通用开发框架(JHMPP)架构说明书

1. **适用场景和综述**

系统基于c#.net 4.0, webservice, html5, bootstrap, jquery, redis, winform,mssql2012等相关技术开发。 系统架构，倾向于移动优先，倾向于免前端实现。 大部分常规系统，尤其是行业应用领域，均可使用此系统快速完成业务逻辑实现。

JHMPP的设计，不含具体业务逻辑，目标是尽可能提高开发人员效率，让更多的精力集中在业务逻辑的处理上，方便协作。可以认为，它是介于开发工具和业务逻辑中间的一个工具，一个通用系统架构。

JHMPP对权限控制、登录控制、界面风格、服务接口调用、常用处理方法等进行了规范化的定义，通过这种统一规范以及合理的代码结构，让团队协作效率更高，天然支持海量数据，支持html5标准，支持移动端展示。

1. **优缺点**
2. 支持海量数据存取分页，开发人员无需编写复杂的代码，或计划完全不需要编写代码即可方便的使用此特性。
3. 没有特殊需求的情况下，通过配置可生成全部常见功能，无需编写代码。读取数据库的业务逻辑除外。
4. 支持Redis缓存，或支持使用Redis实现某些特殊功能。
5. 支持分布式部署，支持负载均衡。
6. 对传统webservices技术进行封装，开发人员之间不需要知道调用地址和方法名，无需进行引用，方便更新和修改。
7. 对服务接口进行统一的管理和互相调用授权，同时支持其他各种语言编写的webservices。
8. 支持服务接口之间交叉调用调试，并允许详细日志，方便查找问题节点。
9. 接口调用，支持异步和同步调用。某也业务可以放入队列，提供高并发支持。
10. 自带丰富的常见类库，例如字符换处理，加密处理，图像处理，网络采集等。
11. 一次配置，一套业务逻辑代码，同时服务于html5和手机端访问和winform客户端。
12. 提供统一的数据库处理类库，丰富的类库方法适用于不同的场景。
13. 最简结构下，需要建立三个iis站点，分别处理接口管理，前端展示，业务逻辑。
14. 各站点均可根据业务逻辑拆分并部署到集群。
15. 前后端完全分开，前端禁止直接连接数据库。
16. 只能基于windows server服务器部署，且不支持普通FTP虚拟空间，需要完全可掌控的主机支持。
17. 无法进行界面风格自定义，无法进行控件风格定义，但界面布局可以根据需要自定义。
18. **设计思路描述**

大部分系统，主要组成部分可以大致分为表单和列表， 表单包括添加删除修改查看打印等， 列表包括数据表格、查询、饼状图、曲线图、柱状图等。

大部分情况下，列表功能的实现无需编写任何代码，仅通过配置即可完成。 常规表单代码量极少，编写代码只为处理业务逻辑和特殊的交互。

整个系统的设计上，将界面、业务逻辑、数据库、页面关系四个方面完全分开考虑，通过配置和代码规则，最终整合在一起。 因此可以灵活进行拼凑。不自动生成数据库结构，不自动生成添加删除修改代码， 添加和修改之间并无直接逻辑联系，可通过规范整合，也可以分别视为独立功能。

同时兼顾了代码快速开发，功能模块灵活多变的需要。在灵活和固化中间找到了一个相对合理的平衡点。

1. 登录控制和权限控制

设定了固定数量的基础权限（位运算权值）。 分为固定的五大类： 后台菜单权限、前台导航权限、全局独立权限、特殊权限、备用权限。 每种类型的权限基础权值上限是6500条。 根据需要进行灵活运用。完全可以涵盖任何系统的全部权限需要。

固定权值的中文名称只是一个内部标记，不在验证权限的时候使用，真正使用的是权限基础值。有个基础权限枚举数据表。 在来看用户权限，当某个用户需要分配某些权限的时候，把需要的权值进行位运算得到一个数字，这个数字就是用户的最终权限（当然分为了五种不同情况的类型）。

验证权限，只需要对照权限枚举权值，和用户最终权限就可以了。

权限组的实现同理，实际上就是把提前设定的权限最终值分配给力一些组名，然后把组名关联上用户就行。 得到最终权限只需要简单的进行位运算。

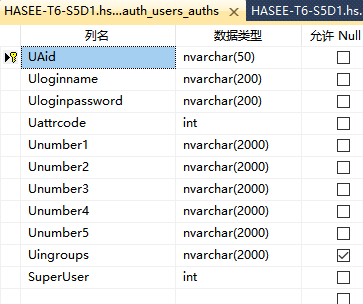
场景：

对于登录来说，调用已固化可修改的登录接口，可进行验证并返回特定格式的用户信息。

对于菜单权限，可完全通过配置，做出任何规则的菜单控制。

对于功能按钮权限，需要自己调用代码进行验证，提供规范的验证接口。

用户权限记录在auth\_users\_auths表中，



Uaid代表用户唯一识别号。 Uloginname和Uloginpassword代表账号和密码。Uattrcode代表离职或冻结。 -1为正常，1为离职，2为冻结。

Unumber1到Unumber5分别对应后台菜单权限、前台导航权限、全局独立权限、特殊权限、备用权限。 这些字段记录权限枚举表中的权限标识异或运算后的一个值。通过这个值，与需要判定的权限枚举值进行或运算，即可判定是否有权限。

