方正璞华移动开发平台

帮助手册

V1.1

2015年10月03日

# 文档版本

|  |  |  |
| --- | --- | --- |
| 版本 | 日期 | 说明 |
| 1.0 | 2015-08-11 | 初稿 |
| 1.1 | 2015-10-03 | 增加移动开发平台框架图 |
|  |  |  |

# 目录

[文档版本 2](#_Toc459218858)

[目录 4](#_Toc459218859)

[更新包结构说明 6](#_Toc459218860)

[BPMCS\_update 6](#_Toc459218861)

[lib 6](#_Toc459218862)

[mobile 6](#_Toc459218863)

[Web 7](#_Toc459218864)

[mobile\_client 7](#_Toc459218865)

[mobile\_form\_tools 7](#_Toc459218866)

[mobile\_manager 7](#_Toc459218867)

[移动开发平台框架图 8](#_Toc459218868)

[移动开发平台框架图 9](#_Toc459218869)

[开发说明 10](#_Toc459218870)

[Mobile Client 10](#_Toc459218871)

[配置文件 10](#_Toc459218872)

[移动表单开发 10](#_Toc459218873)

[表单常见控件 13](#_Toc459218874)

[mobile\_form\_tools 18](#_Toc459218875)

[使用方法 18](#_Toc459218876)

[Mobile manager 18](#_Toc459218877)

[用户管理 18](#_Toc459218878)

[角色管理 18](#_Toc459218879)

[菜单管理 19](#_Toc459218880)

[推送设备 19](#_Toc459218881)

[推送消息 19](#_Toc459218882)

[App管理 19](#_Toc459218883)

[二维码工具 19](#_Toc459218884)

[日志 19](#_Toc459218885)

[BPMCS 19](#_Toc459218886)

