# 东风物流前端开发文档

版本：2.0

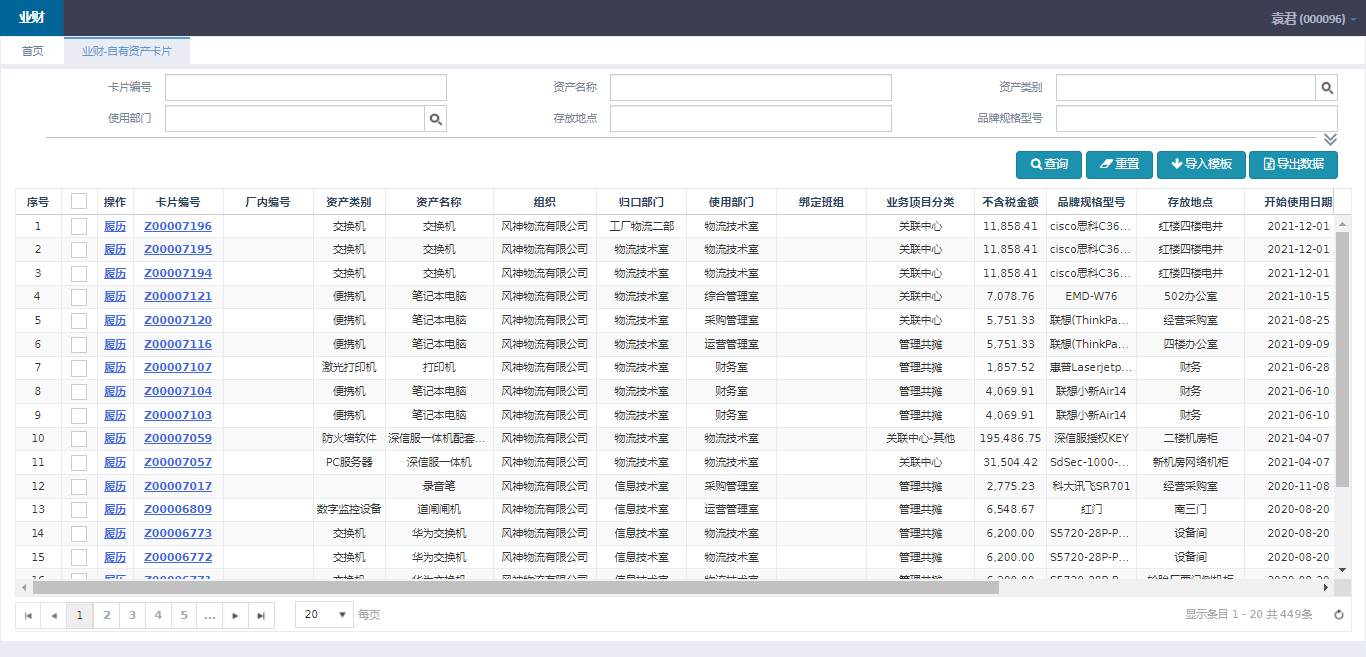
时间：2022年6月17日

## PC端

### 页面样式

#### 创建搜索页

搜索页一般由三部分组成：第一部分为查询条件，第二部分为工具栏，第三部分为结果列表。整个页面都包裹在一个form表单之中。如图：



#### **创建查询条件**

查询条件是由一个form表单组成的，首先创建一个div，并添加class属性为page-content（此属性为集团搜索页的默认样式）

<div class="page-content"></div>

然后，创建一个form表单，并添加class属性为form-horizontal（此属性为Bootstrap 框架自带属性）

<div class="page-content">  
 <form class="form-horizontal">  
 </form>  
</div>

然后，在表单里面创建一个div，包裹查询条件，并且在div里面创建查询条件的div并且添加class属性为form-group（此属性为Bootstrap 框架自带属性），一个div对应一行查询条件，如下代码，有三个class属性为form-group的div，则代表有三行查询条件。

<div class="page-content">  
 <form class="form-horizontal">  
 <div>

<div class="form-group"></div>

<div class="form-group"></div>

<div class="form-group"></div>

</div>  
 </form>  
</div>

然后，在class属性为form-group的div里面用bootstrap进行布局（如果这里不懂，请查看bootstrap框架文档）。并给对应的输入框添加class属性为k-textbox（此属性为kendoUI框架自带属性）并绑定data-bind即可（此处不懂请查看kendoUI开发文档，我写的版本）。完整代码如下：

