 左内边距: | <input type="text"/> |
| 右内边距: | <input type="text"/> |
| 上内边距: | <input type="text"/> |
| 下内边距: | <input type="text"/> |

3.2.10 设计器中页面刷新

本工具是用 delphi 开发的，设计器也是用 delphi 组件模拟了页面的动态布局。相对浏览器而言，设计器的布局模拟要简单的多。为了设计时的流畅，vjpage 没有一次性将刷新彻底运行，可能需要三次刷新才能正常。每个拖拽动作、某些属性、打开时都会自动刷新一次。

3.2.11 生成页面

可以生成本页面、生成部分页面、生成全部页面。

生成之前需要配置生成页面的路径。

生成 html、js 文件。生成路径中最好是空目录。

可以选择样式风格。





3.3 vjpage 详细功能请参考帮助

vjpage 其他功能可以自己摸索，或者参考帮助，功能非常简单。

4 详细沟通需求

定制项目中，用户需求是一切的核心，而用户的需求通常是模糊的，不确定的。所以我们通过界面的细化、精确化尽量将需求快速、准确的确定下来。

vjpage 界面时最主要的沟通方式。此外，常规的流程图、word 文档也都是必要的。vjpage 不够用时，可以使用 axure、草图等常规工具。

必须制定非常详细的需求计划，保证用户能够按照需求计划进行配合。这是整个项目的关键。必须确定具体的配合人员。这个过程的松紧决定了项目的总体周期和成本。

4.1 大场景走通

场景列出来，用草图、流程图。

看每个场景是否能走通，功能是否够用。用户使用时是否清晰。

配置页面是否齐全。

页面补全。按钮补全。

页面交互的合理性，选择不同的页面模板。

4.2 页面详细内容

界面上的业务字段是否够用、是否有多余。

术语是否合理。

字典补全。

界面上的字段顺序。

校验、必填项。

筛选条件合理。

列表显示内容、顺序、宽度。

界面上的标签尽量和数据库字段名一致。

整个过程中可以和用户随时沟通。

最后和用户每个页面详细过一遍。

业务界面中必须做成相关的模拟业务数据。——数据表格里的数据必须是和业务一样的。用户看起来会非常有感觉。这些数据必须是和用户沟通后要的。

4.3 其他需求