# 更新包结构说明

## BPMCS\_update

针对BPMCS系统的更新包，主要针对V3.7.X版本的BPMCS系统支持移动端功能。

### lib

将该文件夹下的jar放到BPMCS Web的lib目录下，并设置Web添加该三个jar的引用。如下图：

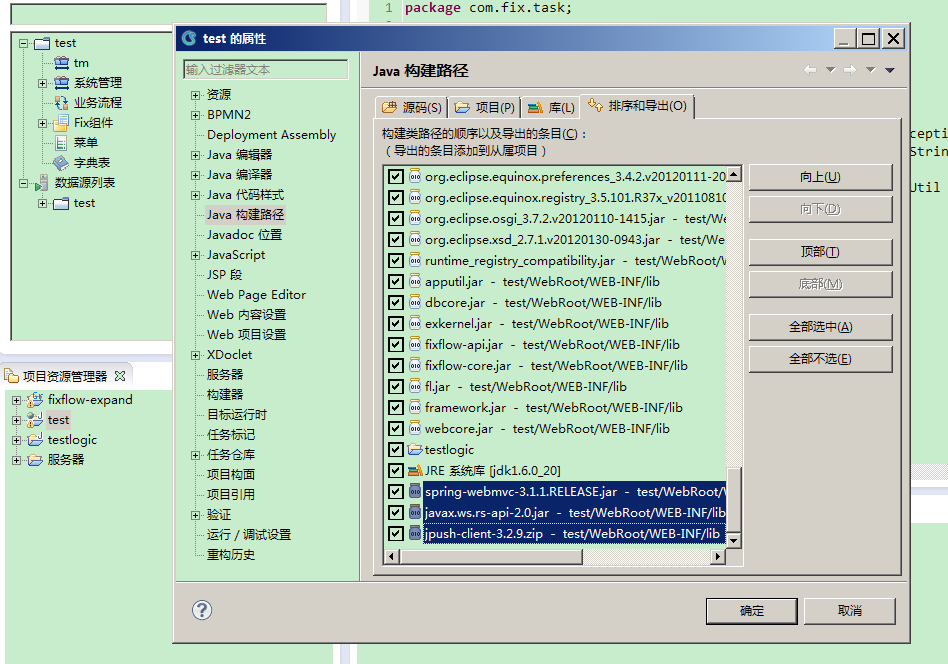


图 1

test为Web站点。

### mobile

BPMCS支持移动的接口实现，BPMCS的Web工程添加link source指向mobile/src，如下图：

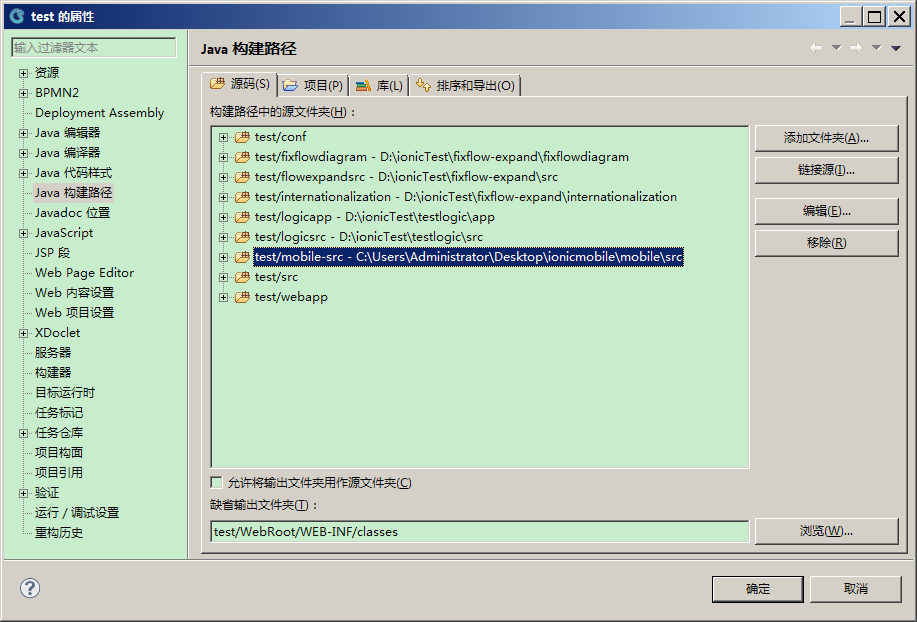


图 2

### Web

1. 导入服务组件Tasksev.zip。
2. 创建FlowQueryUtil.java和FlowTask.java，包名：com.fix.task。
3. 在web.xml中加入以下内容：

<!-- Spring MVC configuration -->

<servlet>

<servlet-name>springmvc</servlet-name>

<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

<init-param>

<param-name>contextConfigLocation</param-name>

<param-value>

classpath\*:springMVCRest\*.xml

</param-value><!-- 现定义为src下config包里(个人习惯) -->

</init-param>

<load-on-startup>1</load-on-startup>

</servlet>

<servlet-mapping>

<servlet-name>springmvc</servlet-name>

<url-pattern>/rest/interface/\*</url-pattern>

</servlet-mapping>

文档最后添加：

<!-- Rest Interface Filter -->

<filter>

<filter-name>RestApiFilter</filter-name>

<filter-class>com.founder.fix.filter.RestApiFilter</filter-class>

</filter>

<filter-mapping>

<filter-name>RestApiFilter</filter-name>

<url-pattern>/rest/\*</url-pattern>

</filter-mapping>

## mobile\_client

客户端程序，基于Ionic框架搭建，运行该客户端程序需要Ionic环境支持，Ionic环境的搭建详见文档《移动开发环境配置手册.docx》。

## mobile\_form\_tools

移动客户端的表单生成工具。

## mobile\_manager

移动客户端的后台管理系统，主要用于管理移动端的用户、菜单和权限、推送信息等信息。

图 3

# 移动开发平台框架图

图 4

Mobile Client为移动的客户端，包括Android和IOS两种移动操作系统，客户端基于Ionic框架实现移动端的展示功能，通过Restful的方式和业务系统（BPMCS）做业务交互；另外也通过Restful的方式和Mobile manager做功能上的交互，如绑定移动设备用于客户端推送、移动端用户和菜单的管理等。