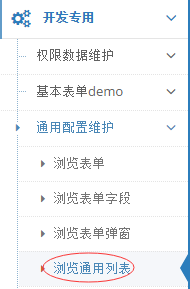
Uingroups记录用户隶属的权限组。多个权限组用逗号隔开。

SuperUser标记是否为超级管理员，值为1，则具备全部权限。

auth\_group表记录了五中权限分别对应的权限组最终权值。

1. **列表配置详述**
2. **建立一个列表的完整步骤**

进入菜单：



新增或克隆一个列表。 注意，sys\_开头的配置不要乱动和修改，这是系统级别的固定配置，删除了系统就不能用了。

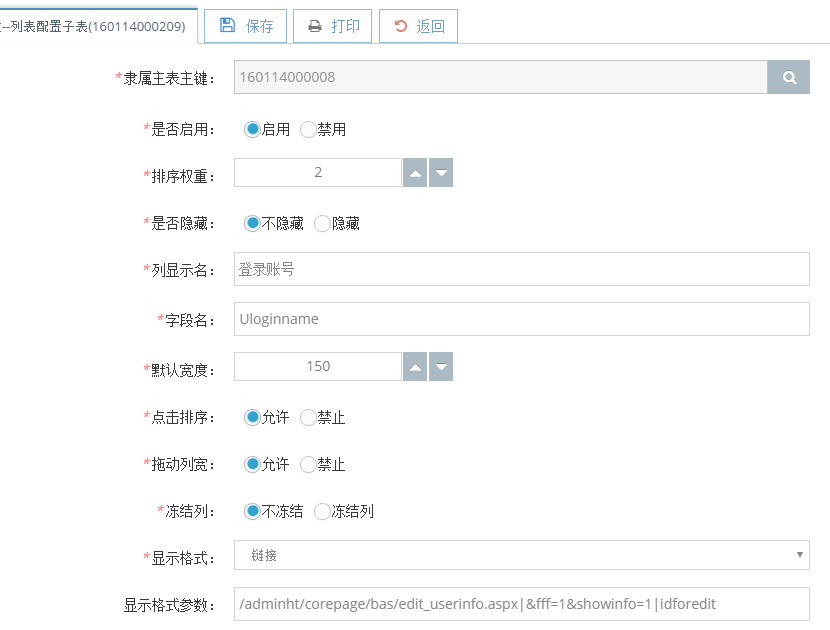


建立后，点击这个列表主键进入详细配置，蓝色区域一般无需调整。

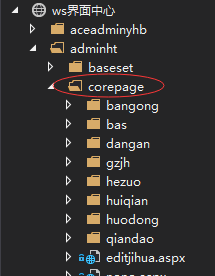
 

列表内要显示的字段在配置界面以子表形式体现，可以直接进行配置。为了方便管理字段，也可以点击列表名称，再点击字段主键，进入独立编辑界面，对该列表中某个字段的显示进行配置。



配置好列表列表参数后，这个列表基本就建好了，下一步是在界面上显示出来。

打开解决方案，在这个目录下新建一个目录，



并在新建目录下通过复制（/adminht/linshi/demo\_dblist\_chart.aspx） 新增一个aspx模板文件， 命名规则一般可以是list\_页面名.aspx。 注意如果是复制，要新的文件名和类名对应。 这个页面代码很少，通常需要修改列表识别号。 也就是上一步建立的列表主表主键的值。让这个页面知道，这个页面用来显示这个主键下的配置。



至此，一个列表就建立好了。下面，需要将这个页面配置到菜单中显示，或者根据需要也可以不配置到菜单里面，通过某个页面中的某个功能连接也可以直接访问。





选中或新建菜单，在这个菜单下增加刚才要建立的页面的,



然后配置这个新的菜单



其中，节点名，就是菜单显示的中文名。

链接地址，就是刚才新建列表页面的绝对地址。

Tip提示和打开方式暂时没用。

图标只对一级菜单有效，符合 fontawesome 开源图标库标准的名称即可，例如fa-cogs，这个图标名称其实是样式表名称，如果需要可以自己定义样式，一般就从fontawesome图标库找一个就行。 （网址http://fontawesome.io/icons/），有些最新图标可以不支持，需要自己更新图标库本地文件，大部分都支持。

高亮跟踪，一般留空，是指如果这个新建列表中某个字段有自定义的链接，点击链接跳转后，需要保持左侧菜单定位，所以，要定义可能会跳转到的页面的地址标志。这个规则比较复杂。

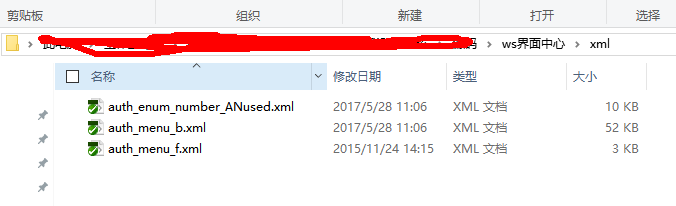
规则一. 左侧菜单默认定位到与“链接地址‘’匹配的节点名上。

规则二. 如果留空，用户点击此列表页面内的链接跳转了，左侧导航栏定位默认保持不变，但只在第一次跳转有效，跳转后再跳转左侧菜单定位就会找不到了。 但前提是被跳转页面标志没有被配置到其他菜单中才行。

规则三. 如果不留空，则可以让多次跳转菜单依然可以准确定位，一般只需要定义被跳转页面的绝对链接即可，不需要带参数。多个可能跳转的地址，用★符号隔开。 如果是同一个被跳转页面在其他菜单也被设为高亮标志了，要用不同参数区分。使用正则表达式匹配。例如

配置好菜单以后，点击保存修改。 菜单就保存到了数据库，但并不生效显示，需要点击

来生成菜单，才能显示。 注意，如果修改了权限枚举，一样也需要点击这里的生成xml。因为菜单和菜单权限配置每次加载页面并不读取数据库，而是读取xml。Xml生成后的地址在这里