调研每个模块的数据权限。

沟通需求：业务需求、功能需求、开发逻辑。数据库、字典、其他字典、弹出选择、主菜单、角色、权限、机构、 workflow、报表、倒数据等等。

和用户沟通原则：1、定时间。2、不能等用户。3、列出所有场景，基本覆盖大部分功能。4、按场景和用户挨着过，视频会议，或者面对面。5、用户方人要少而精。

有些细节可以下来后用户补充，比如字典，校验一类。

5 字典

5.1 字典定义

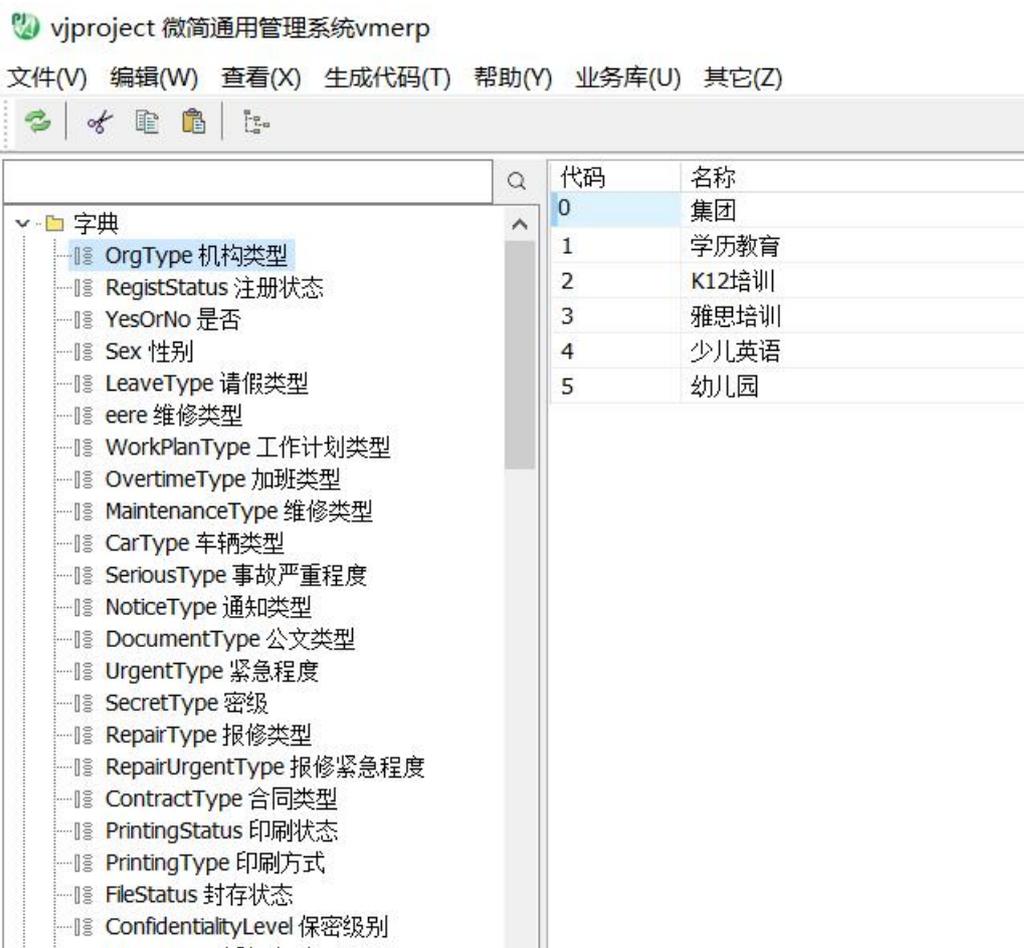
字典广义的可以认为是所有下拉框里的选项。

需求过程中，字典内容尽量都确定出来，并写到 vjproject 工具中。

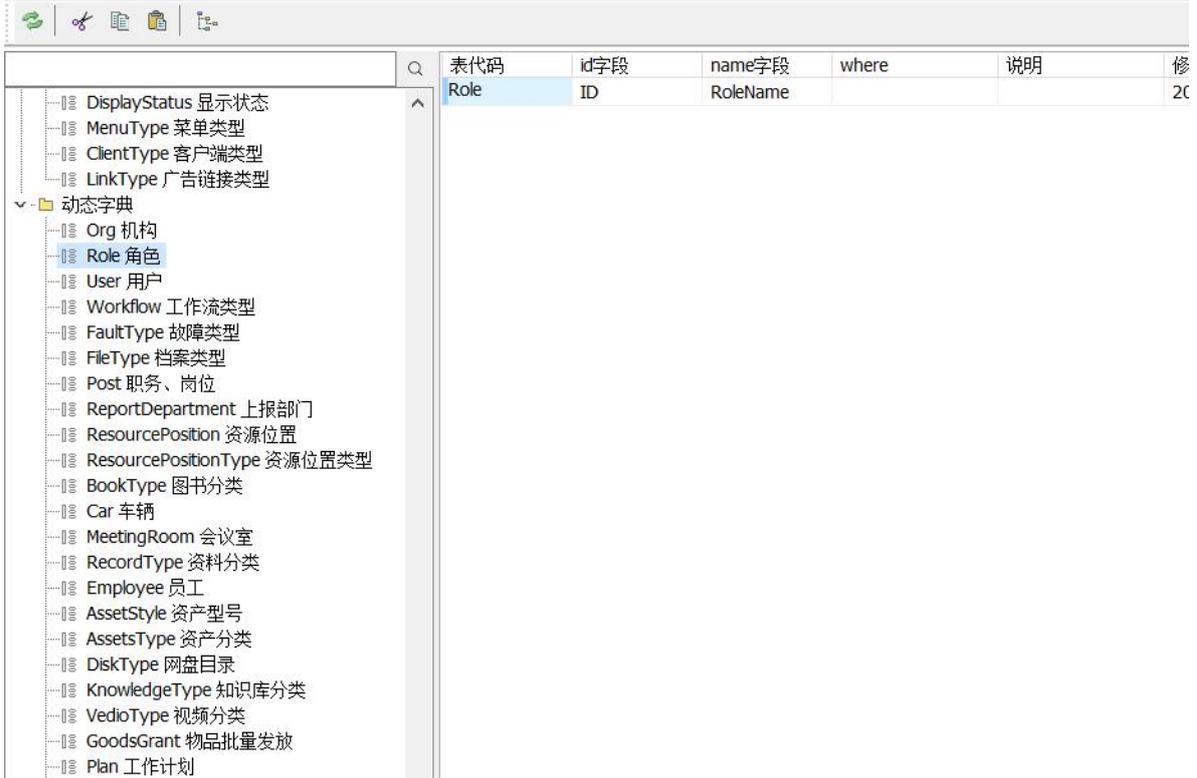
其他字典：动态字典，用户可配置的各种数据内容。

字典名尽量用字段名，没有字段时也可以在数据库设计好之后再修改。

字典：

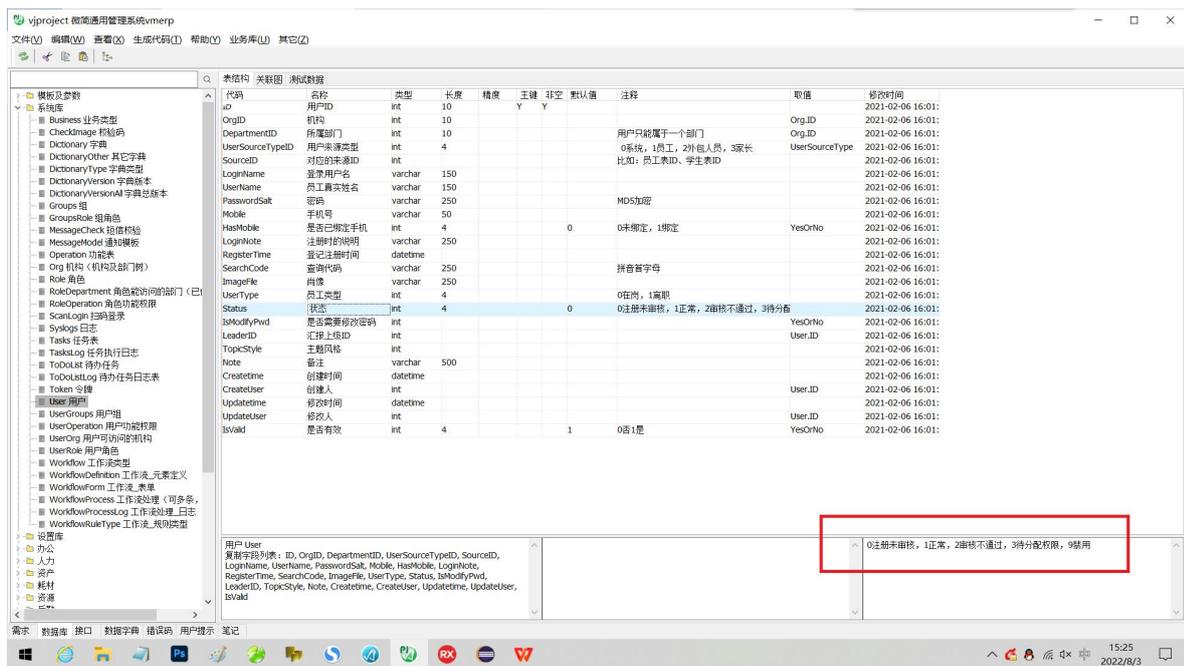


动态字典：

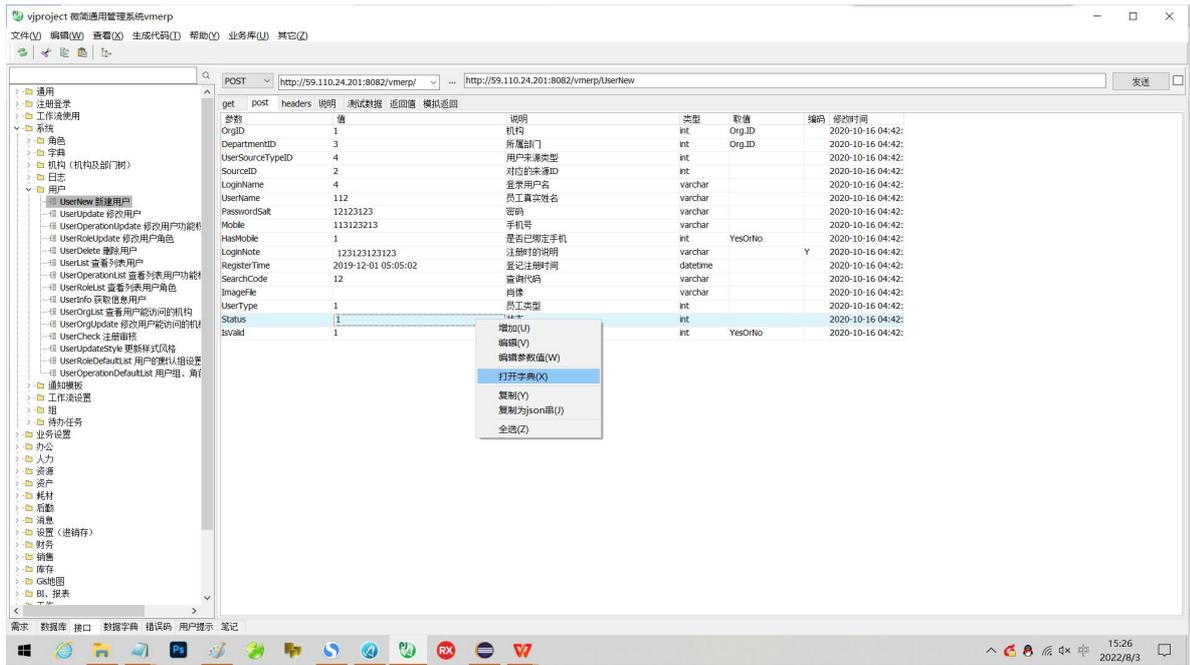


5.2 Vjproject 查看对应的字典

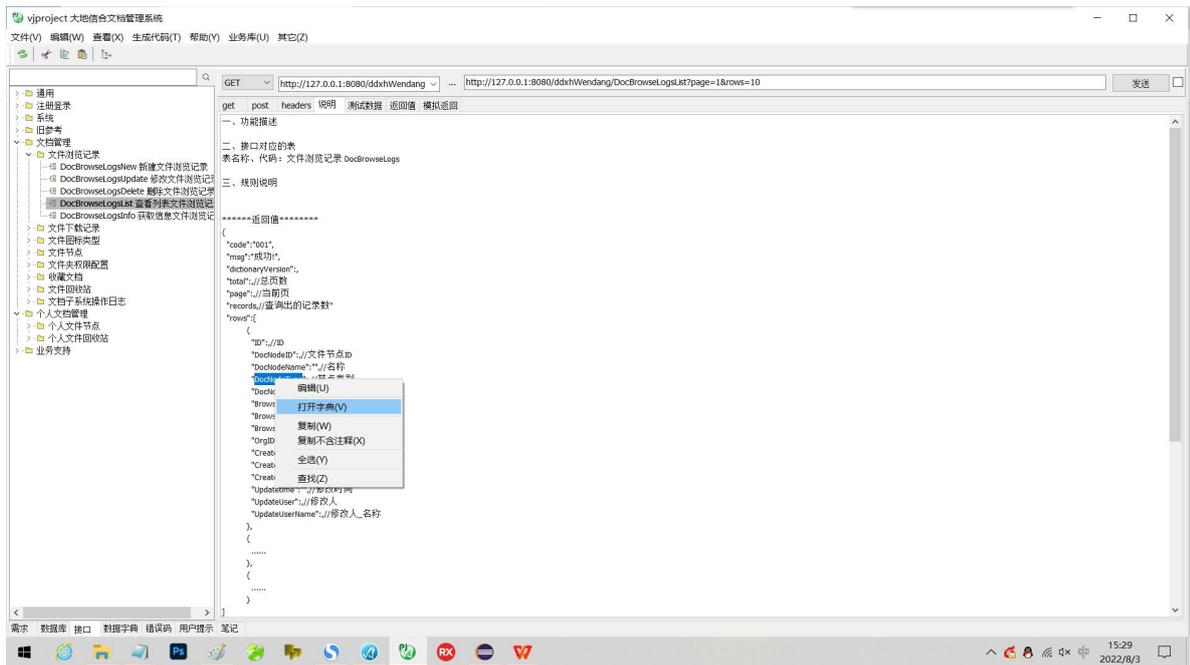
1、数据库视图中，选择字段，右下部显示对应的字典值。

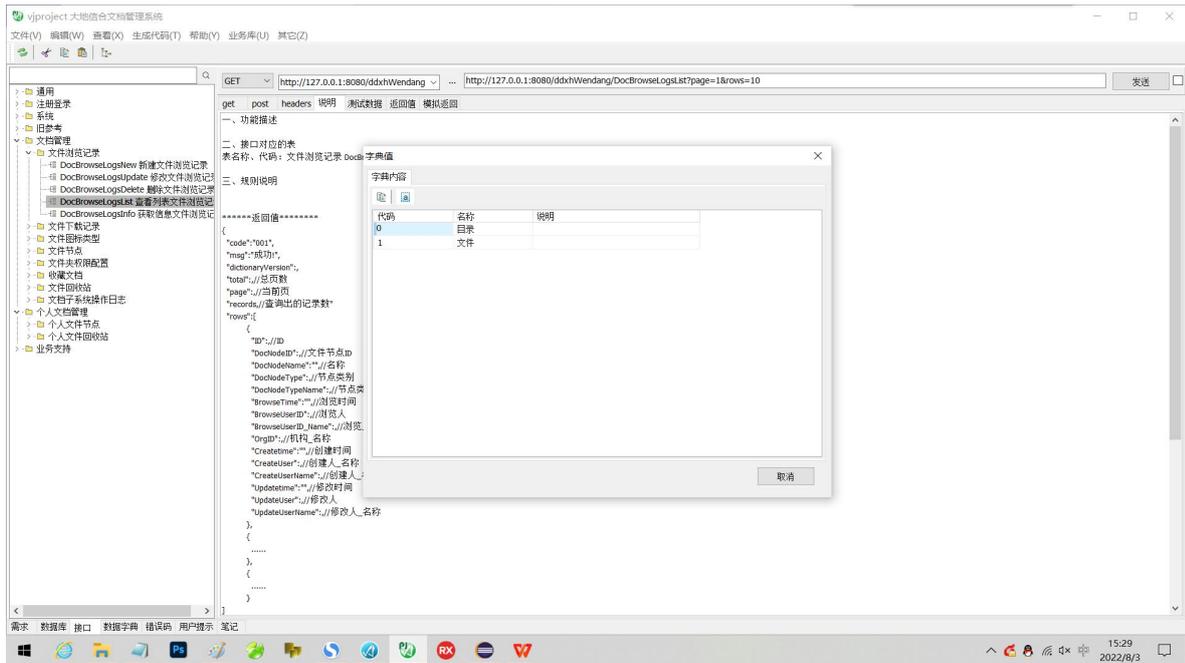


2、接口参数部分：右键，打开字典。



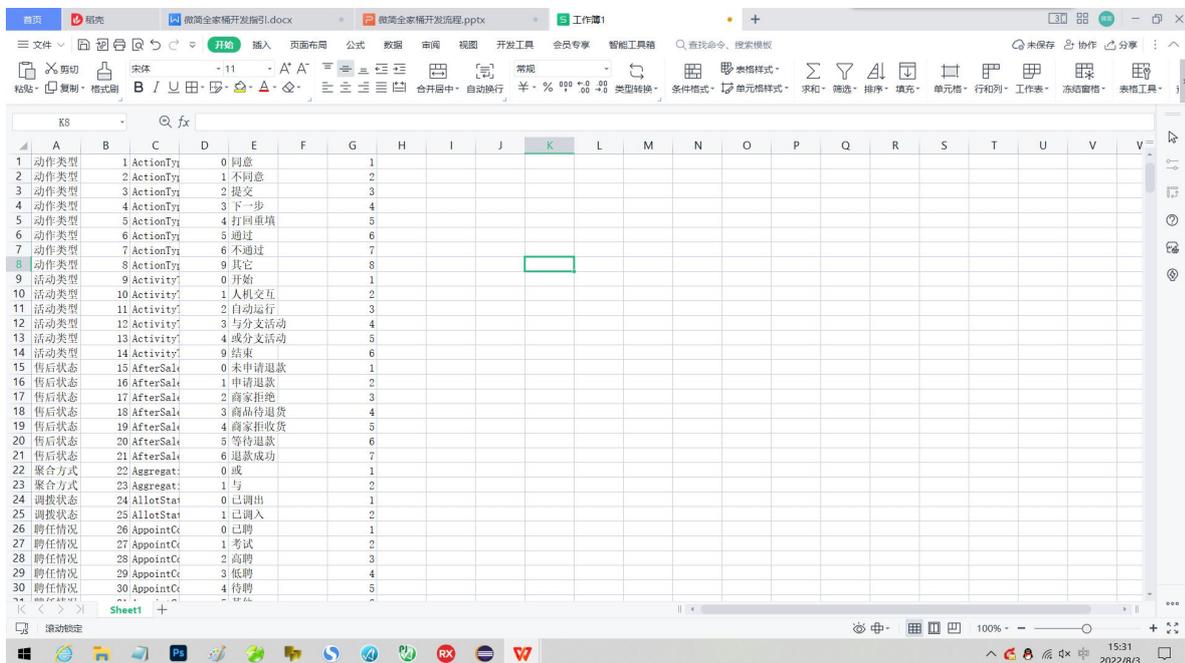
3、接口说明部分：选择文本，右键，打开字典。

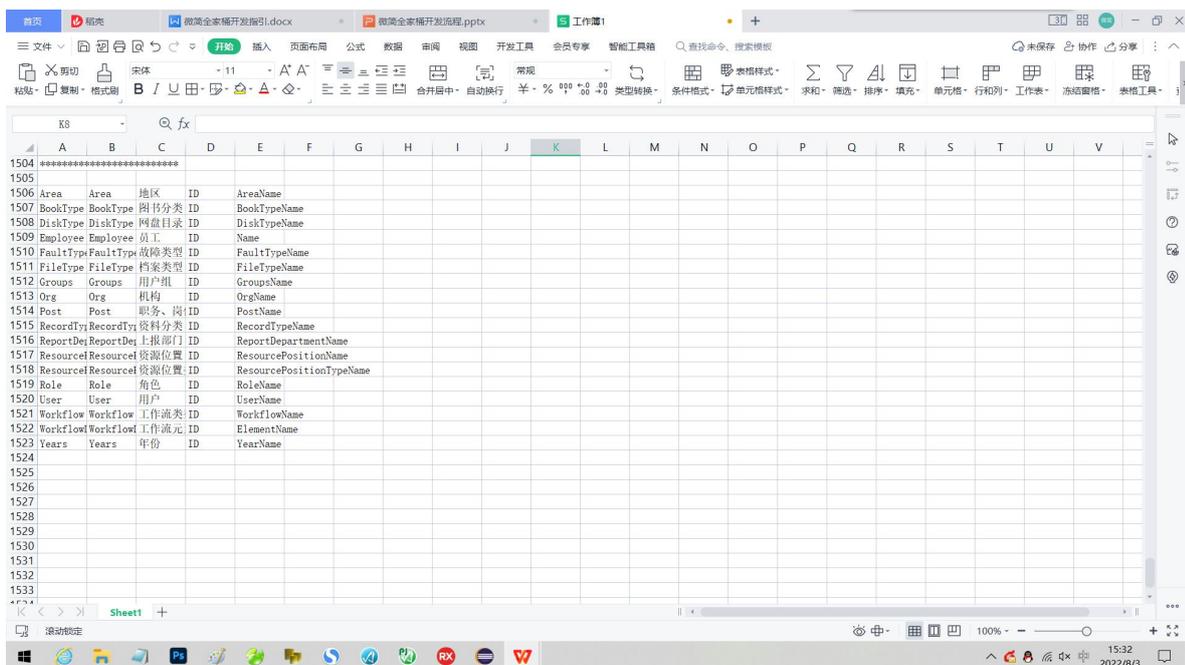
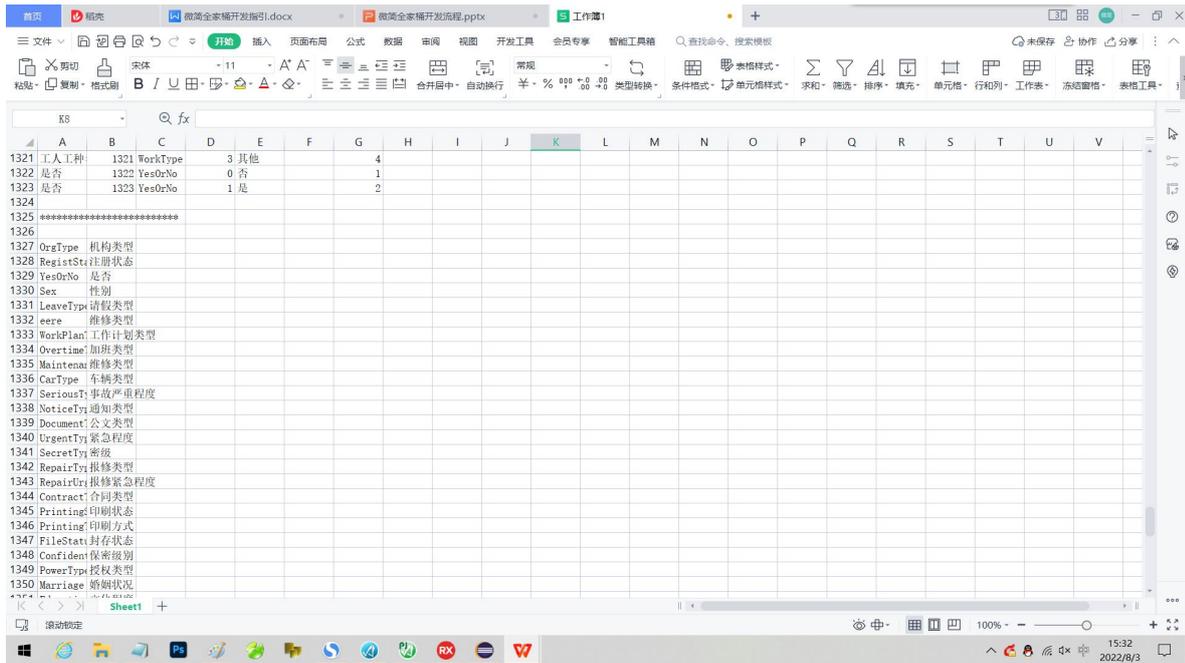




5.3 字典入库、更新

- 1、Vjproject 中，切换到数据字典页。
- 2、选中任意树节点，右键，复制字典值。
- 3、打开 excel，粘贴。





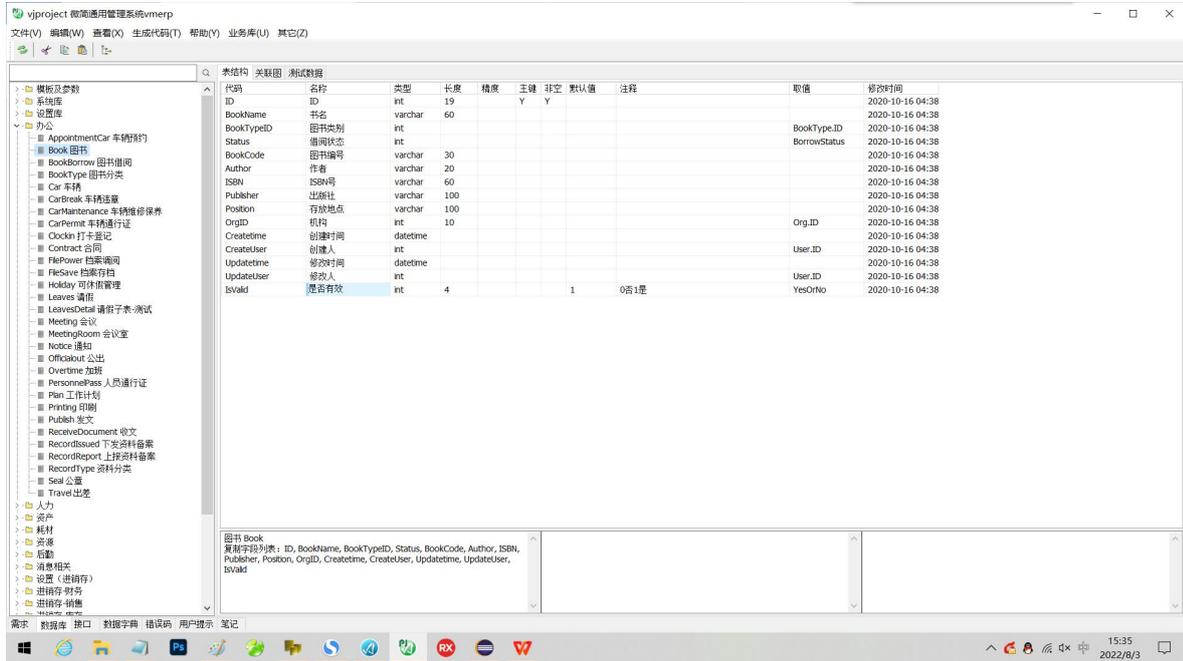
4、将对应部分复制到数据表里。（字典 Dictionary，其它字典 DictionaryOther，字典类型 DictionaryType）。