# 移动开发平台请求流程图

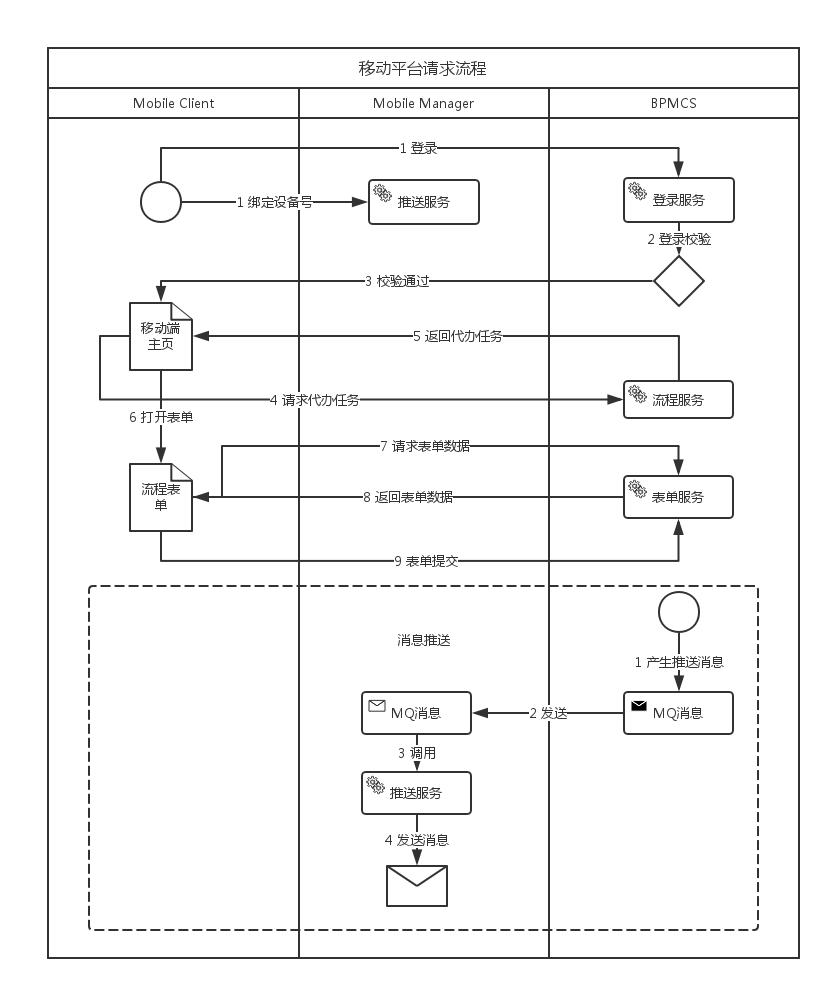


图 5

# 开发说明

## Mobile Client

mobile client基于混合式Ionic框架，大部分的操作基于ionic插件+AngularJs+JavaScript的方法实现功能的开发。

### 配置文件

#### config-url.js

mobile client与BPMCS系统、mobile manager交互的Restful接口地址。

**\_server:**与业务系统交互的地址，填写bpmcs系统的访问地址+/rest/interface。

**mServer:**与后台管理交互的地址，移动端后台管理url地址+/rest。

**\_mServer:**android版本app的更新地址。

**updateUrl\_ios:**iOS版本app更新地址（默认使用蒲公英App托管平台）。

#### config.js

该文件动态加载表单的配置文件，每添加一组表单则需要在config.js中添加引用。

另外selectGroup.flowDef为首页流程分类查询的下拉选择信息。

#### developer.js

该文件为动态加载表单扩展事件的文件。每当需要增加表单的加载后和保存前事件的调用时，需要在该文件中添加引用。

### 移动表单开发

以出差申请流程中的Daily\_Office/OA\_TRAVEL\_APPLICATION/TA\_Sure\_new.html表单为实例。

#### 新建表单

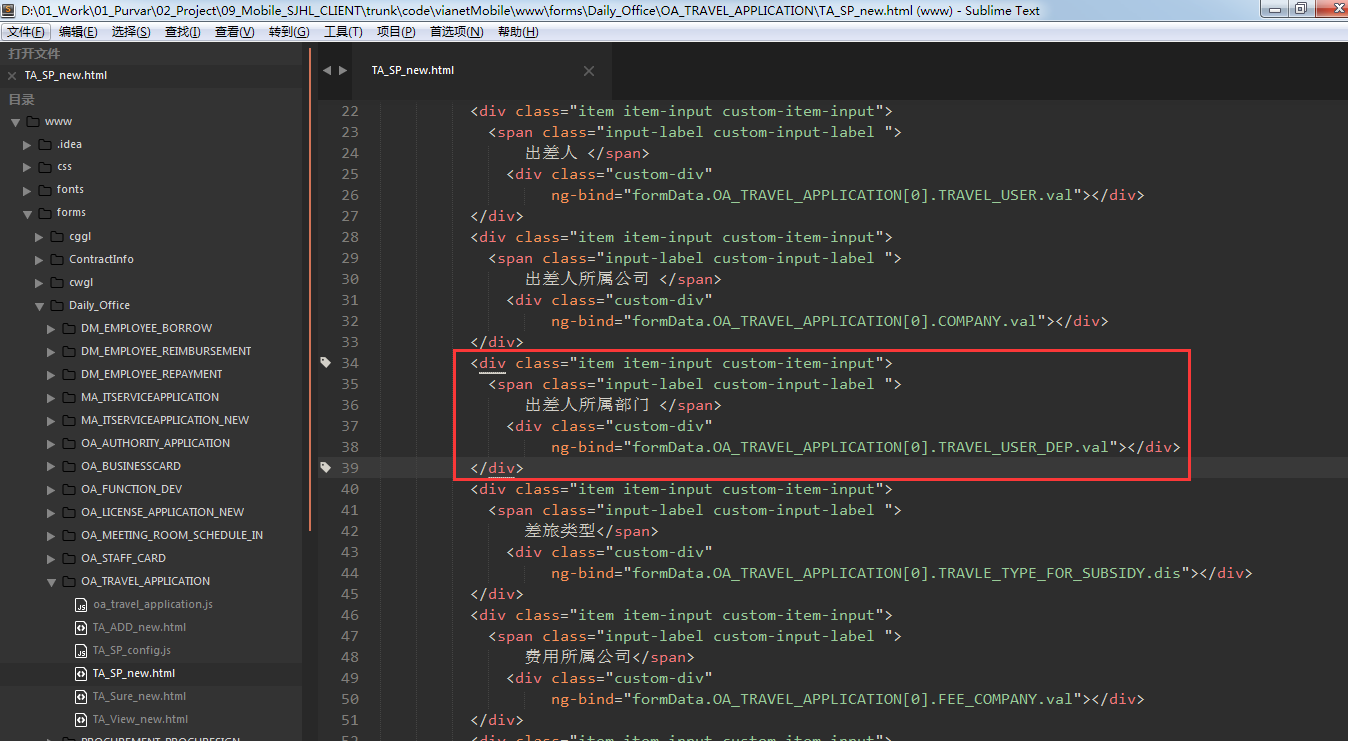


图 6

在mobile\_client\www\forms目录下建立和CS平台表单相同的目录结构（如Daily\_Office/OA\_TRAVEL\_APPLICATION），在文件夹下新建和CS平台表单名相同的表单文件，如图5。