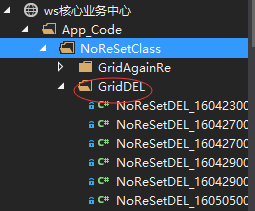
生成后，即可在菜单中看到建立好的列表页面全部功能了。

1. **列表配置字段常用规范**

|  |  |
| --- | --- |
| 列表名称 | 报表的名字，实际上界面不显示。用于方便查看的。通常可以使用菜单的名字，以方便查阅。而且不要有重复。 |
| 获取数据接口 | 不需要修改，系统标准规范名称。 |
| 删除数据接口 | 一般情况保持“框架免代理通用接口删”名称不变 |
| 删除按钮 | 列表中是否显示批量删除按钮，这个开关只控制是否显示。在业务中心按照规范新增.cs文件进行业务逻辑处理。 |
| 导出按钮 | 如果显示，则自动支持导出数据，无需编写代码，可自动导出用户动态改变条件后的全部数据。如果取值列的某列在显示在了列表中，并且实际上表或试图中没有这一列，则应该把列名称前面增加一个小数点符合(.),让其不导出，否则自动导出解析会出错。 |
| 新增按钮 | 配合新增按钮链接使用。 输入一个地址，用户点击新增按钮时，将跳转到这个地址。 |
| 自定义按钮 | 这是特殊处理规范，定义图标、名称、处理方法名标志。在业务中心按照规范新增.cs文件进行业务逻辑处理。 |
| 列宽形态 | 建议选择 展开列宽，字段会按照设置的宽度显示，可能出现横向滚动条。 更好的使用移动端。 如果是自适应，则按照宽度设定比例显示，不会出现滚动条，但手机端就会挤到一起。 |
| 复合表头 | 复合表头的设置js代码，具体规则详见jqgrid文档(有错位bug，还没处理好)  //复合表头  jQuery(grid\_selector).jqGrid('setGroupHeaders', {  useColSpanStyle : true,  groupHeaders : [  {  startColumnName: '唯一编号',  numberOfColumns : 2,  titleText : '一级累计1'  },  {  startColumnName: '省份',  numberOfColumns : 2,  titleText: '一级累计2'  }]  }); |
| 取值字段 | 要获取的字段，全取值就是星号。 不管取出几个字段， 字段配置表里面配置的显示列将从这里取值，因此必须确保字段配置表里每个字段是这里设置字段的子集。 同时确保字段名跟表或试图中里一致。  例如  SID,Sname ,Spassword ,Ssex ,Scity ,Sdiqu ,Sint ,Sdecimal ,Convert(varchar(10),Stime,120) as Stime |
| 取值表 | 数据来源的表名或者视图名称，不用视图直接写链表代码也行。  链表例子  (select xxx,yyy,ccc from A join B on A.id=B.id ) as tab1 |
| 取值条件 | 取数据的默认条件。附加一些用户无法控制的固定条件。 没有固定条件就留空。特殊通配符{idforedit}，可以自动获取当前页面url中的idforedit参数值。也就是目前正在编辑的表id  例如  type=1 and open='2' |
| 取值排序 | 取数据的默认排序规则代码。 排序不能留空，必须有个默认排序  例如  CreateTime desc,SID asc |
| 取值主键 | 主键或联合主键。这个必须存在，有时候select出来的数据可能没有主键，这时必须拼接一个联合主键。否则分页混乱。本来就存在主键的，就写上主键字段名就行了。  如果是拼接临时的联合主键，使用加号连接，可以连接好几个：  name+type+class |
| 分页数 | 分页每页数量的配置值，可以有多个。第一个是默认的，其他的是供用户可选的。用逗号分开。 至少要设置一个。不能留空。 这里不允许编辑，如果需要修改，则需要在数据库直接修改。  例如:  20,25,50 |
| 快捷查询 | 是否启用快捷查询配置。如果开启，列表顶部最多三个快速查询。 高级查询自动生成，一直存在，直接列表中显示的所有字段的查询。 |
| 快查标题 | 快捷搜索 的显示名称，不带冒号。 如果是下拉框类型，则可以配置下拉框内容。 固定内容格式：性别\*男,女  固定内容值和现实分开格式：性别\*男|0,女|1  从枚举表动态提取单号内容格式： 性别\*[sqlone] select 内容带逗号的字段 from 配置表 where 条件  从普通表中取出多行内容格式：性别\*[sql]select 内部值,显示文本 from 表 where 条件 |
| 快查控件名 | 快捷搜索的控件标示id，同时也必须是查询字段的列名 |
| 快查类型 | 目前只支持，输入框，时间段，下拉框 |
|  |  |
| 列表配置的字表中对于显示的相关控制规范 | |
| 隶属主表主键 | 对应列表列表基本配置表的主键。 就是FUP\_FormsList表的主键。 |
| 是否启用 | 这个字段是否启用。 |
| 排序权重 | 列默认排序。这个影响列显示的顺序，数值越小越往前排。允许小数。通常先设定整数，以方便后续用小数插入列的顺序 |
| 是否隐藏 | 如果隐藏，虽然用户看不到，但js程序可以取值。 |
| 列显示名 | 各列的显示名称。 可以是中文，但同一个列表内不能有重名，并且不能带有逗号冒号。也不要用奇怪的字符。最好也别用中横线，不然导出会失败。 |
| 字段名 | 必须对应数据库字段名一致，必须是英文，别有奇怪字符。如果DID\_formatter是“自定义”，就不用非得一致了。 |
| 默认宽度 | 默认的列显示宽度，自适应模式下变为比例权重。 |
| 点击排序 | 表示是否允许点击排序。 true为可以排序，false为不能排序。 不是所有列都适合排序，如果DID\_formatter是“自定义”，就一定要搞成不允许排序。 |
| 拖动列宽 | 是否允许拖动列宽改变宽度，false为允许拖动，ture为固定不变不能拖动宽度。 比如时间，自定义列等具备固定长度数据的类型，建议设置为不允许拖动。 |
| 冻结列 | 冻结列不知道为啥不能用，有bug。但winform端可以正常过解析。 |
| 显示格式 | 数据显示格式。 包括 字符串、链接、整数、小数、日期时间、仅日期、自定义 。 必须使用规定的这几个字符串。 |
| 显示格式参数 | 数据格式的特殊参数， 如果是链接就代表链接参数，如果是自定义就填写具体jqgrid支持的代码，可以实现其他复杂的功能。 如果DID\_formatter不是链接也不是自定义，此列没用,可以留空。  链接参数例子：  /xxx.aspx|&spsp=xxxxx|linkid或/xxx.aspx|&spsp=xxxxx|linkid|\_blank  链接的结果是："/xxx.aspx?linkid=150604000045&spsp=xxxxx"  如果想要点击链接进入编界面，则需要符合表单规范，详见表单规范的部分。  自定义代码例子：  formatter: function (cellvalue, options, rowObject) { return "<img height='20' width='20' src='/" + cellvalue + "' />" }  自定义代码另一个例子：  formatter: function (cellvalue, options, rowObject) { return "<a href=\"demo\_info.aspx?idforedit=" + $(rowObject).find('SID').text() + "\" target='\_blank'><button type='button' class='btn btn-minier btn-info'>详情</button></a> " } |