<div class="page-content">

    <form class="form-horizontal" >

        <div>

            <div class="form-group">

                <div class="col-sm-4">

                    <label class="control-label col-sm-4">卡片编号</label>

                    <div class="col-sm-8">

                        <input type="text" class="k-textbox" data-bind="value:model.cardNo" />

                    </div>

                </div>

                <div class="col-sm-4">

                    <label class="control-label col-sm-4">资产名称</label>

                    <div class="col-sm-8">

                        <input type="text" class="k-textbox" data-bind="value:model.assetName" />

                    </div>

                </div>

                <div class="col-sm-4">

                    <label class="control-label col-sm-4">资产类别</label>

                    <div class="col-sm-8">

                        <input" type="text" data-bind="value:model.assetType" />

                    </div>

                </div>

        </div>

    </form>

</div>

#### **创建隐藏的查询条件（更多查询条件）：**

在原先的的查询条件里的div添加一个more属性，并且添加style="display: none"。代码如下：

<div class="page-content">

      <form class="form-horizontal" id="query-form">

        <!-- 查询条件 -->

        <div>

          <!-- 没有被隐藏的查询条件 -->

          <div class="form-group">

          <!-- 省略里面代码，此处代码与前节相同 -->

          </div>

          <div class="form-group">

          <!-- 省略里面代码，此处代码与前节相同 -->

          </div>

          <!-- 隐藏的查询条件 -->

          <div class="form-group more" style="display: none">

          <!-- 省略里面代码，此处代码与前节相同 -->

          </div>

          <div class="form-group more" style="display: none">

          <!-- 省略里面代码，此处代码与前节相同 -->

          </div>

          <div class="form-group more" style="display: none">

          <!-- 省略里面代码，此处代码与前节相同 -->

          </div>

        </div>

      </form>

    </div>

然后，在form里面创建一个div，添加id属性。然后写一个script标签，使用kendoUI的kendoDivideLine创建一个分割线，target为需要隐藏条件的class属性。

    <div class="page-content">

      <form class="form-horizontal">

        <!-- 查询条件 -->

        <div>

          <!-- 没有被隐藏的查询条件 -->

          <div class="form-group">

          <!-- 省略里面代码，此处代码与前节相同 -->

          </div>

          <div class="form-group">

          <!-- 省略里面代码，此处代码与前节相同 -->

          </div>

          <!-- 隐藏的查询条件 -->

          <div class="form-group more" style="display: none">

          <!-- 省略里面代码，此处代码与前节相同 -->

          </div>

          <div class="form-group more" style="display: none">

          <!-- 省略里面代码，此处代码与前节相同 -->

          </div>

          <div class="form-group more" style="display: none">

          <!-- 省略里面代码，此处代码与前节相同 -->

          </div>

        </div>

        <div id="line"></div>

      </form>

    </div>

    <script type="text/javascript">

      // 添加分隔线

      $("#line").kendoDivideLine({

        target: ".more",

        hasIcon: true

      });

    </script>

#### **创建工具栏**

在form里面创建一个div，添加class为pull-right属性（此属性为Bootstrap 框架自带属性），然后添加span标签，在span标签添加class为btn btn-primar的属性（此属性为kendo框架自带属性）。在span标签里添加图标和文字，span上绑定对应的查询方法。查询和重置已有封装好的方法，只需要添加<@spring.message "hap.query"/>标签，span标签为 data-bind="click:query" 。完整代码如下：

<div class="page-content">

  <form class="form-horizontal" id="query-form">

    <!-- 省略前面查询条件代码 -->

    <!-- 工具栏 -->

<div

class="pull-right"

style="padding-bottom:10px;margin-right: 15px"

>

      <span

class="btn btn-primary"

data-bind="click:query"

type="submit">

        <i class="fa fa-search" style="margin-right:3px;"></i>

        <@spring.message "hap.query"/>

      </span>

      <span

class="btn btn-primary"

type="button"

data-bind="click:reset" >

        <i class="fa fa-eraser" style="margin-right:3px;"></i>

        <@spring.message "hap.reset"/>

      </span>

      <span class="btn btn-primary" onclick="openCreateAdmitTab()">

        <i class="fa fa-play" style="margin-right:3px;"></i>

        申请

      </span>

   </div>

  </form>

</div>

#### **创建结果列表**

先在form里面创建一个div，设定id

<div class="page-content">

  <form class="form-horizontal" id="query-form">

    <!-- 省略前面查询条件代码 -->

    <!-- 结果列表 -->

    <div id="grid"></div>

  </form>

</div>

然后，使用kendo框架的方法。创建一个表格，详细的表格

var grid = $("#grid").kendoGrid({

        columns: [{

          field: "name",// create a column bound to the "name" field

          title: "Name" // set its title to "Name"