更新的时候，先删除库表里的再粘贴。自增 id 起始值设为 1。

养成良好的习惯：修改字典的时候，必须先改 vjproject 中对应部分。因为前端开发、后台开发、数据库表修改都要统一，都是以 vjproject 中的为准。

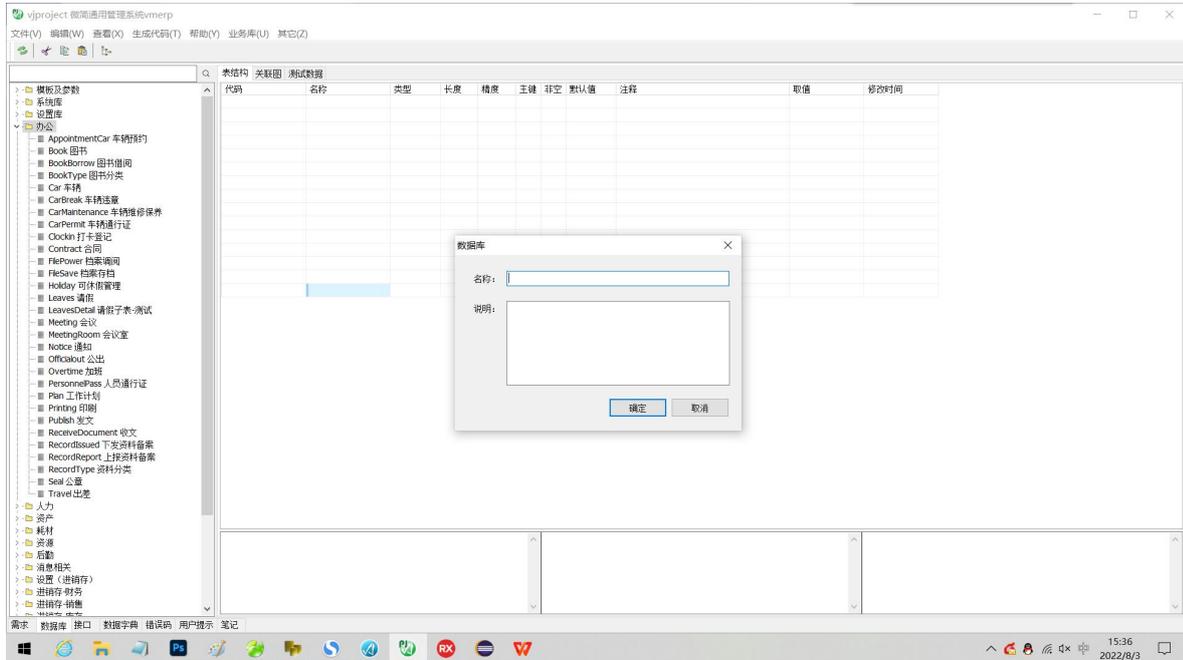
6 数据库设计

vjproject 中设计数据库。



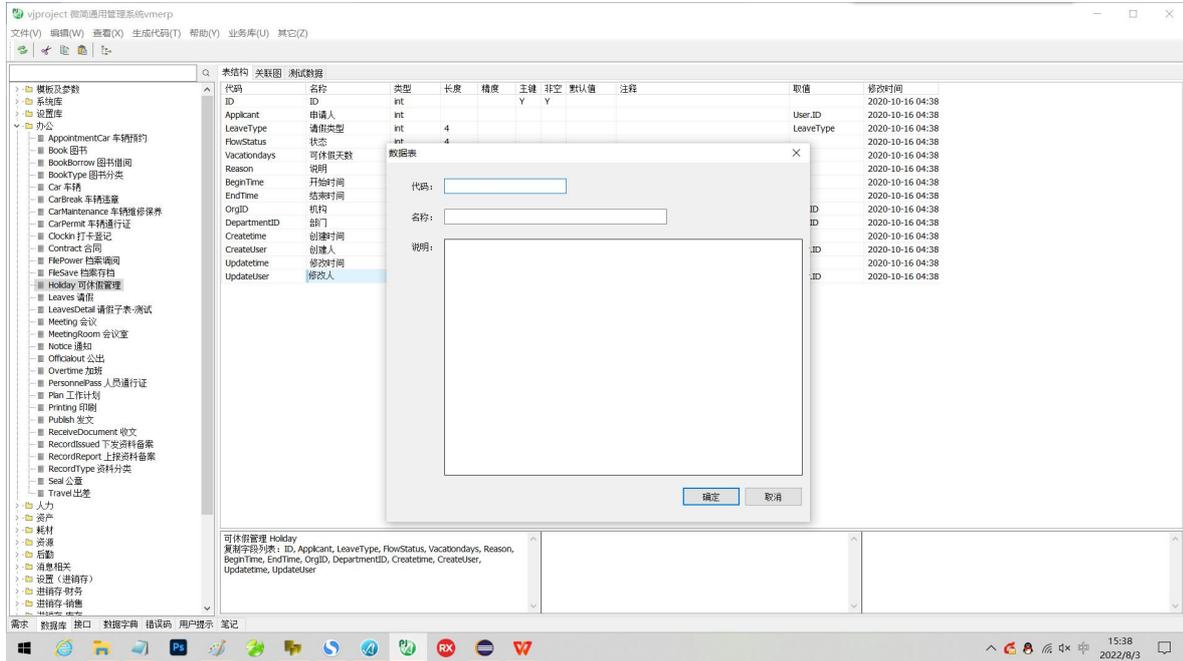
6.1 添加数据库

这个数据库不是常规意义上的数据库，只是对数据表做一个分类。

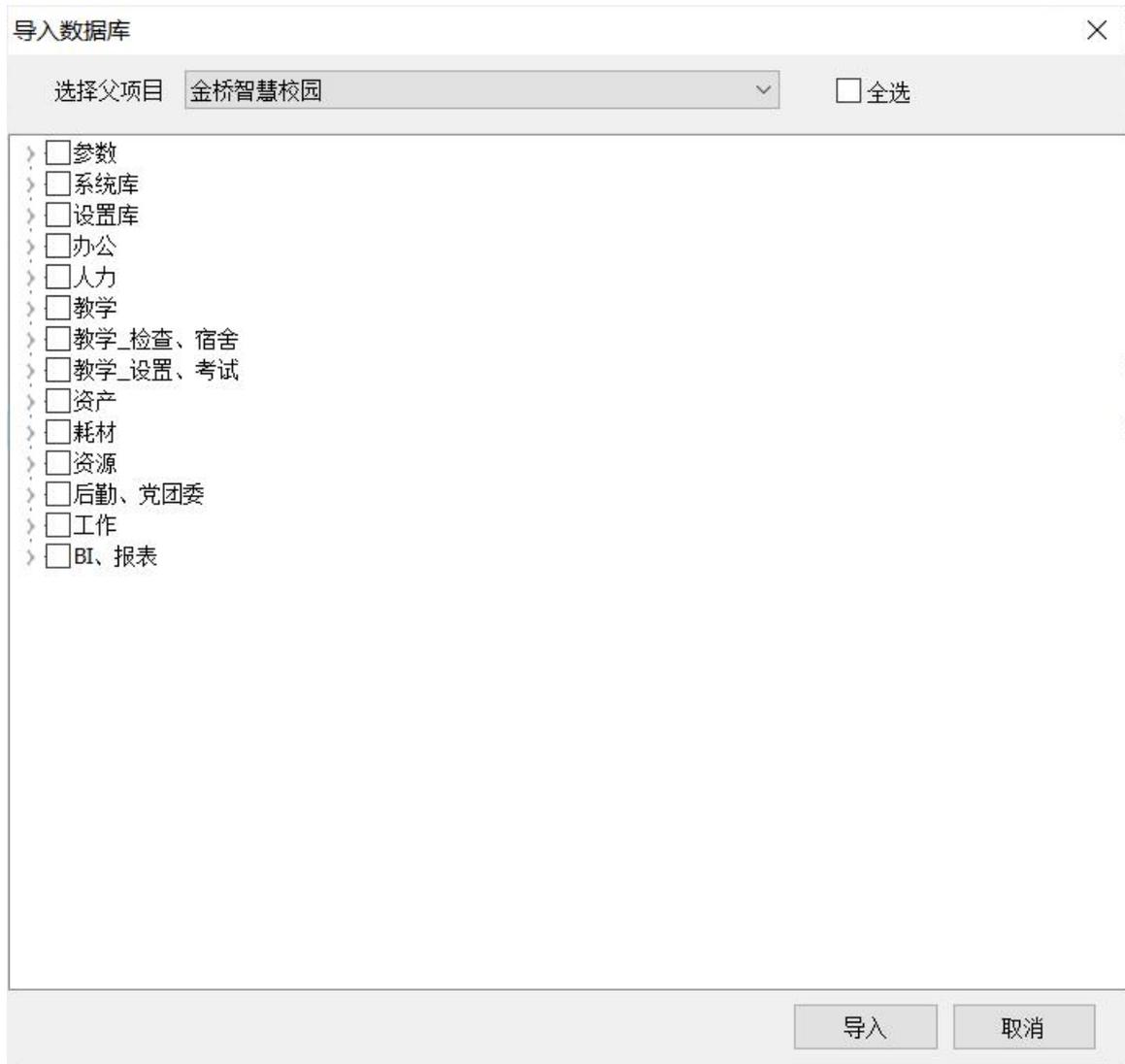


6.2 添加表

添加一个表设计。

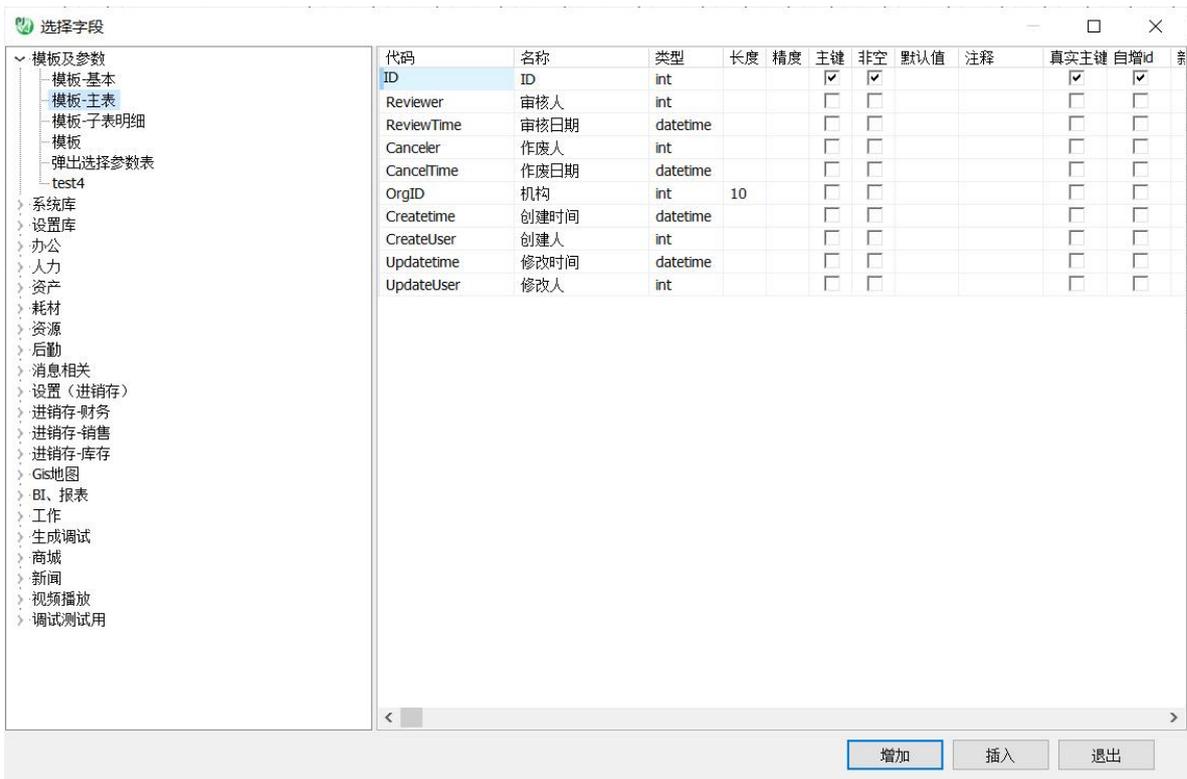
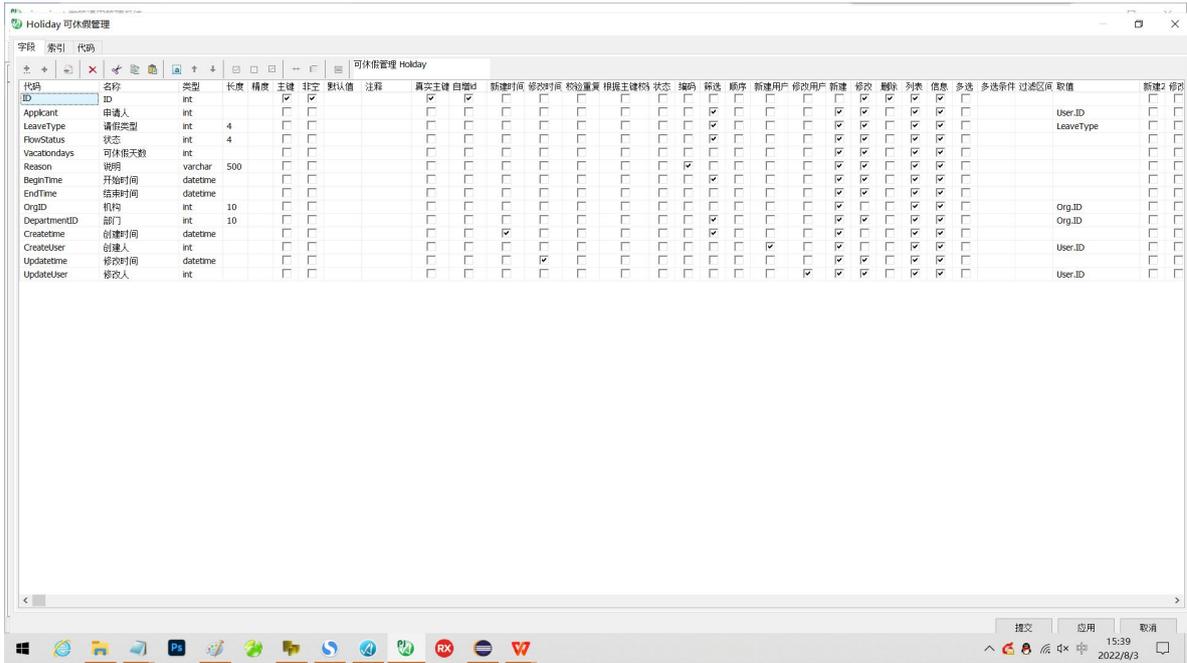


6.3 从父项目导入



6.4 设计表结构

批量编辑，从模板里选择字段，添加或插入。



除了 id，其他字段不设置非空（其他地方控制）。

表名：不要带“表”“管理”。“记录”如果是辅助性表，就加记录，比如收款记录等。

标准业务表应该有：ID、OrgID、Createtime、CreateUser、Updatetime、UpdateUser。

字段名和界面上的标签进来完全一致，这样生成代码、注释可以完全一致，避免

歧义，减少沟通成本。

字段补全。

特别注意：表名的命名，以可读性为第一。尽量用单个单词。如果单个单词无法表达、或在整个库中有歧义，用两个单词，谓语+主语，比如学生请假，用 LeaveStudent。

所有 workflow 相关的表：FlowStatus 状态（系统字段）。流程相关的过程：处理人、处理时间、处理意见都不用管。

字段名、表名：首字母大写，驼峰式，每个单词首字母都大写。

主子表、明细表：Order、OrderDetail。加 Detail。字表里必须有主表 id: OrderID。

7 配置数据库

勾选数据库表字段选项，用于生成接口、java 代码、前端数据表格等内容。

7.1 批量配置数据表

先对数据库做批量配置：主菜单——生成代码——批量配置。勾选相应的选项，然后确定即可。可以对全部表、部分表进行操作。

7.2 手工配置

每个表需要再单独设置。

需要配置内容：

- 1、自增 id；
- 2、新建、修改、删除、信息（详情）的字段内容。
- 3、该字段是否新建时间、修改时间、新建用户、修改用户。
- 4、校验重复。
- 5、汉字：是否需要编码传输。前端组件如果是多行编辑框，需要配置。
- 6、筛选字段：可以多选一些，不能少于实际筛选项。
- 7、取值：对应的字典、其他字典。