1. **对于列表的特殊处理规范**
   1. **编写删除业务逻辑**

在业务中心，这个目录下，添加一个.cs文件



文件命名规则 ：NoReSetDEL\_列表配置主键.cs

用户点击批量删除按钮后，会自动调NRS\_DEL 方法，参见NoReSetDEL\_sys\_demo01.cs的例子。parameter\_forUI参数，可以取到传过来的待删除id和当前操作人等信息。根据业务逻辑，自行编写代码对多个表进行操作。

* 1. **编写自定义操作按钮业务逻辑**

在 自定义按钮配置上，如果配置了自定义按钮，则允许进行自定义批量处理，例如审核、复制、作废等等。用逗号隔开配置可以添加多个按钮.

格式例如： 克隆|fa-files-o blue|kelong,复制|fa-copy red|fuzhi

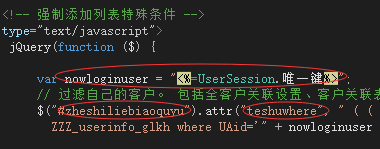
其中，|用来分开同一个按钮的不同参数。 第一个参数是显示的中文名，第二个参数是按钮图标。 第三个参数是处理方法的标志。

当用户点击按钮时，会自动调用 NoReSetDEL\_列表配置主键.cs里面的NRS\_ZDY\_方法标志，这个方法。 类似删除的业务逻辑定义。同样可以自行处理。详见NoReSetDEL\_sys\_demo01.cs的例子

* 1. **在前端对列表附加动态特殊条件**

很多时候，我们需要在前台动态增加一些特殊条件，例如传入并限制当前用户相关的数据，例如界面上增加三个快捷查询以外的其他条件。总之如果要对查询条件在配置之前进行动态处理。需要增加一段js代码在前端aspx页面中。

例如：



其中框起来的红色区是固定的。 通过UserSession来拿到当前用户的Uaid，再根据需要，拼出where语句，并将动态新增的条件语句，放入$("#zheshiliebiaoquyu")的teshuwhere属性上。给属性赋值后，点击快速查询或高级查询或刷新列表，会自动带入新的条件。灵活运用这个属性，可以实现很多特殊功能。

* 1. **给快捷查询增加默认值**

一般通过js进行。F12可以看到html源代码。 根据需要随便搞。对不同控件的赋值代码，在pucu/wuc\_script.ascx文件中都能找到。

强制给日期类型控件赋值是这个：



* 1. **某条数据上的功能链接的详细规范**

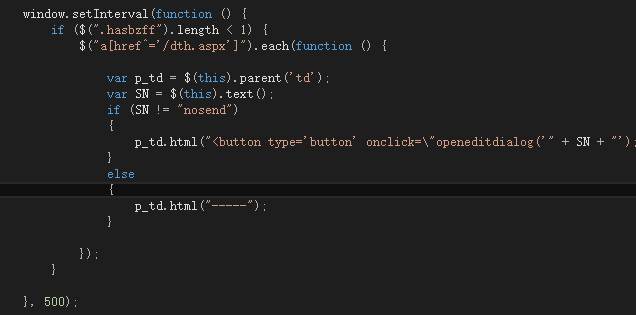
这个一般通过配置就可以完成。规范是，列选择链接类型，在显示格式参数中，增加例如这样的配置：

/adminht/corepage/huodong/edit\_cysw.aspx|&fff=1&showinfo=2|idforedit|\_blank

其中，fff=1代表这是要进入编辑页面，没有fff参数就是新增。Showinfo=1代表进入编辑前先进入查看界面，然后点击自动生成的编辑按钮才进入编辑。Showinfo=2代表只进入查看界面，无法进行编辑。 Idforedit代表把这行数据的主键值赋值给这个参数。 最后一个参数可选，代表是否从新页面打开连接。

* 1. **批量动态处理列表**

有时候，需要对列表展示进行特殊的处理，比如统一把某列加上按钮、弹窗，或其他可能的特殊处理。此时可以这么干：



总之，就是可以利用jquery代码对已经生成的页面进行各种自定义处理以达到特殊交互目的。

有时候需要给列表增加其他特殊区域。用这样的方法可以动态调用已经存在的处理类库。



* 1. **Ajax获取某些数据并处理**

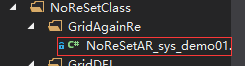
框架中标准的ajax处理代码：



另外，其实每个页面都有 一些公共方法可以用，比如getUrlParam是获取当前页面参数。 bootbox.alert("发送短信失败，"+msg);是弹窗。

* 1. **让列表中的当前页进行二次处理**

某些情况下统计分析数据运算比较复杂，如果使用存储过程或视图，会占用数据库资源，并且调试不便，执行效率低。 可以使用二次处理类进行加工，由于二次处理类加工的是当前分页的有限数据，因此处理起来很灵活，基本不存在效率问题。



这个目录下，以 NoReSetAR\_ 开头的文件，后面跟列表识别号，可以对数据进行二次处理，必须有一个固定方法名NRS\_AR。

例如原来的数据有一列复杂的列“200天内最大值”，配置的时，不需要让语句取这个 值，只需要预留这个字段。 然后通过二次处理，遍历当前分页的N条数据，把这个“200天内最大值”的真正值再补充进去。就可以高效的显示了。极大减少了资源占用。不过这种方式，会导致这个字段不能自动导出了。只适合查看。