表单文件Html页面代码参考现有的TA\_SP\_new.html。上图红框所示为一个字段的配置，所示范围内的ng-bind属性为指明div对应的数据formData.数据库表名[0].字段名.val（注：当控件为单选组、下拉框等显示值和实际值不相同的控件时，可以通过更改为ng-bind=””formData.数据库表名[0].字段名.dis”）。

**说明：**

移动端的表单路径和表单名要与PC端表单路径和表单名相同。

#### 添加表单执行事件

在TA\_SP\_new.html同级目录新建js文件，文件名随意，用于编写表单的事件扩展（可参考mobile\_client\www\forms\Daily\_Office\OA\_TRAVEL\_APPLICATION\oa\_travel\_application.js）。

|  |  |
| --- | --- |
| 5 | *var* OA\_TRAVEL\_APPLICATION\_TA\_SP\_new = { |
| 6 | afterDataLoad: *function* (*$scope*, *commonService*) { |
| 7 | //表单数据加载后事件。 |
| 8 | }, beforeDataSave: *function* (*$scope*, *commonService*) { |
| 9 | //表单保存前，如果不提交则return false |
| 10 | return true; |
| 11 | } |
| 12 | }; |

如上代码中，afterDataLoad是指页面加载后事件，beforeDataSave是指页面保存前事件。

在该js文件新建完成后，需要在developer.js中进行相应的配置，如下图。.