其它选项可不管。

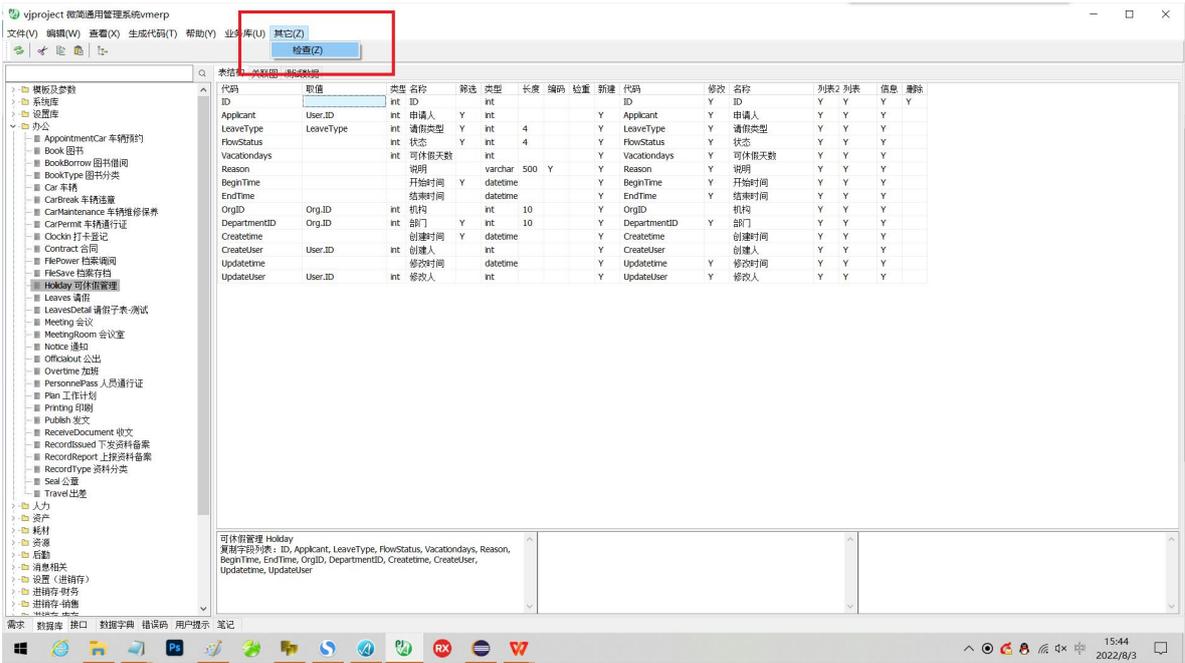
具体勾选之后的效果，生成接口、生成代码的时候对比一下就明白了。

7.3 检查

勾选配置之后，点击工具栏：其它——检查。

配置完后，至少检查三遍，基本能保证不出错，也可以两个人交叉检查。

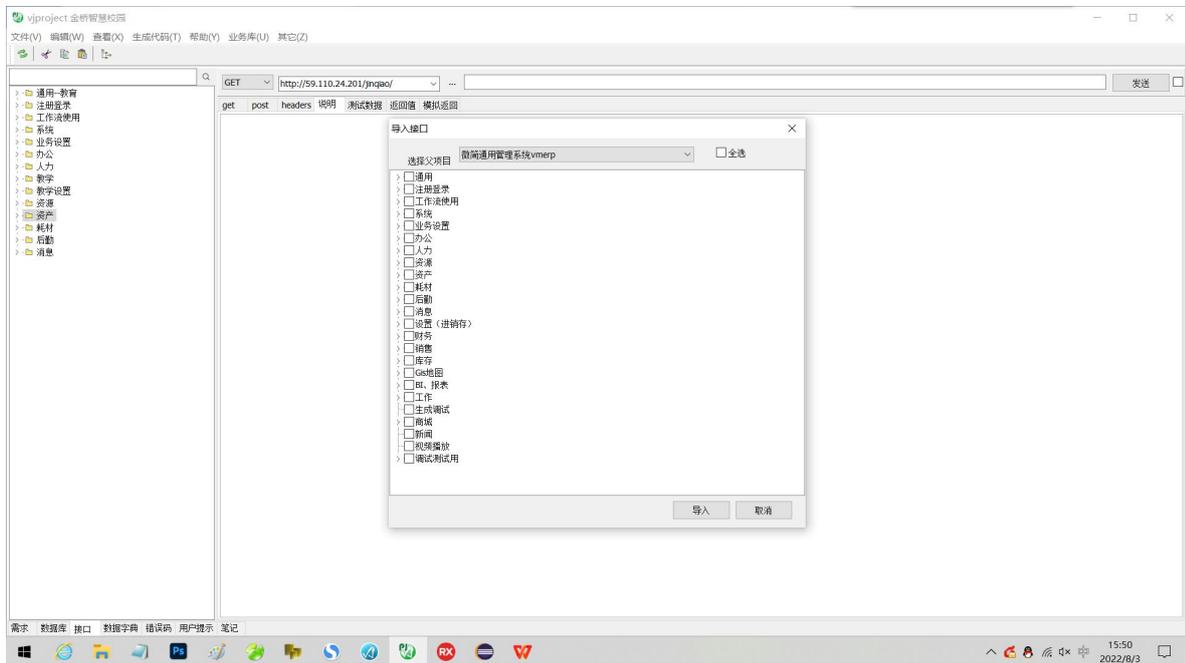
检查的时候要逐项进行，也就是检查完一项再检查另一项。检查内容：取值、类型、筛选、长度、汉字、校验重复、增删改查详情等。