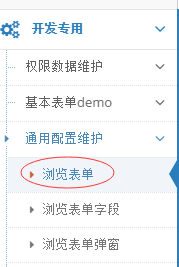
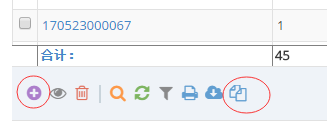
还有一种清空，就是不对数据进行二次数据，而是对查询条件进行自定的二次处理。有时候单纯where后面的条件，难以满足查询要求。这时候可以通过NoReSetAR\_tsthcs\_开头的类，对传过来的条件进行二次处理。

* 1. **让列表上方显示饼状图或曲线图或柱状图**

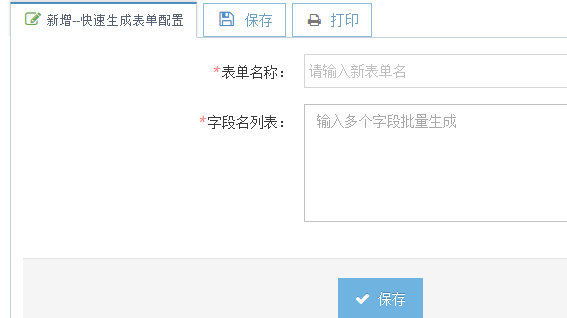
如果需要显示可视化的报表，则可以利用数据二次处理类库，进行处理。使用简单的一句话，能把原来回传的数据集，转化成图形报表能识别的格式。这样报表会自动显示在界面上，而不需要编写报表界面代码。可以非常方便快速的做出图形报表。

详见NoReSetAR\_sys\_demo01.cs中的例子。主要通过jsontodatatable.re\_chart这个 函数处理。

1. 表单配置详述
2. 表单的建立过程

这里新建或克隆一个表单。如果是新建表单，只需要输入表单名和字段名即可。字段名不需要与数据库一致，但建议还是一致。因为获取数据给表单赋值时，根据取值sql语句中的字段名或某些特殊规范，会给表单自动赋值。



建立表单后，点击表单配置主键，会进入编辑界面。

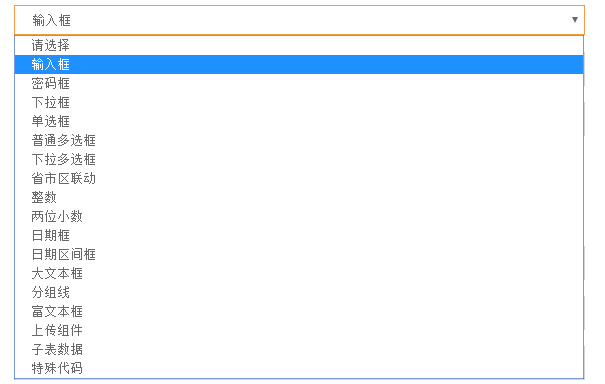
 

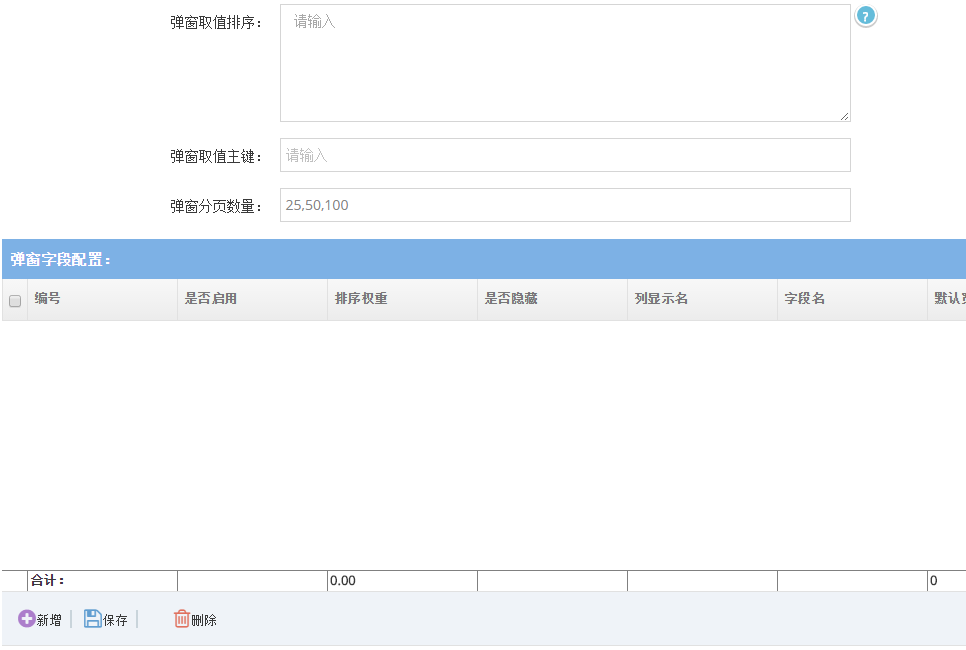
这个界面可以对表单字段进行编辑，99%的常规表单可以通过配置生成，无需编写代码。

表单名称会显示在最终界面上。

新增数据接口、编辑数据接口、编辑前获取数据接口，通常不需要配置，使用默认值即可。特殊表单可以自己配置自定义接口。

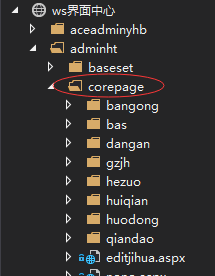
表单上显示的字段，在表单字段配置子表中进行维护。为了方便维护，也可以在表单列表中点击表单标题，进入字段列表，在点击字段唯一编号，即可单独维护这一个字段。



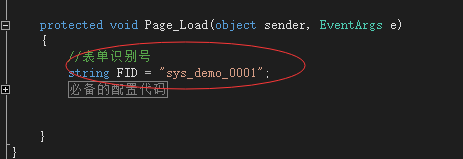
  

配置好表单参数后，这个表单基本就建好了，下一步是在界面上显示出来。

打开解决方案，在这个目录下新建一个目录，



并在新建目录下通过复制（/adminht/linshi/demo\_pz\_edit.aspx） 新增一个aspx模板文件， 命名规则一般可以是edit\_页面名.aspx。 注意如果是复制，要新的文件名和类名对应。 这个页面代码很少，通常需要修改表单识别号。 也就是上一步建立的表单主表主键的值。让这个页面知道，这个页面用来显示这个主键下的配置。