        }, {

          field: "age",// create a column bound to the "age" field

          title: "Age" // set its title to "Age"

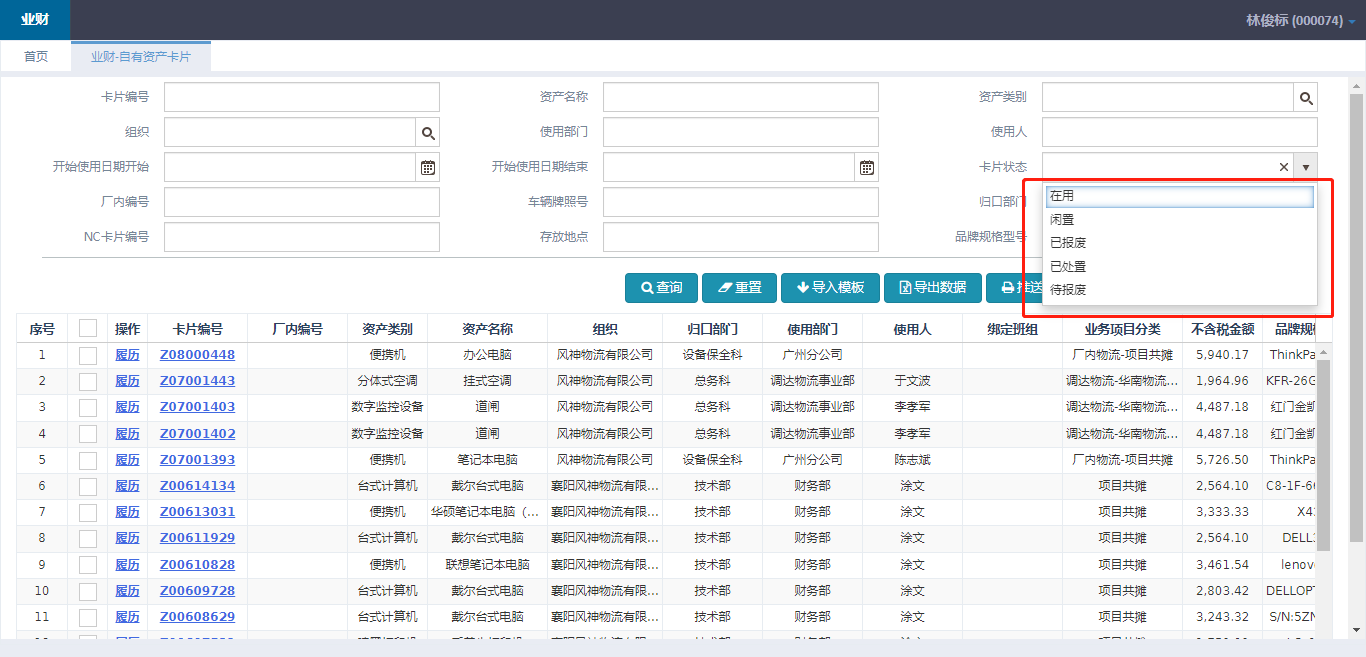
        }],

        dataSource: [{ name: "Jane", age: 30 }, { name: "John", age: 33 }]

      });