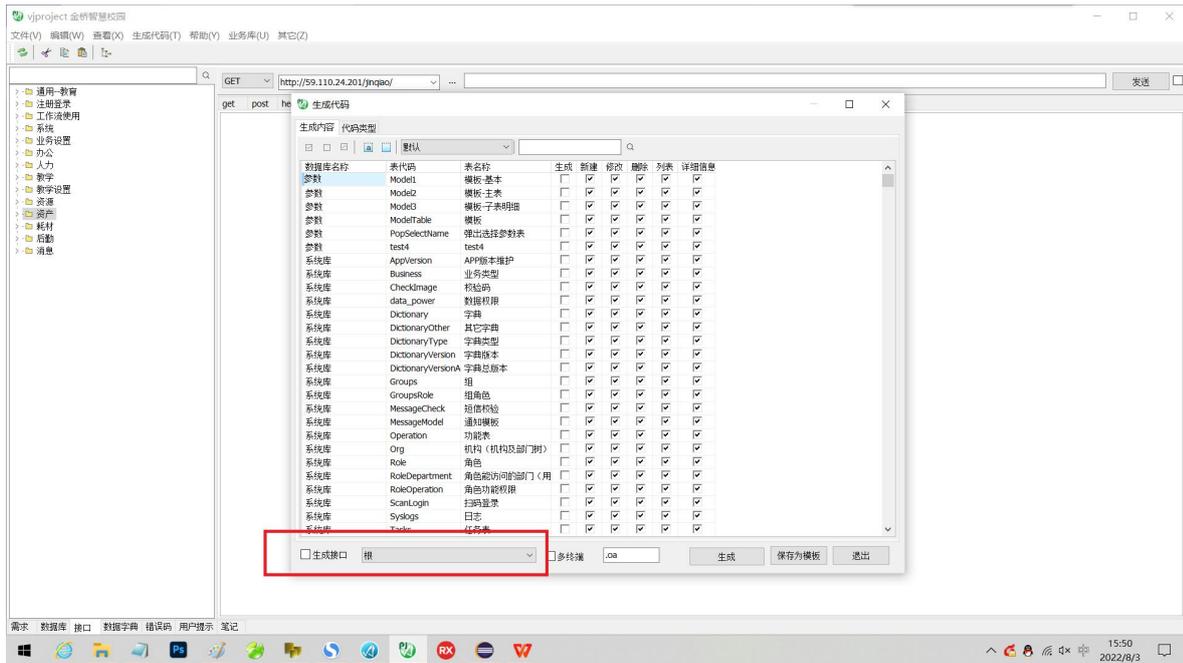
8 接口设计

8.1 接口定义的几个来源

1、从父项目导入进来。



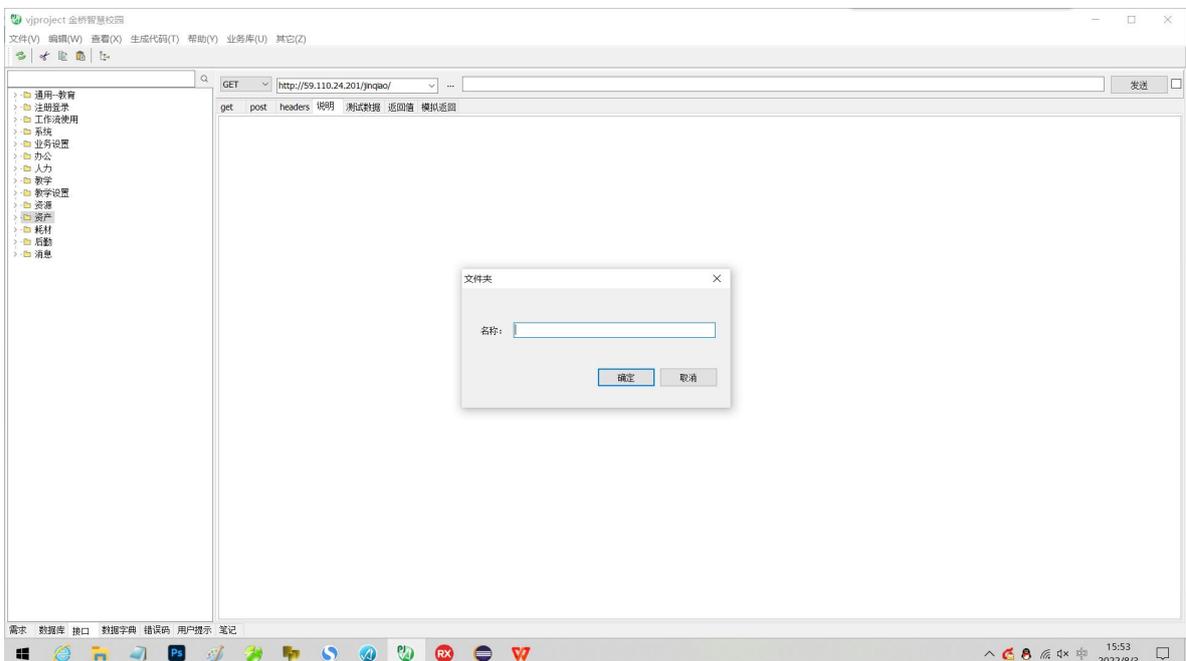
2、从数据库自动生成（生成 java 代码时有选项）。



3、手工设计接口。

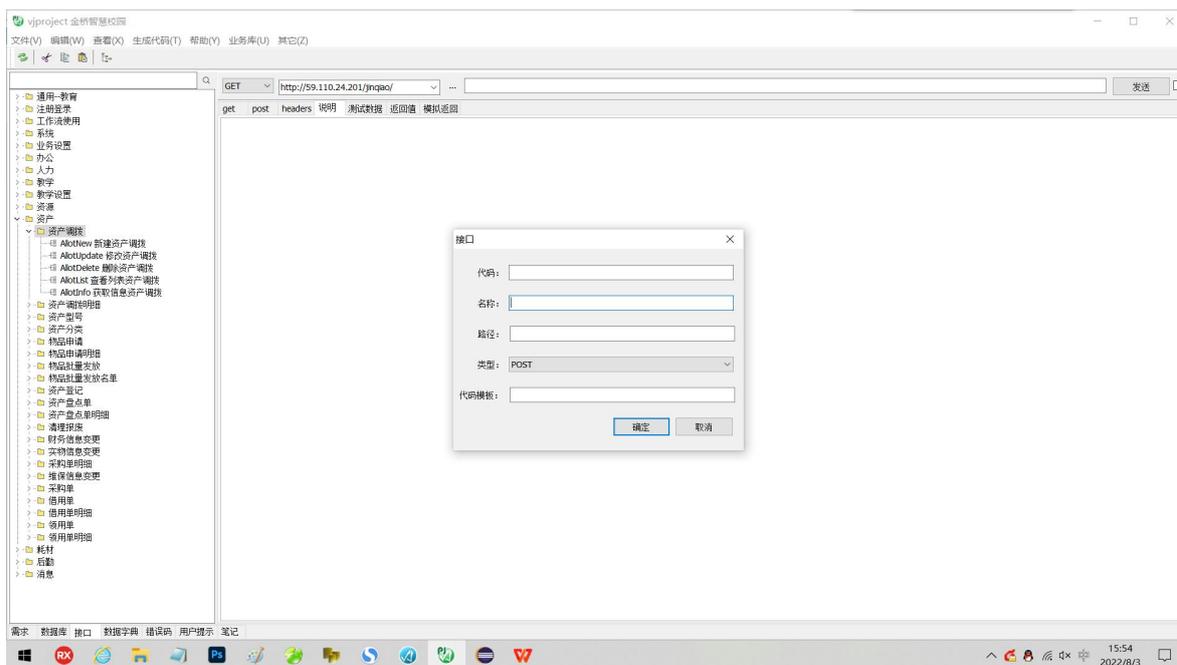
8.2 添加文件夹、子文件夹

文件夹是对接口的一个分类。



8.3 添加接口

添加接口，只需输入代码、名称、类型即可。其它两个属性不用管。



基本原则：逻辑尽量放到后端来做，因为逻辑比较复杂，需要沟通需求，后台对需求时比较熟悉的，后台写逻辑省去了沟通需求的过程。而且后台代码的灵活性比前代要很多。

■ 手工设计的时候注意：

新建接口的时候，使用批量插入功能，直接从库表中选择字段。

注意 get、post 参数。Header 部分的参数是统一的，用复制，粘贴，不要自己写。

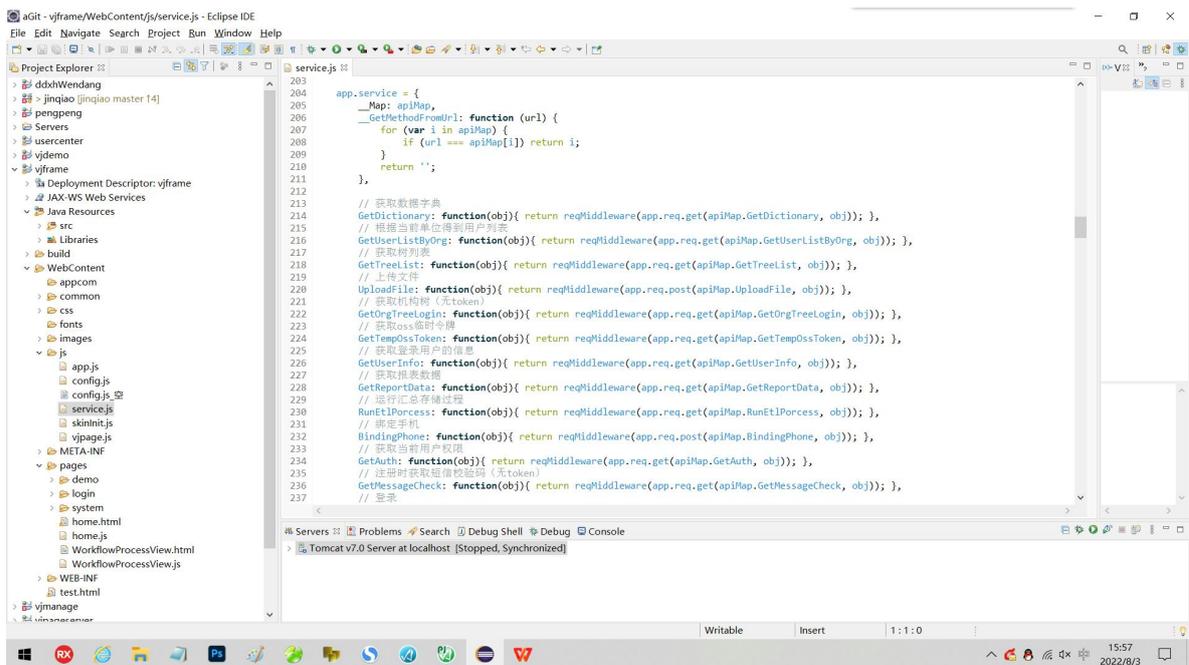
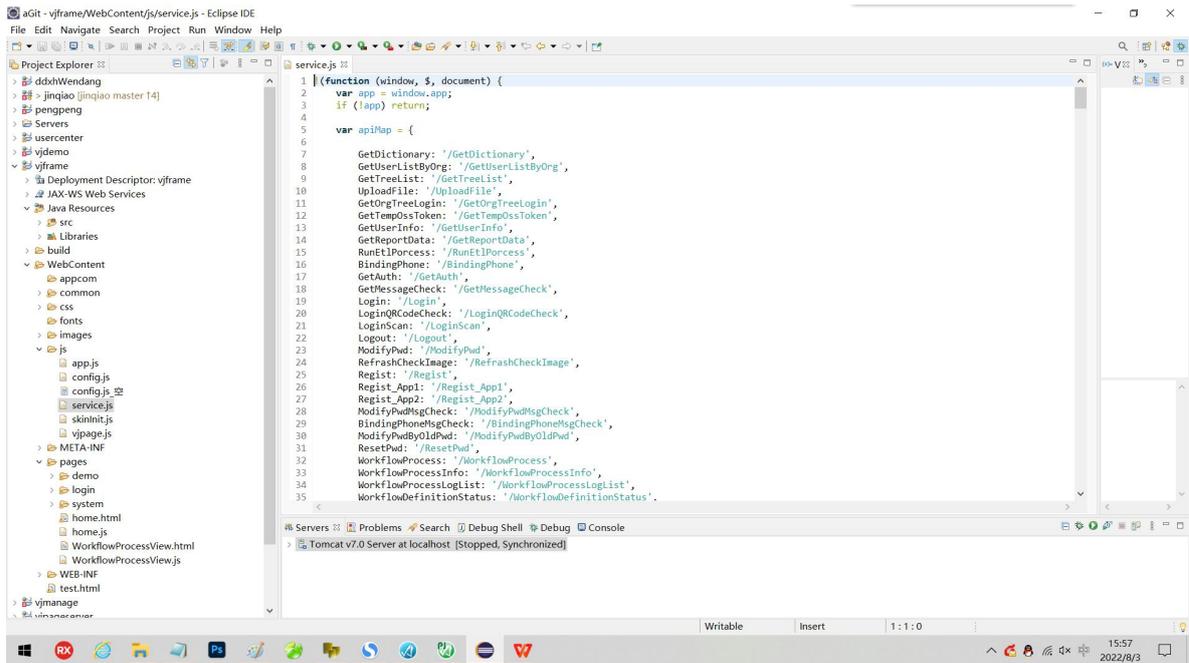
说明部分也是从其他地方复制过来，然后再改，统一一个格式。

接口设计没有特殊规定，按照常规方式即可。

8.4 复制接口定义

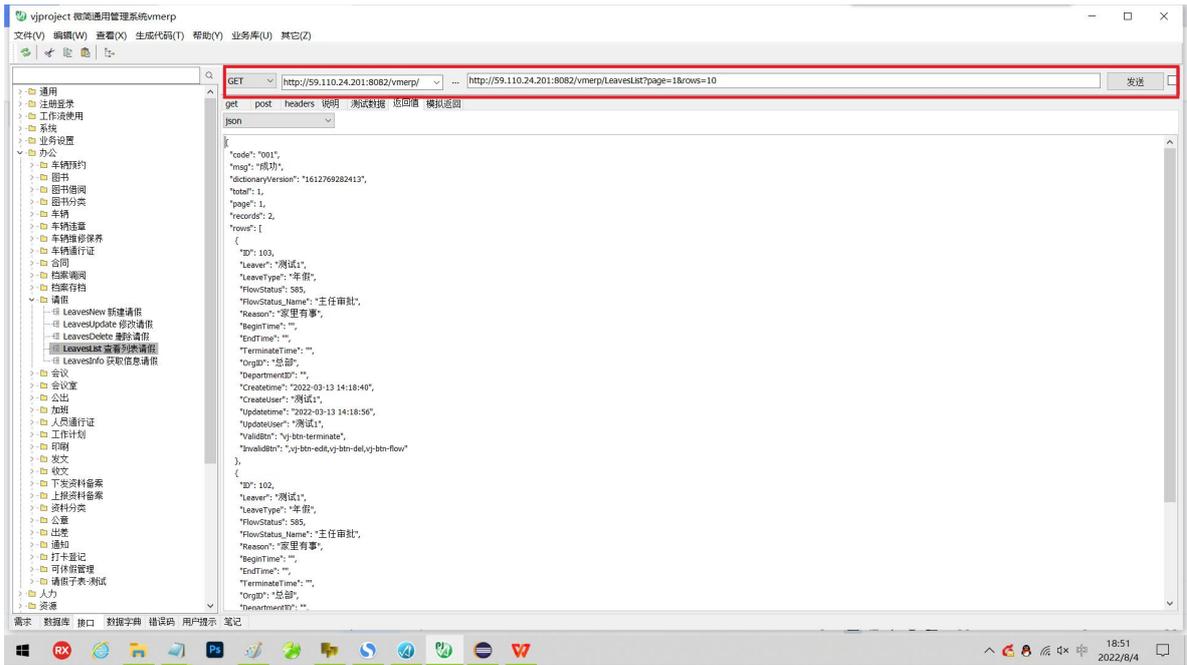
前端使用的时候，vjframe 中，在 js/service.js 中配置接口定义。

在 vjproject 中切换到接口页，选中任意树节点，右键弹出菜单，复制接口定义。粘贴到前端对应的接口定义部分即可。先粘贴到记事本里，然后再选择对应部分往代码中粘贴。

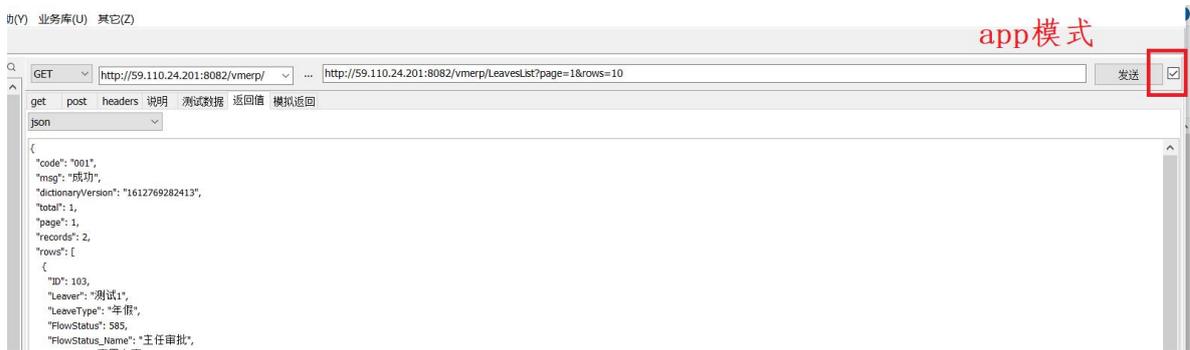


8.5 接口调试

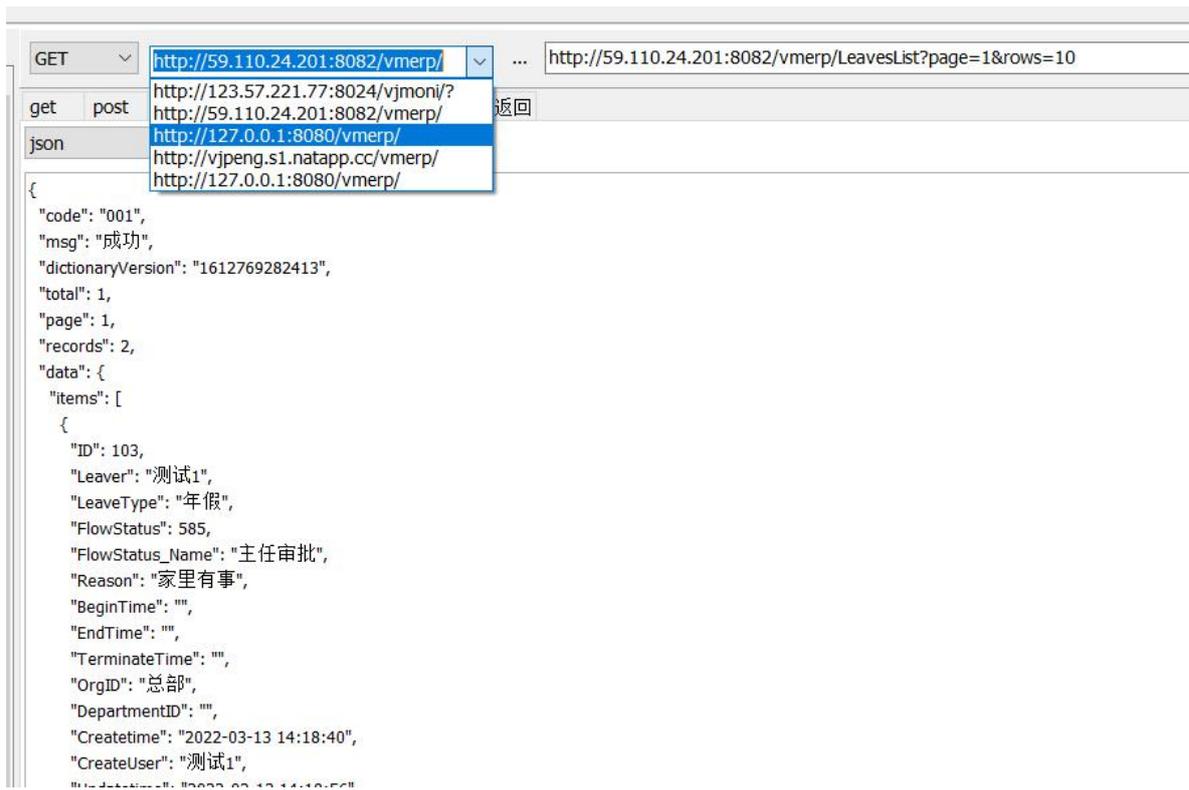
可以直接通过 vjproject 进行接口调试。



List 接口中，Pc、app 同时调试是，如果勾选了这个选项，则返回的为 app 模式的列表。



可以不同的地址切换



ip 路径地址在 vjmanage 中统一设置。也可以在本机自己设置。



9 前端页面命名

使用 vjpage 制作的前端页面，可以直接生成前端开发代码。生成之前，需要进行配置、命名。

可对页面元素进行附加样式、弹出选择参数、字段的命名。

在 vjpage 中，打开对应页面，主菜单：其他——修改名称。

命名的具体操作方式：选中页面中元素，双击左侧树节点对应的内容即可。

字段命名会自动修改对应的组件 name 属性。表单字段 name 属性会设置为该配置项。



附加样式命名会修改“开发命名”属性。生成前端开发代码时，会在 class 中包含该配置项。



注意：每个页面模板命名都不一样。需要根据前端代码模板进行相应的配置。

9.1 附加样式命名

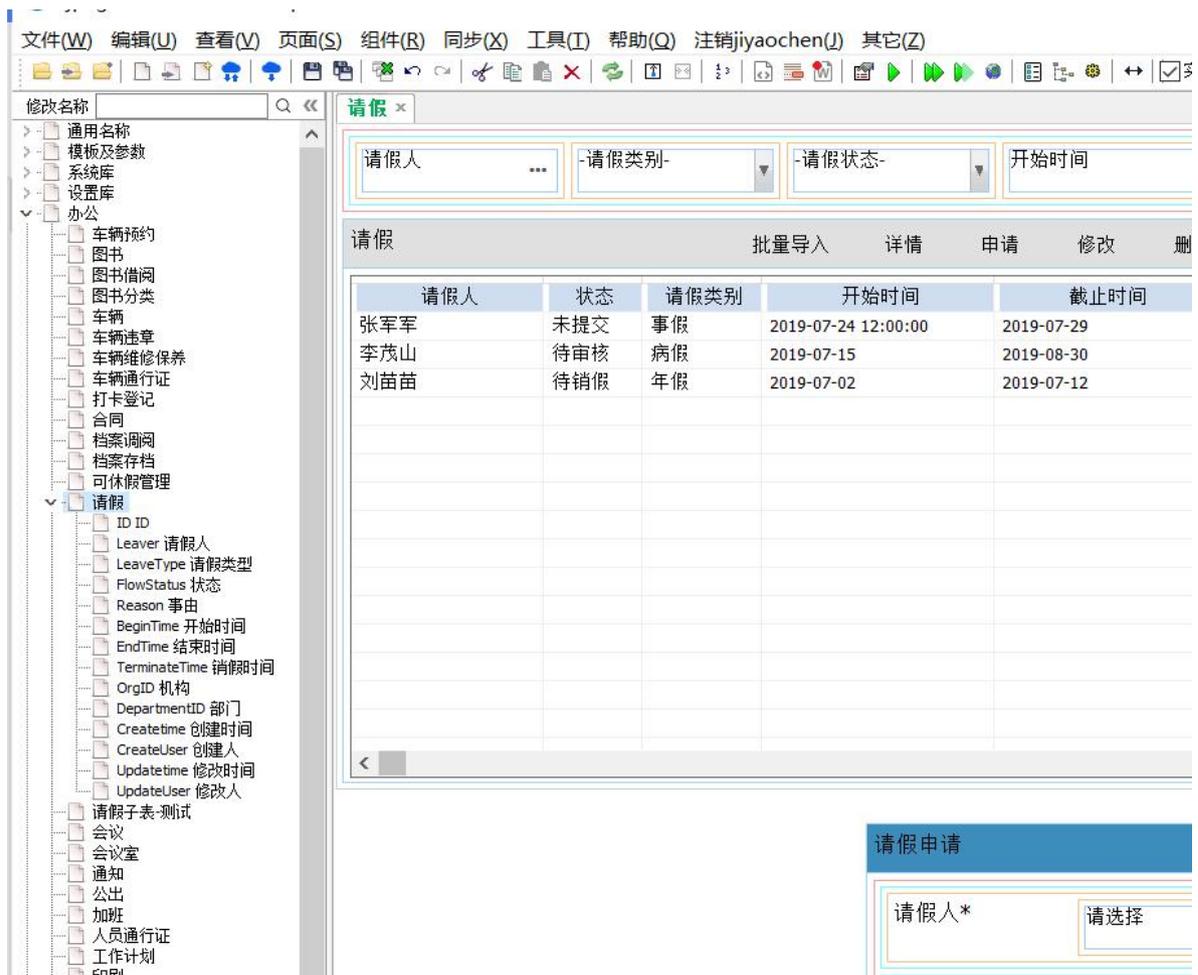
打开页面，选中对应的组件，然后双击左侧树。比如，选中添加按钮，单击左侧添加(vj-btn-add)。把对应的组件全部命名即可。