至此，一个表单就建立好了。下面，需要将这个页面配置到菜单中显示，或者根据需要也可以不配置到菜单里面，通过某个页面中的某个功能连接也可以直接访问。具体方式在列表建立中已有说明。

1. 表单配置详细规范

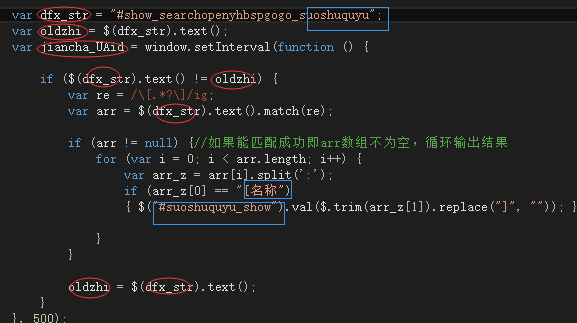
|  |  |
| --- | --- |
| 隶属主表主键 | 隶属的表单基本设置主表主键。也就是FUP\_FormsMainInfo的主键。 |
| 是否启用 | 即便不是启用状态，字表分录中的弹窗也一样可以使用这个配置 |
| 控件类型 | 输入框、密码框、下拉框、单选框、普通多选框、下拉多选框、省市区联动、整数、两位小数、日期框、日期区间框、大文本框、分组线、富文本框、上传组件、子表数据 。 必须使用这些规定的字符串。 |
| 控件名 | 一般情况跟数据库表字段相同即可，会作用控件名使用，所以只能使用英文或下划线。 不一定非要跟数据库字段相同，接口里面通过这个名字获取提交上来的表单内容。注意某些特殊控件会自动把名称变形。 |
| 控件标题 | 显示在前面的文字标题，不要加冒号。 |
| 最小值 | 对于文本框来说是允许的字符数下限。对于数字来说是允许的最小值 |
| 最大值 | 对于文本框来说是允许的字符数上限。对于数字来说是允许的最大值 |
| 默认值 | 设置输入框或数字框的初始默认值，只对输入框和数字类型控件有效 |
| 内部提示 | 内部灰色提示。 只对文本输入框有效 |
| 悬浮提示 | 外部黑色悬浮提示。 只对文本框有效。数字或其他类型控件无效。 |
| 必填项 | 0代表非必填项、1代表必填项。影响前台js验证。 |
| 必填提示 | 必填项目时，没有填写项目的错误提示文字。若留空会按标题自动使用默认文字。 |
| 排序权重 | 优先级索引，越小的值，越往前排，允许小数值 |
| 静态内容 | 下拉框的静态内容，用逗号隔开。 内部值和显示值用|隔开。 以[sql]或[sqlone]开头可以动态取数据.  固定内容格式：性别\*男,女  固定内容值和现实分开格式：性别\*男|0,女|1  从枚举表动态提取单号内容格式： 性别\*[sqlone] select 内容带逗号的字段 from 配置表 where 条件  从普通表中取出多行内容格式：性别\*[sql]select 内部值,显示文本 from 表 where 条件 |
| 输入掩码 | 简单掩码控制。例如999-9999。留空就是不控制格式。(jquery maskedinput) |
| 是否只读 | 若要只读。 设置成readonly。 若要允许编辑留空即可。 |
| 启用弹窗 | 0代表没有弹窗按钮，1代表附加弹窗选择按钮。弹窗按钮只对文本输入框有效。 |
| 弹窗引入值设置 | 引入值和显示值的配置，冒号前是引入值，冒号后是引入显示值。多个值用逗号分开。 这里的配置必须使用FUP\_FormsSubDialog表里面的DID\_showname。也就是弹窗配置子表里面的字段显示名。 引入值将被带入输入框，显示值显示在输入框下方。  例如：  账号:地区,性别 |
| 弹窗自适应列宽 | 是否自适应列宽。 false代表依据字段宽度设定展开，true代表不出现滚动条并且字段的宽度设定的含义变成了比例权重。。建议展开列宽。 |
| 弹窗复合表头 | 复合表头的设置js代码，具体规则详见jqgrid文档(有错位bug，还没处理好) |
| 弹窗取值字段 | 要获取的字段，全取值就是星号。 不管取出几个字段， 字段配置表里面配置的显示列将从这里取值，因此必须确保字段配置表里每个字段是这里设置字段的子集。 同时确保字段名跟数据库里一致。  例如  SID,Sname ,Spassword ,Ssex ,Scity ,Sdiqu ,Sint ,Sdecimal ,Convert(varchar(10),Stime,120) as Stime |
| 弹窗取值表 | 数据来源的表名或者视图名称，不用视图直接写链表代码也行。  链表例子  (select xxx,yyy,ccc from A join B on A.id=B.id ) as tab1 |
| 弹窗取值条件 | 取数据的默认条件。附加一些用户无法控制的固定条件。 没有固定条件就留空。特殊通配符{idforedit}，可以自动获取当前页面url中的idforedit参数值。也就是目前正在编辑的表id  例如  type=1 and open='2' |
| 弹窗取值排序 | 取数据的默认排序规则代码。 排序不能留空，必须有个默认排序  例如  CreateTime desc,SID asc |
| 弹窗取值主键 | 主键或联合主键。这个必须存在，有时候select出来的数据可能没有主键，这时必须拼接一个联合主键。否则分页混乱。本来就存在主键的，就写上主键字段名就行了。  如果是拼接临时的联合主键，使用加号连接，可以连接好几个：  name+type+class |
| 弹窗分页数量 | 分页每页数量的配置值，可以有多个。第一个是默认的，其他的是供用户可选的。用逗号分开。 至少要设置一个。不能留空。 这里不允许编辑，如果需要修改，则需要在数据库直接修改。  例如:  20,25,50  如果字段类型是子表数据类型，则这里不能使用逗号，应该直接写入一个比较大的数，因为子表不能分页。 |
| 弹窗字段配置（也用于字表字段配置），部分字段含义和标准与列表相同，多出来的配置如下： | |
| 显示格式 | 数据显示格式。 包括 字符串、链接、整数、小数、日期时间、仅日期、自定义 。 必须使用规定的这几个字符串。 |
| 显示格式参数 | 数据格式的特殊参数， 如果是链接就代表链接参数，如果是自定义就填写具体jqgrid支持的代码，可以实现其他复杂的功能。 如果DID\_formatter不是链接也不是自定义，此列没用,可以留空。 |
| 是否允许编辑 | 编辑模式专用。是否允许编辑。 |
| 编辑是否必填 | 编辑模式专用。是否必填项。 |
| 编辑控件类型 | 编辑模式专用。指定编辑控件类型，只对字符串类型有效。包括：文本框、下拉框、 弹窗选择 |
| 编辑特殊参数 | 编辑模式专用。特殊设置，比如下拉框的选项等。 以[sql]或[sqlone]开头可以动态取数据 |