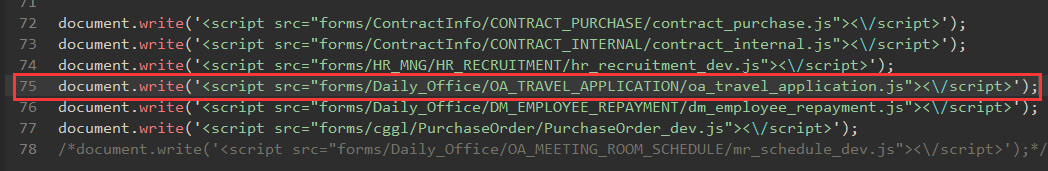


图 7

#### 新建表单配置文件

在TA\_SP\_new.html同级目录新建js文件，文件名随意，用于配置表单执行事件、表单需要提交的数据项等（可参考mobile\_client\www\forms\Daily\_Office\OA\_TRAVEL\_APPLICATION\ TA\_SP\_config.js）。

|  |  |
| --- | --- |
| 1 | FormConfig.OA\_TRAVEL\_APPLICATION = {}; |
| 2 | FormConfig.OA\_TRAVEL\_APPLICATION.TA\_SP\_new = { |
| 3 | formModal: { |
| 4 | onAfterDataLoad:OA\_TRAVEL\_APPLICATION\_TA\_Sure.afterDataLoad, |
| 5 | onBeforeDataSave : OA\_TRAVEL\_APPLICATION\_TA\_Sure.beforeDataSave, |
| 6 | formTabs: [ |
| 7 | { |
| 8 | pageId: "OA\_TRAVEL\_APPLICATION", |
| 9 | tabName: "业务信息" |
| 10 | }, { |
| 11 | pageId: "detail-OA\_TRAVEL\_APPLICATION\_DETAIL", |
| 12 | tabName: "行程" |
| 13 | } |
| 14 | ] |
| 15 | }, |
| 16 | relatedObj: { |
| 17 | OA\_TRAVEL\_APPLICATION: { |
| 18 | APPLICATION\_ID: { |
| 19 | "isPk": true, |
| 20 | "dataType": "BigDecimal" |
| 21 | } |
| 22 | }, |
| 23 | "OA\_TRAVEL\_APPLICATION\_DETAIL": { |
| 24 | APPLICATION\_DETAILID: { |
| 25 | "isPk": true, |
| 26 | "dataType": "BigDecimal" |
| 27 | } |
| 28 | } |
| 29 | } |
| 30 | } |

在该文件中配置了表单的加载后和保存前的事件声明，若表单不需要这两个事件则可以不声明。formTabs为表单上的Tab页配置，pageId为表单上区域的div的id，tabName为标签页上显示的tab名称。

relatedObj对象为表单在保存时需要提交的字段。

表单配置文件新建完成后，需要在config.js中添加对该配置文件的引用，如下图。

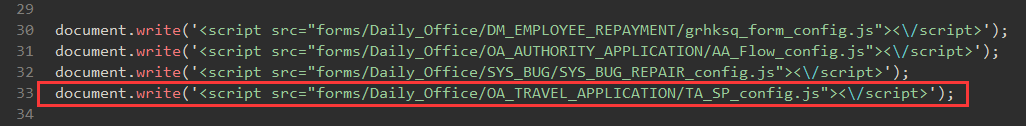


图 8

#### 说明

表单路径为模块\业务对象，针对当前业务对象的所有表单统一放在该目录下。同一个对象表单文件夹下，只能有一个developer.js和config.js文件，所有html文件相应配置都在这两个js文件中配置。