9.2 字段命名

必须先要在 vjmanage 中进行项目关联。把 vjpage 项目和 vjproject 的项目关联起来。这样在命名时才能显示表结构、字段。

选中表单组件，然后双击树上对应的字段即可。



9.3 其他注意的:

日期、时间字段必须和数据库一一对应。

编辑里：开始时间、结束时间，拆成两个字段。

筛选区：日期段选择、日期时间段选择。

编辑：日期下拉选择、日期时间下拉选择。

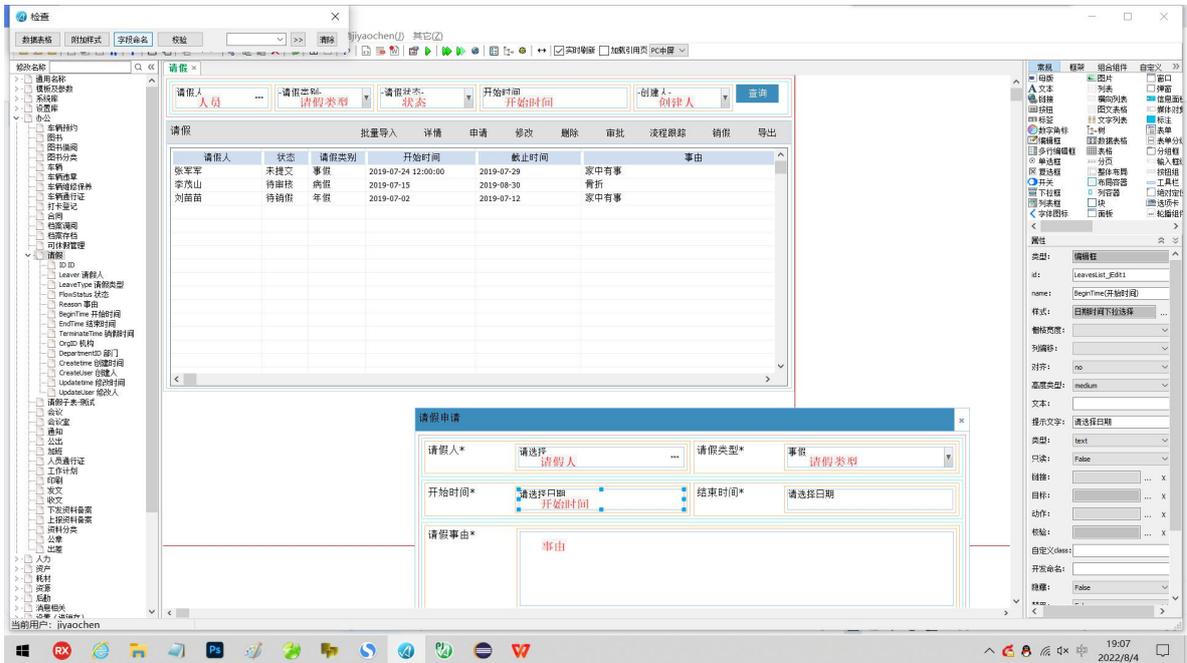
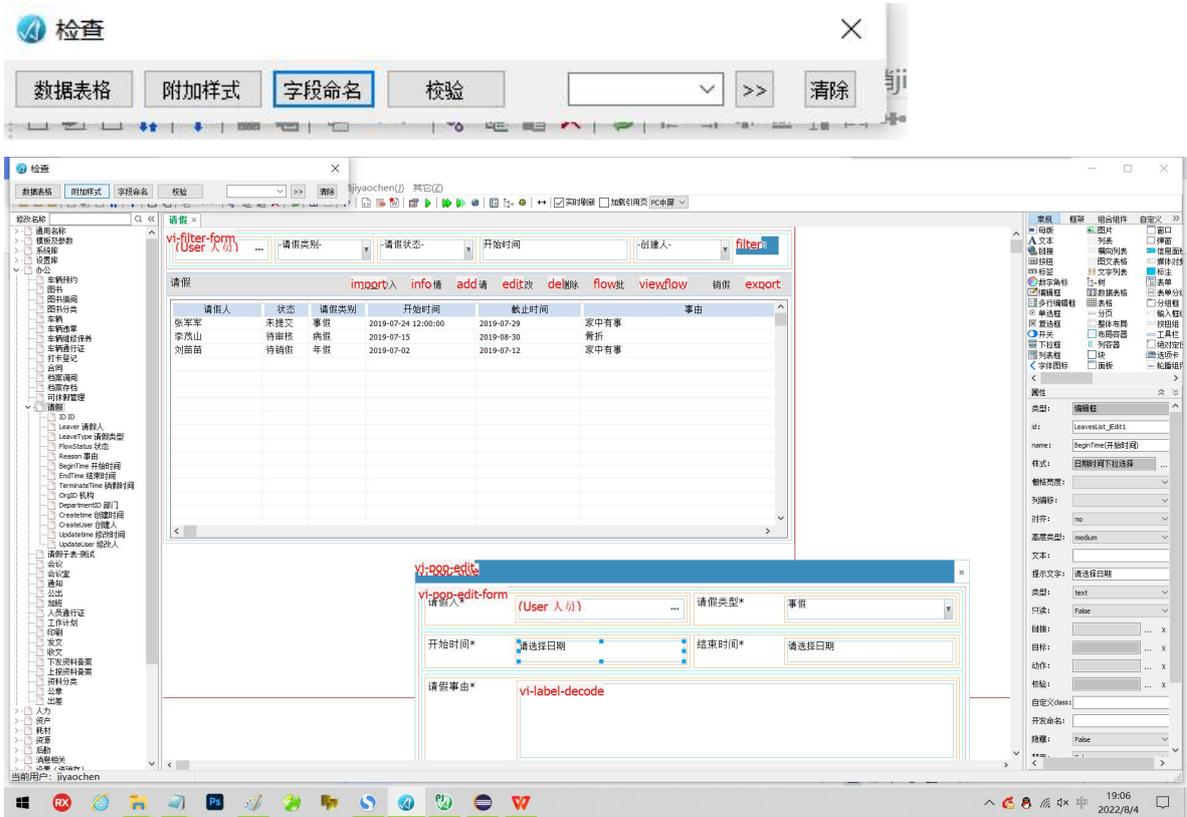
批量命名，按照内容逐项进行，比如：字段命名、校验。所有页面字段命名完成之后再作校验。

组件命名时会影响到校验。校验在做需求的时候尽量加上。正式开发的时候，要重新过一遍校验的内容。因为复制粘贴组件页面、需求的变化等因素，校验肯定是乱了，需要重新梳理。