1. 弹窗类型的输入框

如果输入需要使用弹窗，需要将配置设置为启用弹窗。并且配置好弹窗引入值。

弹窗内的字段显示和字表类型控件字段显示共用一个配置子表。

提交后台后，只能获取到带入输入框的值，如果要获取弹窗带入的其他值，需要增加一些字段，并手动编写js将值放入其他字段中。规范如下：



这段代码中，如果一个页面有多个弹窗值要带入并赋值到其他控件，需要修改红色变量名为不同的变量名。通常是复制到记事本中，然后批量替换。

蓝色部分第一个地方，是控件名，也要修改。

蓝色部分第二个地方，是要把那个字段带入。注意第一个[符号都要带着。

蓝色部分第三个地方，是带入到哪个控件。复制这段if语句，可以带入多个不同的控件中。或自行根据需要进行其他处理。

1. 子表分录类型的控件

字表分类类型，实际跟弹窗共用一个配置子表，只是引入值等配置已不作用。

另外需要注意，分页数量不能使用逗号隔开了，强制写一个5000或更大的值即可。

1. 子表分录中不允许编辑的特殊处理

虽然子表配置可以配置某个字段是否允许编辑，但由于某些情况，需要手动更改，也就是，允许编辑，但输入控件不可用。 主要体现在，如果子表里面也使用了弹窗，并且要带入数据的时候，就要进行这种特殊处理。

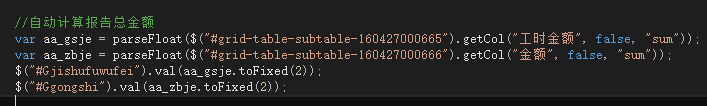
这种特殊处理需要放在window.setInterval中定时制定。500毫秒。

例如



红色区域的编号，是子表字段的唯一编号。思路是，先全部搞成制度，然后开放允许编辑的字段。

还有一种特殊处理，就是需要将子表里面某个字段的合计值计算出来，放到某个其他控件中，例如：



1. 子表分录中的弹窗配置
2. 处理表单提交业务逻辑
3. 表单常用场景规范
   1. 区分新增、编辑、查询模式

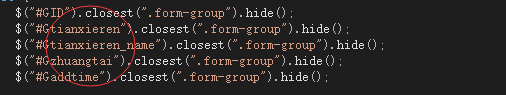
通过 if (getUrlParam("fff") == "1")， 可以根据传入的页面参数，判定表单类型。

fff=1是编辑状态。fff不设置是新增。 showinfo=1是编辑前先显示，点击按钮再编辑。 showinfo=2是仅显示。无法编辑。

也可以通过 getUrlParam获取其他自己传入的特殊参数。用于特殊处理。

参数fff和showinfo是固定参数，根据这两个参数的不同，自动生成的界面上某些按钮会不同。

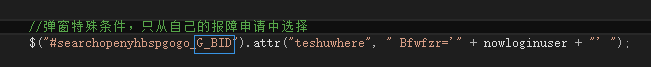
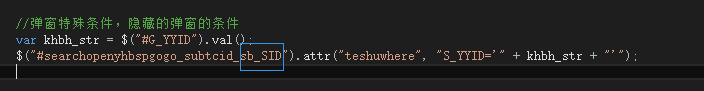
* 1. 强制临时在前端隐藏某个字段



红色是要隐藏的 字段名。

* 1. 给弹窗或子表中的弹窗增加动态附加条件

某些情况，弹窗内容不是固定的，是根据当前用户强制过滤的，或根据界面中其他某个字段的值动态过滤的，需要强制添加条件。代码如下：



注意，带subtcid\_的是子表里面的弹窗。不带的是普通控件弹窗。蓝色区域要修改成这个弹窗的控件名。条件根据需要自己拼接。

* 1. 脱离配置彻底自定义表单

有时候表达配置只是为了带入菜单和控件，某些业务功能可能不能用表单生成，例如地图显示，特殊的布局结构等等。此时需要定义一个临时菜单，但并不使用里面的字段配置，这是有两种可选方案。界面稍有不同。