### 表单常见控件

#### 单选组

**表单页面Html代码**

|  |  |
| --- | --- |
| 1 | <div class="item item-input custom-item-input"> |
| 2 | <span class="input-label custom-input-label "> |
| 3 | 是否框架协议<FONT color=#ff0000>\*</FONT> |
| 4 | </span> |
| 5 | <ion-radio ng-model="formData.CONTRACT\_PURCHASE[0].FRMAEAGREEMENT.val" ng-value="'1'"name="kjxy"> |
| 6 | 是 |
| 7 | </ion-radio> |
| 8 | <ion-radio ng-model="formData.CONTRACT\_PURCHASE[0].FRMAEAGREEMENT.val" ng-value="'0'"name="kjxy"> |
| 9 | 否 |
| 10 | </ion-radio> |
| 11 | </div> |

在同一个表单页面上有多个单选组时，通过name属性来进行区别。

#### 日期控件

**表单页面Html代码**

|  |  |
| --- | --- |
| 1 | <div class="item item-input custom-item-input"> |
| 2 | <span class="input-label custom-input-label ">付款日期 |
| 3 | <FONT color=#ff0000>\*</FONT> |
| 4 | </span> |
| 5 | <input id="TRANSFER\_DATE" type="text" mobiscroll-date ng-model="formData.FM\_MONEY\_TRANSFER[0].TRANSFER\_DATE.val" /> |
| 6 | <i class="ion-calendar"></i> |
| 7 | </div> |

**Js代码**

|  |  |
| --- | --- |
| 1 | angular.element("#TRANSFER\_DATE").mobiscroll().date({ |
| 2 | theme: 'android-holo-light', |
| 3 | lang: 'zh', |
| 4 | display: 'modal', |
| 5 | dateFormat: "yy-mm-dd", |
| 6 | mode: 'mixed', |
| 7 | minDate: new *Date*() //最小日期控制 |
| 8 | }); |

其中TRANSFER\_DATE为表单input的id。

#### Select控件

**表单页面Html代码**

|  |  |
| --- | --- |
| 1 | <div class="item item-input custom-item-input"> |
| 2 | <span class="input-label custom-input-label "> |
| 3 | 保险计划<FONT color=#ff0000>\*</FONT> |
| 4 | </span> |
| 5 | <div class="custom-select"> |
| 6 | <select class="" id="bxjhselect" ng-model="formData.HR\_EMPLOYEE\_ENTRY[0].INSURANCE\_PLAN.val"> |
| 7 | <option value="">--请选择--</option> |
| 8 | <option value="{{lx.CATEGORY\_VALUE}}" ng-if="formData.HR\_EMPLOYEE\_ENTRY[0].INSURANCE\_PLAN.val.indexOf(lx.CATEGORY\_VALUE) >= 0" selected ng-repeat="lx in BXJH">{{lx.DISPLAY}} |
| 9 | </option> |
| 10 | <option value="{{lx.CATEGORY\_VALUE}}" ng-if="formData.HR\_EMPLOYEE\_ENTRY[0].INSURANCE\_PLAN.val.indexOf(lx.CATEGORY\_VALUE) < 0" ng-repeat="lx in BXJH">{{lx.DISPLAY}} |
| 11 | </option> |
| 12 | </select> |
| 13 | </div> |
| 14 | <i class="ion-ios-arrow-down"></i> |
| 15 | </div> |

**Js代码**

|  |  |
| --- | --- |
| 1 | *var* param = { |
| 2 | "filter": "category='insurancePlan'", |
| 3 | "fields": "CATEGORY\_VALUE,DISPLAY", |
| 4 | "orderList": [{ |
| 5 | "order": "asc", |
| 6 | "field": "CATEGORY\_VALUE" |
| 7 | }], |
| 8 | "bizObj": "SYS\_CATEGORY", |
| 9 | "service": "selectMore", |
| 10 | "pageSize": 9999, |
| 11 | currentPageIndex: 1 |
| 12 |  |
| 13 | } |
| 14 | commonService.callMethod("component.getListData", param, *function*(*data*) { |
| 15 | if (data.datas) { |
| 16 | $scope.BXJH = data.datas; |
| 17 | setTimeout(*function*() { |
| 18 | *var* select = angular.element("#bxjhselect").mobiscroll().select({ |
| 19 | theme: 'android-holo-light', |
| 20 | lang: 'zh', |
| 21 | display: 'modal', |
| 22 | minWidth: 200, |
| 23 | onSelect: *function*(*valueText*, *inst*) { |
| 24 |  |
| 25 | } |
| 26 | }); |
| 27 |  |
| 28 | }, 200) |
| 29 | } |
| 30 | }); |