### **数据字典配置**

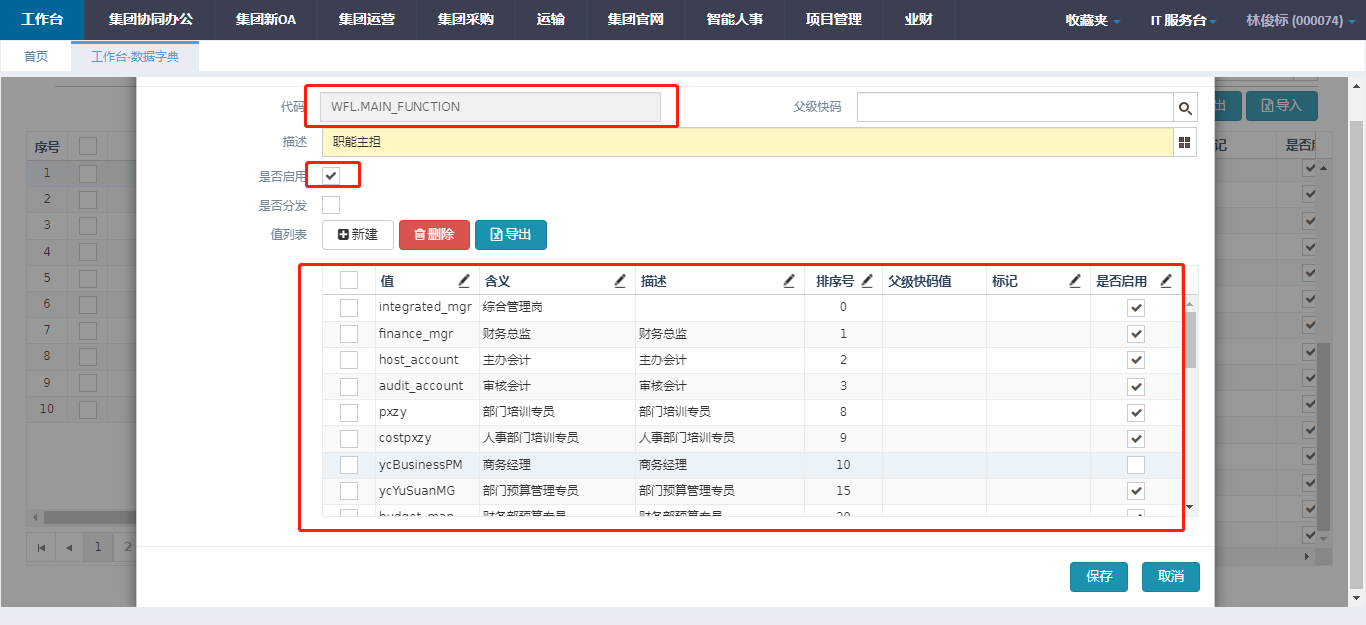
前端的数据字典如下图，需要自己进行配置。配置需要有权限的管理员角色才能进行配置。



进入系统，点击运维管理，点击数据字典，如下图。



点击新建，输入代码和对应的值，如下图。



最后在需要的页面使用js引入即可，如下例。

  <script src="${base.contextPath}/common/code?cardStatusItem=WFL.MAIN\_FUNCTION" type="text/javascript"></script>

### **LOV配置**

lov可以快速创建弹窗对话框，如下图。



创建LOV需要有权限的管理员角色才能进行配置。进入系统，点击运维管理，点击LOV定义，如下图。



点击新建，输入相应的值，如下图。



最后，在需要的地方，直接引入LOV即可。如下例。

              <input name="assetType" type="text" id="assetType"

                data-bind="value:model.assetType,text:model.typeName" />

              <script>

                // 打开资产类别对话框

                $("#assetType").kendoLov({

                  contextPath: \_basePath,

                  locale: \_locale,

                  code: 'AMS\_OWN\_ASSET\_TYPE\_LOV', // lov代码

                  model: viewModel.model, // 数据源

                  textField: 'typeName',

                  // 查询条件

                  query: function (e) {

                    e.param['enabledFlag'] = 'Y';

                    e.param['orgCode'] = '1Z00';

                  },

                  // 选择

                  select: function (e) {

                    viewModel.model.set('assetType', e.item.typeNo);

                    viewModel.model.set('assetTypeName', e.item.typeName);

                  }

                });

              </script>

### **页面配置**

#### **菜单栏配置**

前端编写完页面后，需要将页面配置到系统。进入系统，点击功能维护，如下图。



点击新建按钮，输入对应的内容，功能代码和模块代码为前端页面文件名大写，如下图。



保存后，页面菜单栏将出现对应的菜单。如下图。



#### **详情页配置**

详情页配置与菜单栏的配置大体相同，详情页配置功能在系统的资源管理里面，如下图。



## **移动端**

移动端APP和小程序用的是同一套代码，移动端框架为mui和vue。

Mui开发文档：<https://dev.dcloud.net.cn/mui/>

移动端需要注意的点：

1. 需要引入移动端的css文件

<link rel="stylesheet" type="text/css" href="../../resources/css/dingding\_common.css" />

1. 移动端需要引入附件js文件，如果没引入这份文件，移动端看不了附件。

<script src="../../lib/mui/js/components/association\_file\_list.js" xmlns="http://www.w3.org/1999/html"></script>

1. 移动端可以引入vconsole进行实机调试

  <script src="https://s.url.cn/qqun/qun/qqweb/m/qun/confession/js/vconsole.min.js "></script>

1. 移动端截至目前（2022年6月17日）只能在uat环境发通知。