9.4 检查

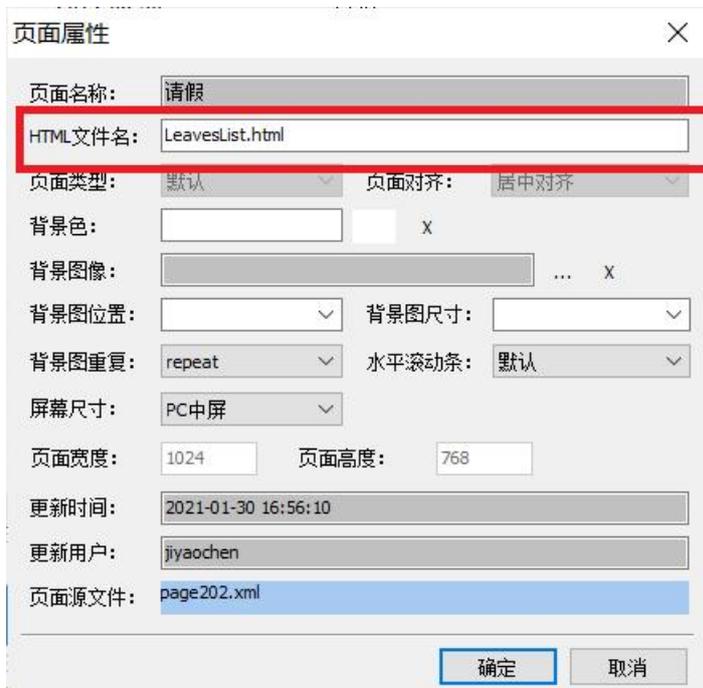
vjpage 中主菜单：其他——检查。

命名完成之后，要检查三次。检查的意义非常重要，能够大量减少测试的 bug。生成代码是固定的，不会有错，但大量详细的配置可能出错。



9.5 页面命名

对页面进行命名，这个名称就是生成 html 文件后的名称。打开页面，右键——页面属性。



9.6 页面命名规则

主页面，表名+List.html。示例：RepairList.html

独立编辑页，表名+Edit.html。示例：RepairEdit.html

独立流程页，表名+Flow.html。示例：RepairFlow.html

独立详情页，表名+Info.html。示例：RepairInfo.html

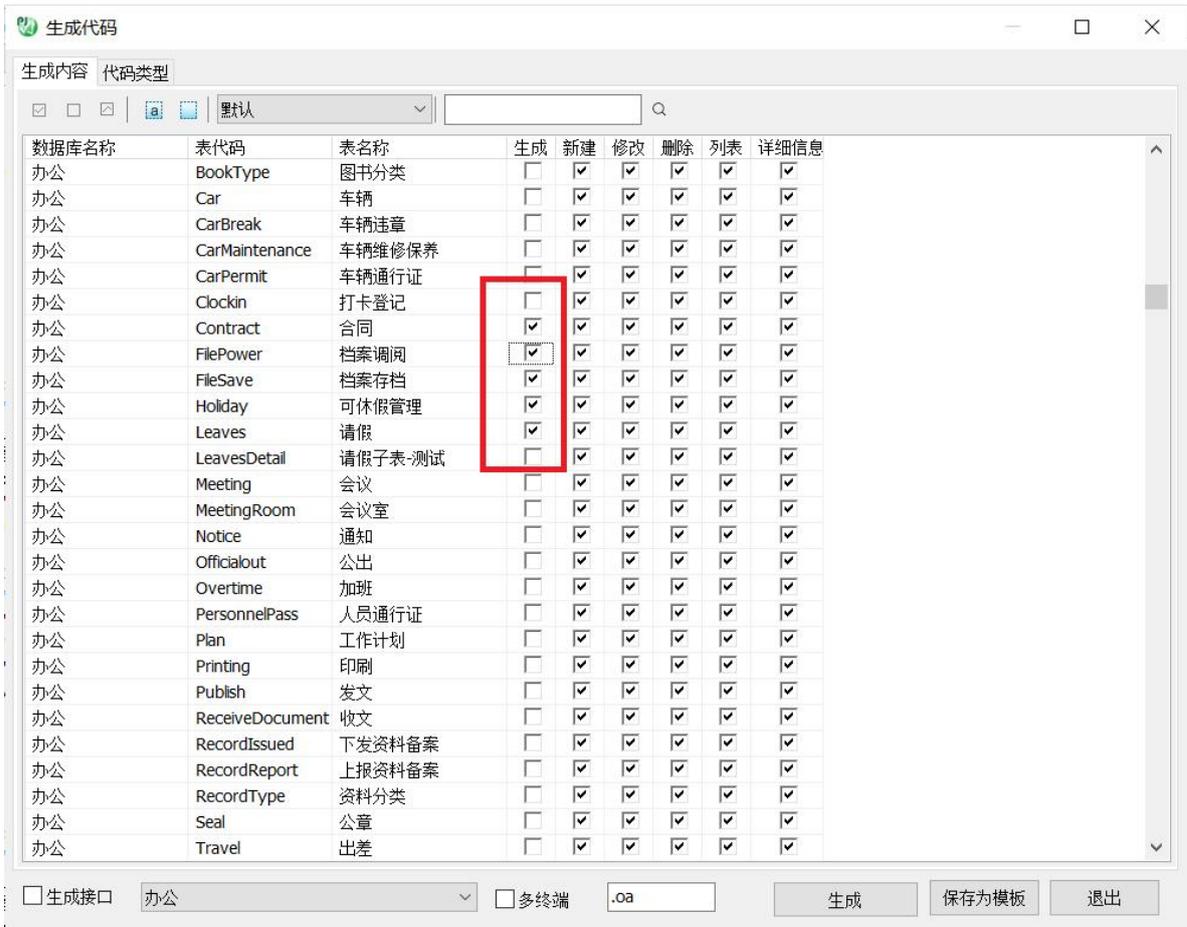
要做到只看页面名称，大概就能判断是那个功能。

10 生成 java 代码

10.1 生成代码

打开 vjproject，主菜单：生成代码——根据数据库生成。

勾选要生成的表。生成时可选择是否生成接口，可指定生成到哪个接口文件夹下。



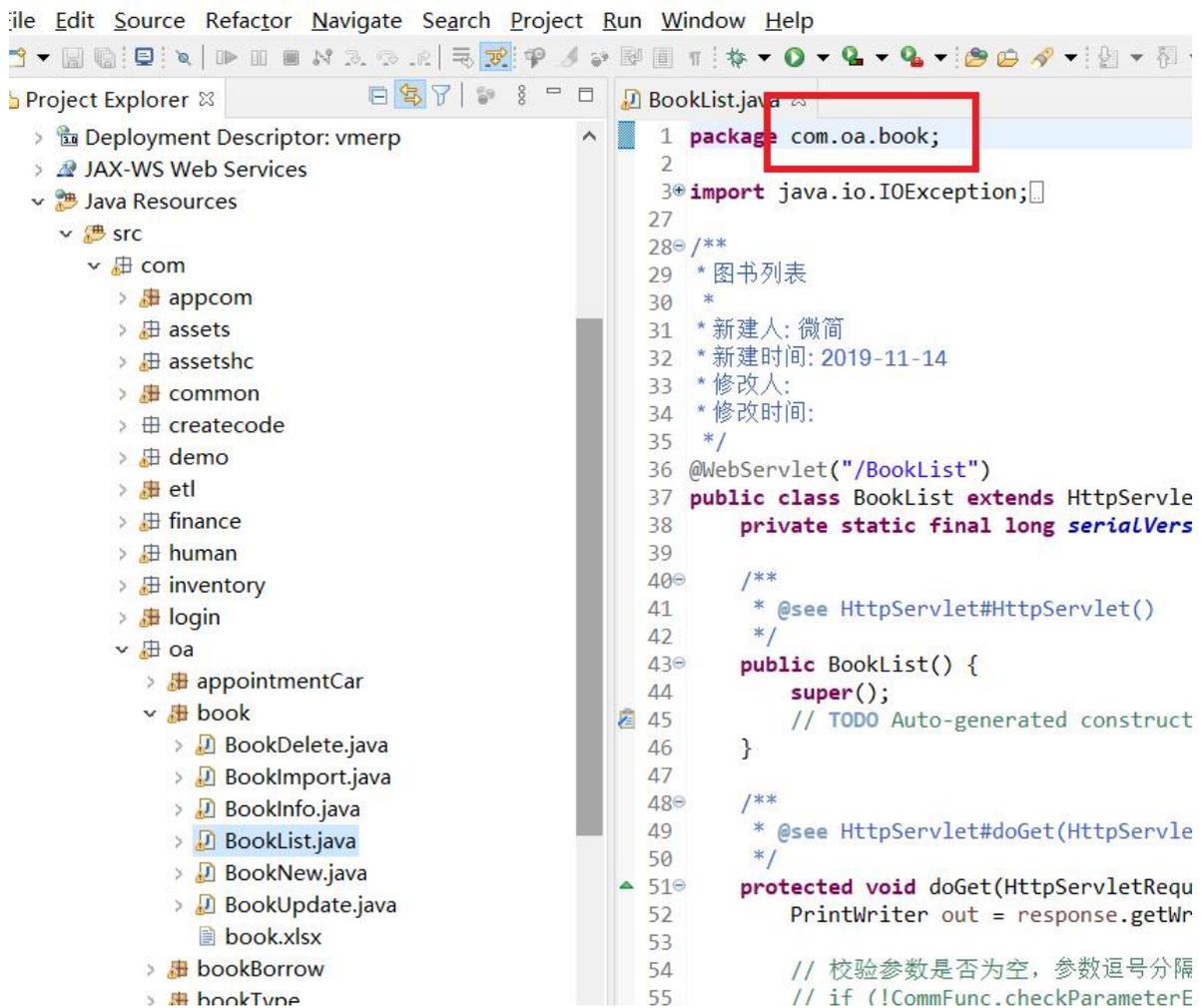
多终端，指的是 list 接口生成 pc、app 两种返回集合。

```

154         if (loginTerminal.equals("pc")) { // pc端
155             while (rs.next()) {
156                 Map<String, Object> row = new LinkedHashMap<String, Object>(); // 每一行数据
157
158                 row.put("ID", rs.getInt("ID")); // ID
159                 row.put("Leaver", CommFunc.getNameById("User", rs.getInt("Leaver"))); // 请假人_名称
160                 row.put("LeaveType", CommFunc.getNameById("LeaveType", rs.getInt("LeaveType"))); // 请假类型_名称
161                 row.put("FlowStatus", rs.getInt("FlowStatus")); // 状态
162                 row.put("FlowStatus_Name", CommFunc.getNameById("WorkFlowDefinition", rs.getInt("FlowStatus"))); //
163                 row.put("Reason", CommFunc.filterStr(rs.getString("Reason"))); // 事由
164                 row.put("BeginTime", CommFunc.datetimeToString(rs.getTimestamp("BeginTime"))); // 开始时间
165                 row.put("EndTime", CommFunc.datetimeToString(rs.getTimestamp("EndTime"))); // 结束时间
166                 row.put("TerminateTime", CommFunc.datetimeToString(rs.getTimestamp("TerminateTime"))); // 销假时间
167                 row.put("OrgID", CommFunc.getNameById("Org", rs.getInt("OrgID"))); // 机构_名称
168                 row.put("DepartmentID", CommFunc.getNameById("Org", rs.getInt("DepartmentID"))); // 部门_名称
169                 row.put("Createtime", CommFunc.datetimeToString(rs.getTimestamp("Createtime"))); // 创建时间
170                 row.put("CreateUser", CommFunc.getNameById("User", rs.getInt("CreateUser"))); // 创建人_名称
171                 row.put("Updatetime", CommFunc.datetimeToString(rs.getTimestamp("Updatetime"))); // 修改时间
172                 row.put("UpdateUser", CommFunc.getNameById("User", rs.getInt("UpdateUser"))); // 修改人_名称
173
174                 data.add(row);
175             }
176
177             JsonResult.put("rows", data); // jqgrid格式
178
179         } else {
180             while (rs.next()) {
181                 Map<String, Object> row = new LinkedHashMap<String, Object>(); // 每一行数据
182
183                 row.put("ID", rs.getInt("ID")); // ID
184                 row.put("Leaver", CommFunc.getNameById("User", rs.getInt("Leaver"))); // 请假人_名称
185                 row.put("LeaveType", CommFunc.getNameById("LeaveType", rs.getInt("LeaveType"))); // 请假类型_名称
186                 row.put("FlowStatus", rs.getInt("FlowStatus")); // 状态

```

可设置包名。



```
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
  Deployment Descriptor: vmerp
  JAX-WS Web Services
  Java Resources
    src
      com
        appcom
        assets
        assetshc
        common
        createcode
        demo
        etl
        finance
        human
        inventory
        login
      oa
        appointmentCar
        book
          BookDelete.java
          BookImport.java
          BookInfo.java
          BookList.java
          BookNew.java
          BookUpdate.java
          book.xlsx
        bookBorrow
        bookType
BookList.java
1 package com.oa.book;
2
3 import java.io.IOException;
27
28 /**
29  * 图书列表
30  *
31  * 新建人: 微简
32  * 新建时间: 2019-11-14
33  * 修改人:
34  * 修改时间:
35  */
36 @WebServlet("/BookList")
37 public class BookList extends HttpServlet
38     private static final long serialVersionUID =
39
40 /**
41  * @see HttpServlet#HttpServlet()
42  */
43 public BookList() {
44     super();
45     // TODO Auto-generated construct
46 }
47
48 /**
49  * @see HttpServlet#doGet(HttpServletRequest
50  */
51 protected void doGet(HttpServletRequest Requ
52     PrintWriter out = response.getWr
53
54     // 校验参数是否为空, 参数逗号分隔
55     // if (!CommFunc.checkParameterE
```

选择要生成的表，单击生成即可。

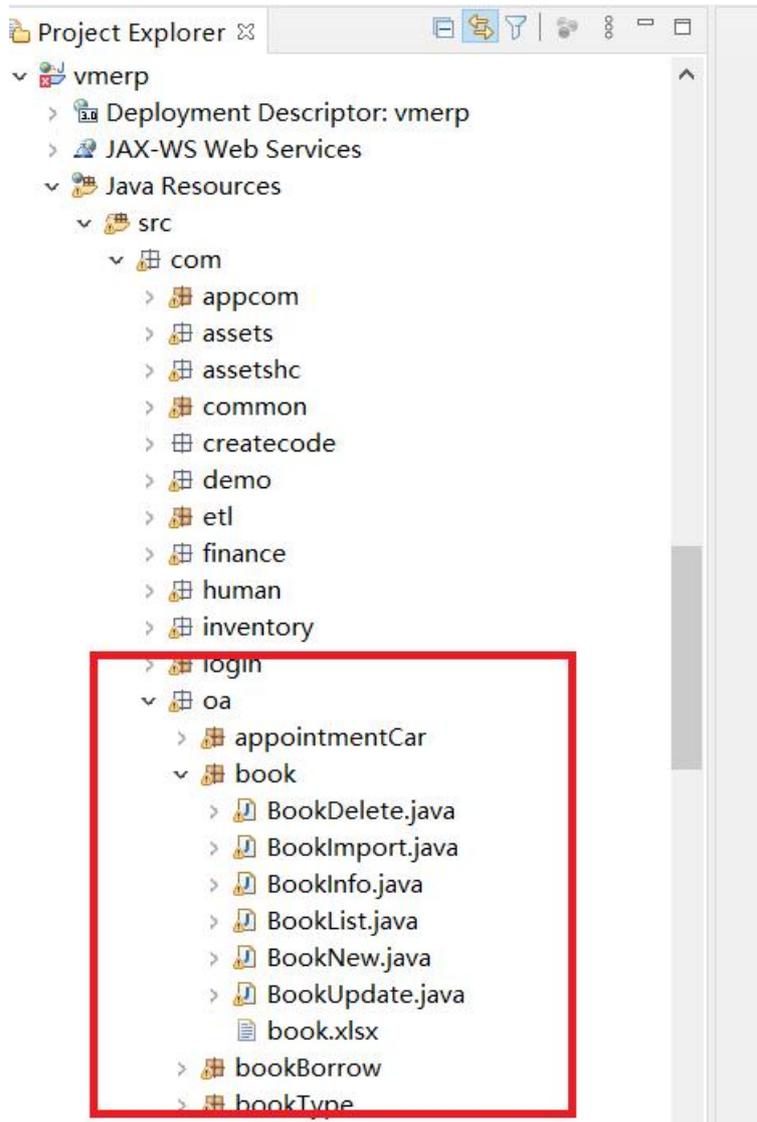
生成后的路径会自动打开。vjproject\createcode\code\api。

10.2 使用方法

生成后的代码复制到 vjframe 代码框架中对应部分即可使用。

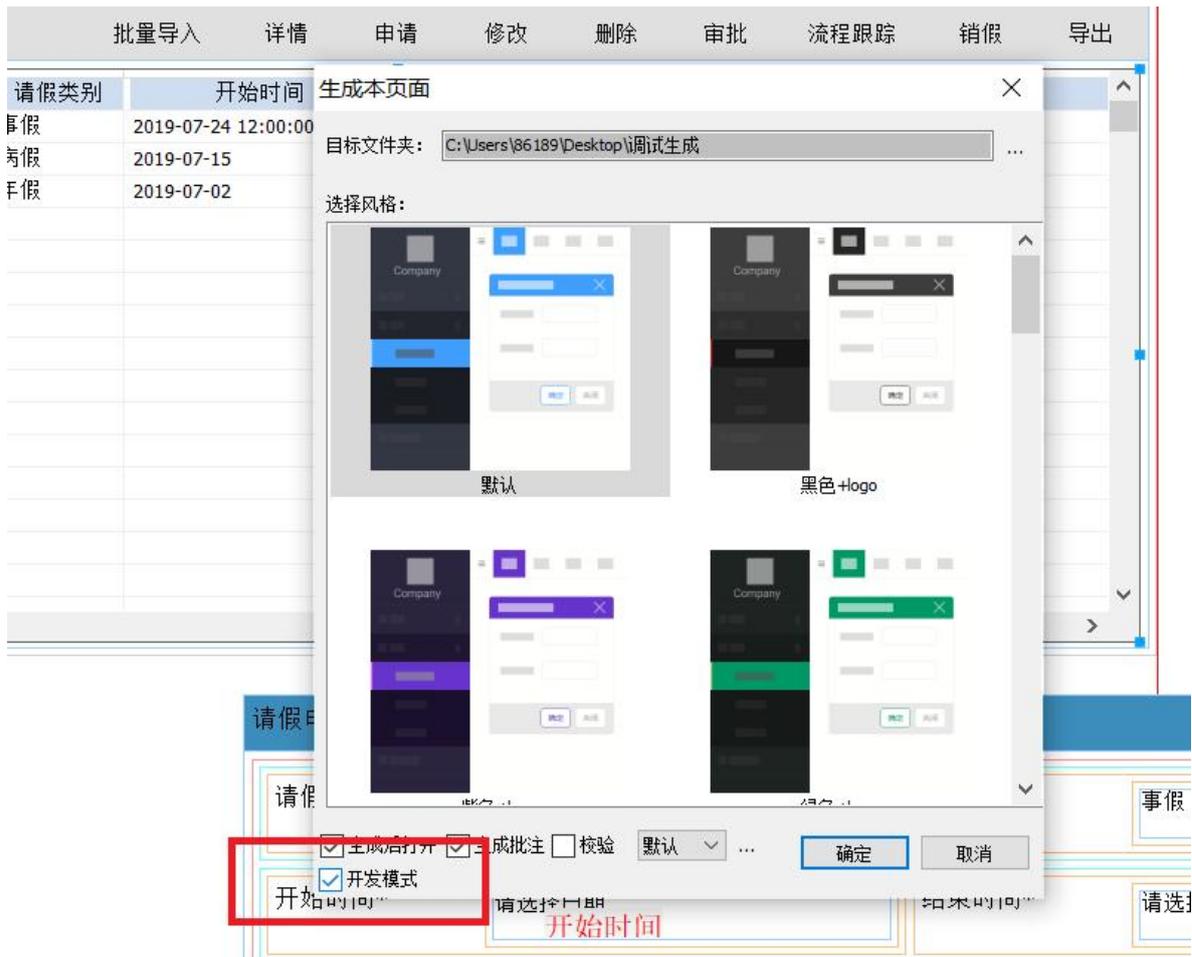
复制到 vjframe 中后，如果有错误，可直接手工调整。

生成的代码不可逆回来。生成后，发现错误必须先修改数据表及设置，然后重新生成。如果已经修改，备份修改的——重新生成——把修改部分拷过来。



11 生成前端代码

vjpage 默认生成原型代码。生成页面时，选中开发模式选项即可。



11.1 前置条件

- 1、页面之前必须配置好。
- 2、配置文件中配了相关项。pageFileConfig.txt
- 3、vproject 中有对应的表。

11.2 设置配置文件

■ 打开配置文件

vjpage 中，主菜单：其它——打开开发配置路径。

打开 pageFileConfig.txt。

对相关页面做配置。

■ 复制表名定义

**配置内容复制：vjpage 中，主菜单：其它——复制表名定义。已经页面命名之后，系统可自动复制表名定义，但不完全准确。粘贴到配置文件中之后，根据实际接口再做修改。

■ 配置说明:

示例:

请假管理,LeavesList.html,Leaves,List

请假编辑,LeavesEdit.html,Leaves,Edit

请假审批,LeavesFlow.html,Leaves,Flow

四列分别是: 页面名称,页面 html 文件名, 表名, 类型

类型包括:

list 编辑弹出、详情右侧停靠

list_pop_pop 编辑弹出、详情弹出

list_full_right 编辑充满、详情右侧停靠

list_btnOnline 按钮在行上 (弹出-右侧)

list_sub 列表带子表

listiframe 编辑详情都是 iframe 引用

listiframe_edit 列表 iframe 对应的独立编辑页

listiframe_info 列表 iframe 对应的独立详情页

listflow 列表流程 (弹出-右侧)

listflow_sub 列表流程、带子表 (弹出-右侧)

listflow_flow 列表流程 iframe 对应的独立审批页

treeList 左侧树、右侧列表 (弹出-右侧)

treeEdit 左侧树、右侧编辑

treeEdit_edit 左树右编辑, 右侧编辑对应的独立编辑页

info 独立的详情页, 仅查看详情

edit 独立的编辑页, 添加、编辑

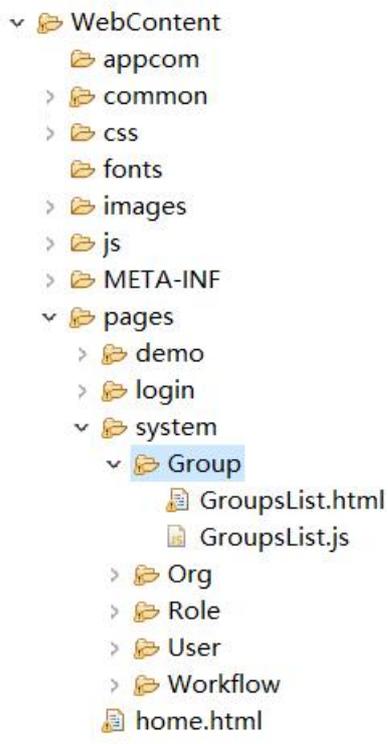
dictEdit 字典维护页

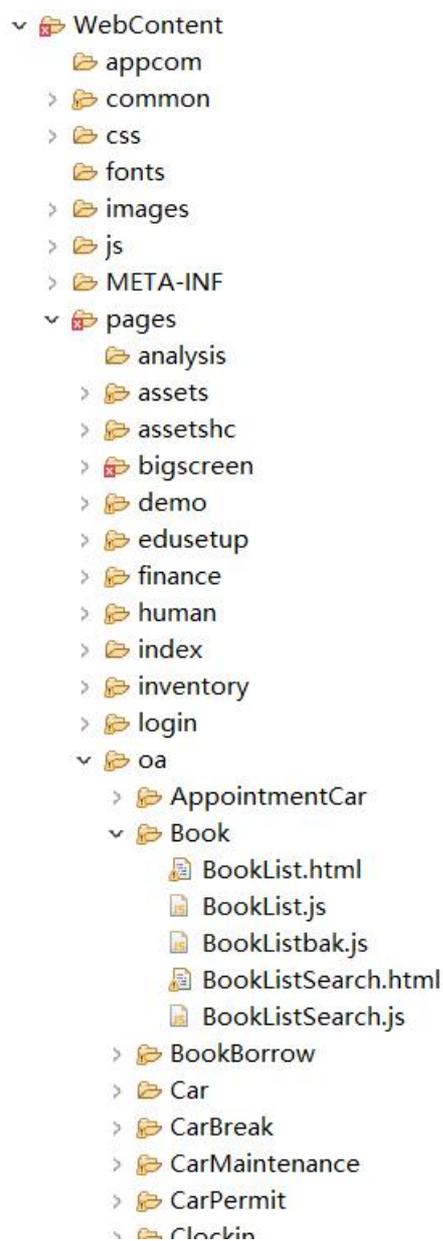
■ 模板扩展

开发代码模板可以自己扩展, 对应的配置文件: devcode.xml。

11.3 使用方法

生成代码后复制到 `vjframe` 对应的目录里就可以用。类似这样的目录结构。





12 生成后前后端代码修改

不管是前端代码还是后端代码，自动生成的时候，不可能全部配出来，也不可能全部配合理。所以生成之后，需要继续手工补充、修改、完善。

生成之后，可以到 `eclipse`、`vscode` 等工具中修改。代码部分即可完全脱离微筒工具。

13 vjmanage 项目管理

13.1 登录入口

<http://101.200.178.217:8080/vjmanage/pages/login/login.html>

注意: vjpage 用户账号必须加入到项目成员中才能登录本系统, 账号密码同 vjpage 一致。必须有 vjproject 项目才能使用 vjmanage。

13.2 编辑项目信息 (配置接口地址)

编辑项目信息

项目编号: pm22071482986p866232 *项目名称: 微筒体验项目

描述: 只读体验项目

立项时间: 2022-07-14

内部计划完成时间: 请选择日期 用户计划上线时间: 请选择日期

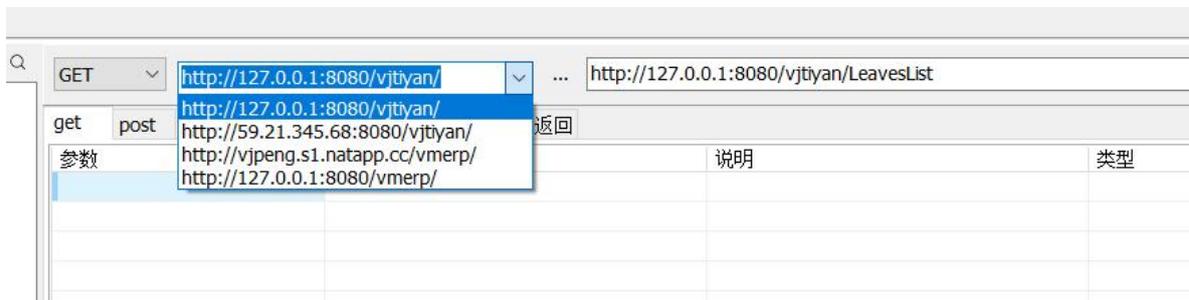
接口地址: **http://127.0.0.1:8080/vjtiyan/, http://59.21.345.68:8080/vjtiyan/**

SVN: 请输入

Git: 请输入

保存 恢复默认

这里配置的接口地址, 会在 vjproject 中接口部分显示。



13.3 添加项目成员、设置权限

添加的用户必须注册 vjpage。用户登录名就是 vjpage 的注册名。

项目角色 (配置 vjproject 中的权限): 项目经理有完全权限。需求人员有完全权

限。后端开发可生成 java 代码。

项目成员添加

*用户登录名

姓名

项目角色 项目经理 快速需求 需求人员 前端开发 后端开发 测试 实施人员
 其他 软件企业人员 甲方

特殊查看权限 查看全部 查看vjproject 查看项目计划 查看项目执行计划 查看项目成员
 查看项目通讯录 查看vjpage项目关联 查看代码备份 查看文档备份

特殊写权限 写需求 数据库设计 接口设计 写数据字典 写错误码 写用户提示 写笔记
 生成后台代码 生成前端开发代码 做测试数据 页面命名 写项目计划
 写项目执行计划 写项目通讯录 上传代码 上传文档

进入项目组时间

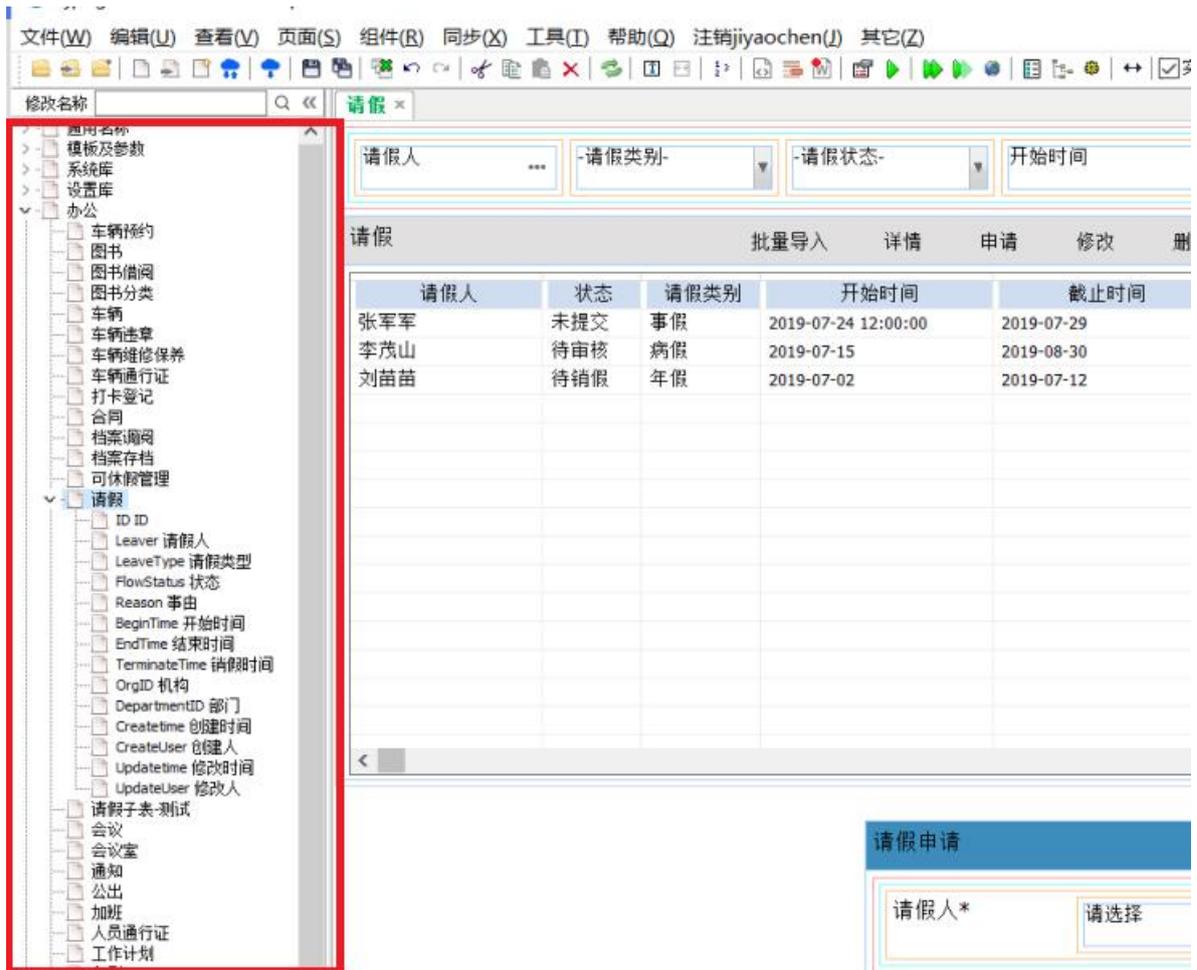
离开项目组时间

13.4 项目通讯录

项目中常用的通讯录。

13.5 vjpage 项目关联

必须配置 vjpage 项目关联，才能把数据库和前端及连接起来，才可以字段命名。否则，vjpage 中只有通用名称，不会出现数据库、表、字段。



输入 vjpage 项目的项目编号、项目名称，确定即可。



Vjpage 的项目编号：打开 vjpage，打开对应的项目，工具——项目属性。

项目属性

项目编号: 2022071615866s282002

项目名称: 测试项目

主题风格: 默认

框架: Bootstrap

框架版本:

页面数量: 0 统计非空页面数

config文件名: config.css X

索引文件名: index_all X

本地缓冲目录: D:\vjpageproject\project197\ 打开

确定 取消

13.6 指定父项目

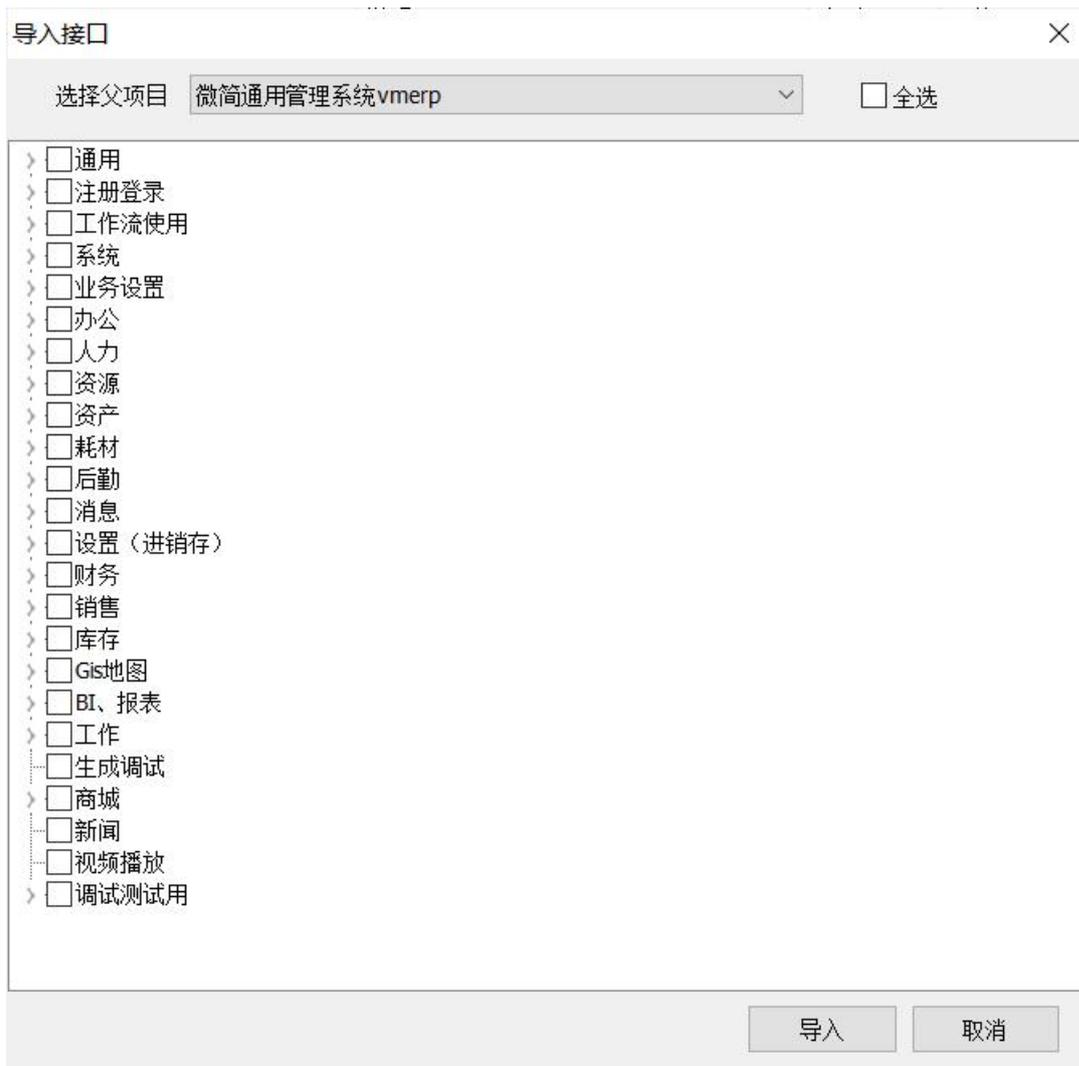
父项目指的是可复用的项目，可以是多个。可以从父项目中导入需求、数据库、接口、字典等内容。

父项目管理添加

* 父项目编号 请输入

确定 关闭

比如：vjproject 中从父项目导入接口。



13.7 文档备份

本系统文档备份指的是成型的文档。常规文档还是使用常规的文档管理工具。

13.8 代码备份

代码备份指阶段性正式发布的完整代码，压缩成一个文件之后上传即可。代码管理还是使用常规的代码工具。

13.9 Bug 管理

系统提供比较简单的 bug 管理。建议使用自己习惯的 bug 管理系统。

14 vjframe 框架介绍

vjframe 是微简自研的 java 前后端分离框架，全部采用原生+函数为主的写法，可读性做到了极致，同时还具备极高的性能。vjframe 是一个动态 web 项目，采用 eclipse 开发。后端采用 servlet 基本的原生架构。前端采用 bootstrap、adminlte、echarts、jqgrid、ztree 等，并做了大量美化。

基本功能：包含了登录、注册、用户管理、角色管理、组管理、权限管理，和一个自研的小型工作流引擎。

vjpage 生成的前端开发代码必须复制到 vjframe 中才能用。vjproject 生成的 java 代码必须复制到 vjframe 中才能用。

全部源码免费开放，可任意修改、扩展。源码、库结构、接口、字典等都完全开放。

注意：vjframe 支持谷歌、搜狗、360、腾讯、火狐、Edge 等浏览器。但 vjframe 框架使用了 ES6 语法，不支持 ie 浏览器。

15 vjframe 框架开发

vjframe 详细的开发文档见：vjproject，笔记，通用教学。