上述代码放在表单加载后事件中，param所定义的参数和PC端控件配置的参数相同。

#### 模糊查询控件

**表单页面Html代码**

|  |  |
| --- | --- |
| 1 | <div class="item item-input custom-item-input"> |
| 2 | <span class="input-label custom-input-label "> |
| 3 | 虚线领导 |
| 4 | </span> |
| 5 | <div id="INVENTEDLEAD" class="custom-div custom-div-autocomplete"> |
| 6 | <input type="text" class="custom-input input-autocomplete" ng-model="formData.HR\_EMPLOYEE\_ENTRY[0].INVENTEDLEADFULL.val" /> |
| 7 | <button ng-click="showSelectINVENTEDLEAD()" class="button button-calm button-autocomplete">选择</button> |
| 8 | <div class="custom-select-r" style="*display*: none"> |
| 9 | <select class="" id="INVENTEDLEAD\_auto"> |
| 10 | <option ng-repeat="person in INVENTEDLEAD | filter:formData.HR\_EMPLOYEE\_ENTRY[0].INVENTEDLEADFULL.val" value="{{person.EMPLOYEE\_CODE + ',' + person.DISPLAYNAME}}"> |
| 11 | {{person.DISPLAYNAME}} |
| 12 | </option> |
| 13 | </select> |
| 14 | </div> |
| 15 | </div> |
| 16 | </div> |

**Js代码**

|  |  |
| --- | --- |
| 1 | *var* select = angular.element("#INVENTEDLEAD\_auto").mobiscroll().select({ |
| 2 | theme: 'android-holo-light', |
| 3 | lang: 'zh', |
| 4 | display: 'modal', |
| 5 | minWidth: 200, |
| 6 | onSelect: *function*(*valueText*, *inst*) { |
| 7 | setTimeout(*function*() { |
| 8 | $scope.$apply(*function*() { |
| 9 |  |
| 10 | }); |
| 11 | }, 200); |
| 12 | } |
| 13 | }); |
| 14 | *var* param = { |
| 15 | "filter": "1=1", |
| 16 | "fields": "EMPLOYEE\_FULL,DISPLAYNAME,EMPLOYEE\_CODE", |
| 17 | "bizObj": "BASEINFO", |
| 18 | "service": "getUserAllInfo", |
| 19 | "pageSize": 9999, |
| 20 | currentPageIndex: 1 |
| 21 |  |
| 22 | } |
| 23 | commonService.callMethod("component.getListData", param, *function*(*data*) { |
| 24 | if (data.datas) { |
| 25 | $scope.INVENTEDLEAD = data.datas; |
| 26 | } |
| 27 | }); |

在有些应用场景下，模糊查询框显示的不是已有字段，需要自定义新字段，如实例中的INVENTEDLEADFULL。

#### Ajax调用

**Js代码**

|  |  |
| --- | --- |
| 1 | *var* param = { |
| 2 | companyCode: companyCode |
| 3 | } |
| 4 | commonService.callMethod("CONTRACT\_SALE.getCompanyIdentity", param, *function*(*response*) { |
| 5 | if (response) { |
| 6 |  |
| 7 | } |
| 8 | }, *function*() { |
| 9 |  |
| 10 | }); |

客户端执行ajax调用，可以直接在表单事件中调用commonService.callMethod方法即可，入参方法名（也是CS平台的服务编号）、方法参数、执行成功的回调函数和失败时的回调函数。

#### 附件展示

**表单页面Html代码**

|  |  |
| --- | --- |
| 1 | <div class="custom-div" ng-if="formData.CONTRACT\_PURCHASE[0].ATTACHMENT.val==''">无</div> |
| 2 | <div class="custom-div attachment"> |
| 3 | <div ng-repeat="attach in formData.OA\_AUTHORITY\_APPLICATION[0].ATTACHMENT.attach" ng-click="fileDownLoad(attach.fieldToken, attach.fileId, attach.fileOldName)"> |
| 4 | {{attach.fileOldName}} |
| 5 | </div> |
| 6 | </div> |