方案1 是，



把主区域设置成不可见。然后自己在wuc\_content下方直接写html5的代码来画界面。

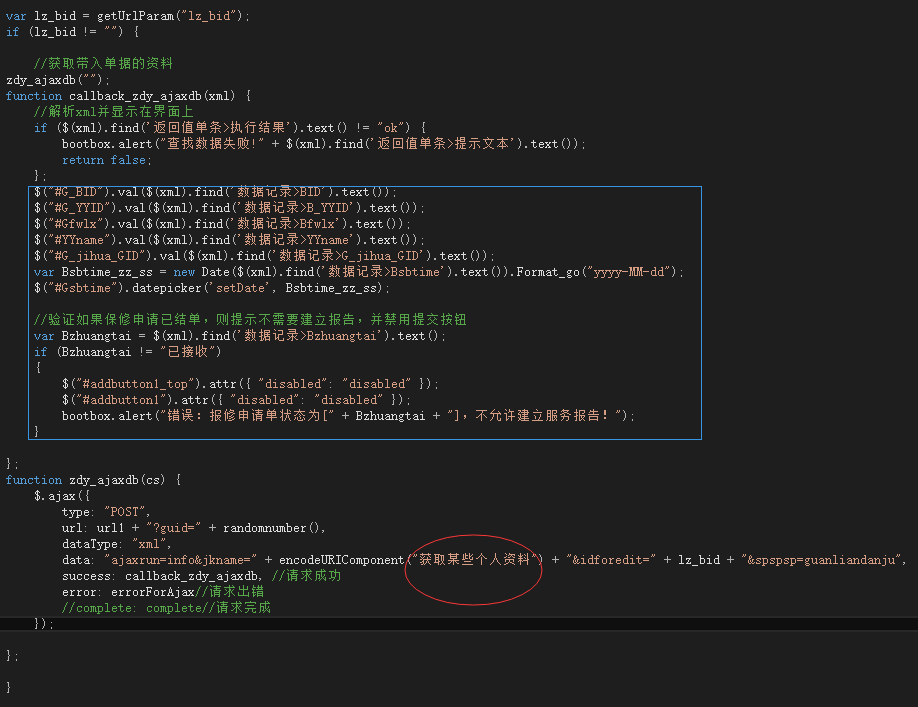
方案2是，把某个控件类型设置成特殊代码，在静态内容中填写自定义控件.ascx的地址。



两种方案各有利弊，根据不同的情况选择。

* 1. 通过ajax获取某些数据并显示在界面上

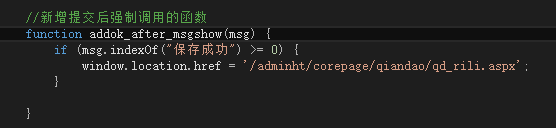
有时候，页面加载后，需要额外通过ajax获取某些数据，以便调整界面或给用户提示。此时允许直接调用某个接口。例如

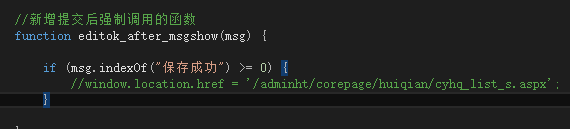


其中，蓝色区域市获取到数据后的处理。 根据需要自己修改。 红色部分是要调用的接口名称。

* 1. 提交后跳转和提示的处理

一般情况下。不需要使用这个特殊代码。若提交后没有按格式返回主键，则会停留在当前提交页面。若回传了主键，则自动进入查看界面，同时出现编辑按钮。 有时，提交后，我们需要强制跳转到某个其他页面，这时候需要添加以下代码：

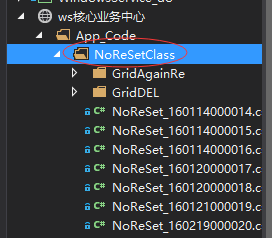




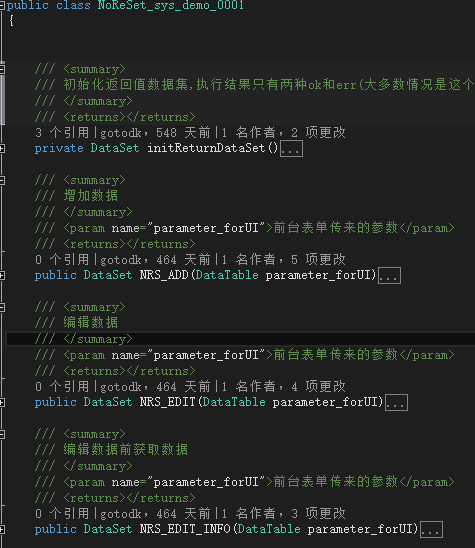
这两个函数不能放到jQuery(function ($) {里面，要放到外面，提交后会自动尝试调用这两个函数。 没有就不会调用。一个是增加的才调用，一个是编辑的才调用。

* 1. 处理表单提交后的业务逻辑

表单业务逻辑在这个目录下添加新文件进行处理。格式是NoReSet\_表单配置主键.cs



参见NoReSet\_sys\_demo\_0001.cs中的例子。



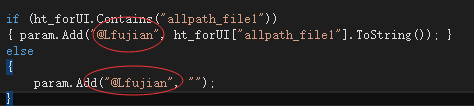
编写业务逻辑需要遵循返回值规范。 见例子。 一个完整的表单，需要实现这四个固定方法名。 第一个不用修改。

第二个NRS\_ADD用于处理新增表单数据。 第三个NRS\_EDIT用于处理编辑表单数据。第三个NRS\_EDIT\_INFO用于在编辑前获取原有数据并显示在界面上。

这里有些特别的注意事项。

1. 获取子表数据时，用的不是空间名，而是显示名。记得修改子表控件唯一编号。
2. 省市区列表、上传控件、日期区间控件、下拉框多选控和单选框取值比较特殊。

对于上传控件来说。



红色是空间名，必须判断是否存在。 Allpath\_file1是固定的名字。 一个表单目前只允许一个上传控件。

日期控件会自动提交上来两个值，分别是，日期段控件名1和日期段控件名2

省市区列表会带有固定前缀。蓝色是固定前缀，红色才是控件名。



下拉框多选控单选框，如果是非必填项，必须进行是否存在的验证。

1. 获取表单新的主键有固定类库。



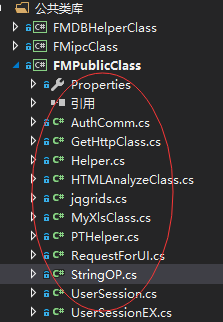
1. 提交上来的内容，自动带有一个yhbsp\_session\_uer\_Uaid。用来取当前操作人。
2. 新增自定义接口和接口调用

详见接口框架文档。

1. 数据库操作规范

详见数据库类库文档。

1. 常用类库



公共类库定义了很多常用的方法。例如对字符串的各种处理，例如截取，加密，转换等等。 或者抓取网络数据，导出表格类库。绝大部分业务系统常见的公共类库基本都有。

1. 常规部署注意事项

详见部署文档