**Js代码**

|  |  |
| --- | --- |
| 1 | getAttachment: *function*(*$scope*, *commonService*) { |
| 2 | *var* fieldToken = $scope.formData.CONTRACT\_SALE[0].ATTACHMENT.val; |
| 3 | if (fieldToken && fieldToken != "") { |
| 4 | *var* \_method = "streamSupportService.getAttachmentInfos"; |
| 5 | *var* \_param = []; |
| 6 | *var* o = {}; |
| 7 | o.fieldToken = fieldToken; |
| 8 | \_param.push(o); |
| 9 | commonService.callMethod(\_method, \_param, *function*(*data*) { |
| 10 | if (data && data.attachments) { |
| 11 | $scope.formData.CONTRACT\_SALE[0].ATTACHMENT.attach = data.attachments; |
| 12 | } |
| 13 | }); |
| 14 | } |
| 15 | } |

getAttachment为通过fieldToken来获取表单相关的附件信息，该方法在表单的加载后事件中执行。

**注意修改上述代码中的相关参数。**

#### 控件说明

以上控件若在明细表中使用，则需要在表单加载后事件中循环初始化明细表中的每一个控件。

## mobile\_form\_tools

该工程为生成移动表单和表单配置文件的工具。

### 使用方法

使用eclipse打开该工程，确认工程可用后。打开Generator.java文件，修改main方法中的变量formDirPath的值为PC端表单据对路径，如“D:\\01\_Work\\01\_Purvar\\02\_Project\\09\_Mobile\_SJHL\_CLIENT\\Fix BPMCS\\workspace\\sjhl\\WebRoot\\forms\\Daily\_Office\\Daily\_Office\\OA\_TRAVEL\_APPLICATION”，变量distDir为生成后的移动表单和表单配置文件的目录，变量修改完成后执行右键🡺Run As🡺Java Application，执行完成后，将distDir配置的文件目录下生成的表单文件和表单配置文件放到Mobile Client中表单对应的路径下。

## Mobile manager

基于springMVC的移动平台的管理后台，用于移动客户端获取最新App最新版本、绑定设备号、推送配置等功能。登录账户：admin,密码:123456

### 用户管理

用于管理app的用户信息，此用户管理和哪些账号可以登录移动端无关，而是用于管理移动端菜单权限的基础用户数据。

用户管理支持同步用户数据，支持将其他系统的用户数据一键同步到Mobile Manager系统中，该功能具体实现在AuthUserInfoAction.syncUserInfo方法中。

可执行用户数据的增删改查操作以及针对用户赋角色功能。

### 角色管理

用于管理移动端的角色信息，可以在mobile manager管理菜单的权限，给菜单赋角色的权限信息。并可以针对角色信息执行增删改查以及批量将用户添加到角色的操作。

### 菜单管理

管理移动端的菜单信息，可对菜单信息进行增删改查以及赋权限的操作。

### 推送设备

查看mobile client推送培训信息，包括账号、设备号、推送消息免打扰相关配置等信息，同时也是单点推送的基础数据，用于将PC端系统产生的到个人的消息推送到设备上。

### 推送消息

记录推送消息以及消息推送的状态，错误信息。

### App管理

主要用于管理Android版本的app版本发布情况，移动客户端通过接口获取到最新的版本，并提示用户下载最新版本的apk覆盖当前应用。

可以对app版本执行增删改查操作，以及生成发布过的app下载地址的二维码。

### 二维码工具

将自定义的字符串生成为二维码的图片，并支持二维码图片的下载。

### 日志

记录客户端系统的操作日志，目前只记录客户端的登录日志。若需要记录其他的客户端操作，可在客户端中调用日志记录接口。

## BPMCS

正常的开发业务即可，包括流程和表单等其他操作。按照更新说明更新BPMCS系统后，BPMCS提供Restful接口，用于系统登录、获取代办事项、表单数据保存、流程提交、查看附件等接口。