bpmx3手机端开发

[bpmx3手机端开发 1](#_Toc347492918)

[1. 手机端与x3系统访问原理 2](#_Toc347492919)

[2. 开发环境 2](#_Toc347492920)

[2.1 Sencha touch 2.1 2](#_Toc347492921)

[2.2 android 2 2](#_Toc347492922)

[2.3 android 4 2](#_Toc347492923)

[3. android开发手机入口 3](#_Toc347492924)

[3.1 activity\_main.xml 页面布局 3](#_Toc347492925)

[3.2 MainActivity.java 页面事件处理 3](#_Toc347492926)

[3.3 AndroidManifest.xml 3](#_Toc347492927)

[3.4 strings.xml 文本配置 3](#_Toc347492928)

[3.5 assets/config.xml 文本配置 3](#_Toc347492929)

[3.5 效果 4](#_Toc347492930)

[4. Sencha touch 开发bpmx3 手机端页面 4](#_Toc347492931)

[4.1 登录 4](#_Toc347492932)

[4.2 mobileMain.js功能菜单 5](#_Toc347492933)

[4.3 mobileUserInfo.js 用户信息 5](#_Toc347492934)

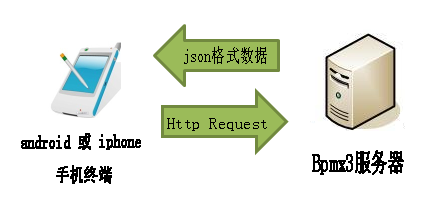
[4.4 具体实现功能 5](#_Toc347492935)

[4.5 自定义开发表单 15](#_Toc347492936)

[5. 打包 18](#_Toc347492937)

[6. 改变访问路径及显示图标 19](#_Toc347492938)

## 手机端与x3系统访问原理



## 开发环境

### 2.1 Sencha touch 2.1

基于ext4 的移动端开发框架

### 2.2 android 2

对应市场上大多数安装android2系统的移动终端

### 2.3 android 4

对应市场上大多数安装android4系统的移动终端

## android开发手机入口

### 3.1 activity\_main.xml 页面布局

android:id=*"@+id/bpmWebView"*

android 内置web浏览器

android:id=*"@+id/loading"*

等待页面，这是一个容器，里面包括了 ProgressBar 和输出等待信息

### 3.2 MainActivity.java 页面事件处理

bpmWebview 内置浏览器

onCreate 手机端初始化函数

AsyncTask 页面加载异步处理

doInBackground 异步加载函数

onPostExecute 异步处理结束，登录页面显示

### 3.3 AndroidManifest.xml

允许内部浏览器访问网络

<uses-permission android:name=*"android.permission.INTERNET"* />

### 3.4 strings.xml 文本配置

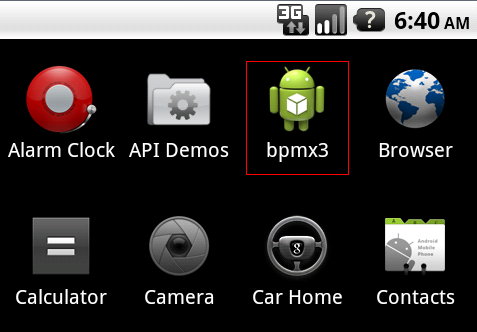
*app\_name* 标题

*loading* 等待信息

### 3.5 assets/config.xml 文本配置

*url* 更改手机端访问系统的路径

### 3.5 效果



## Sencha touch 开发bpmx3 手机端页面

### 4.1 登录

init.js 总入口加载登录页

login.js 手机入口登录页

MobileLoginController 手机端登录事件处理



### 4.2 mobileMain.js功能菜单

配置 MenuConfig 菜单信息

{

cls:'myTask', 定义功能的图标样式

title:'我的待办', 定义功能名称

view:'mobile.MyTask' 功能实现页面

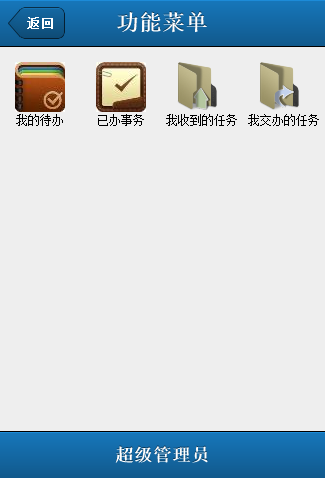
}

在 mobile.css加上样式

*.myTask*{

background:*url(../images/mobile/myTask.jpg)* *no-repeat*;

}



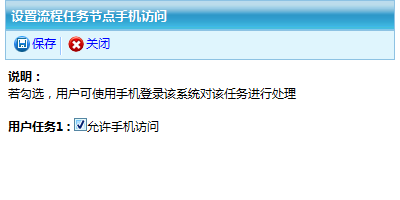
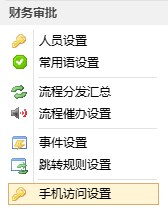
### 4.3 mobileUserInfo.js 用户信息

如上图，在登录成功后，底部显示用户信息

### 4.4 具体实现功能

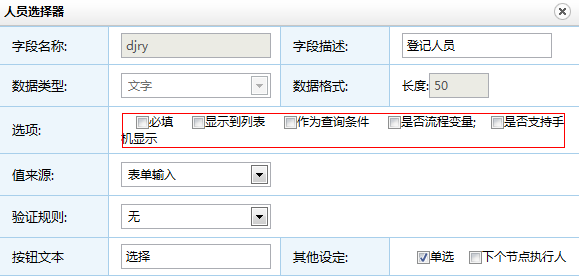
要bpmx3 系统进行手机节点设置

在流程定义的节点设置，如下图



这样可在手机中看到改节点任务

表单编辑中设置字段是否 支持手机显示



##### 4.4.1 我的待办

###### 4.4.1.1 查询列表

查询列表入口：platform/mobile/mobileTask/myTask.ht

controller对应方法：

**public** Object myTask(

HttpServletRequest request,HttpServletResponse response)

对应service接口：

filter.addFilter("userId", userId) // 用户ID

filter.addFilter("subject", subject) // 任务标题

bpmService.getMyMobileTasks(filter)

返回格式：

{

id: 10000011380027, // 任务id

subject: 手机url测试-admin-2012-12-27 15:46:02 // 标题

}

Task.map.xml对应查询语句：getMyMobileTasks



###### 4.4.1.2 查看表单

查看表单入口：platform/mobile/mobileTask/getMyTaskForm.ht

controller对应方法：

**public** Object getMyTaskForm(

HttpServletRequest request,HttpServletResponse response)

对应service接口

mobileTaskService.getFormData(

taskId, // 任务id

request.getContextPath() // 项目根目录

)

返回数据格式：

{

taskId=10000011380027, 任务id

form={ 表单

"emptyForm":true, 是否空表单

"extForm":true, 是否url表单

"fields":"", 主表字段

"formDetailUrl":"", url明细表单

"formEditUrl":"", url编辑表单

"options":{}, 审批意见

"signTask":false, 是否会签任务

"subTableList":[], 子表字段

"tableDesc":"", 主表描述

"tableId":0, 主表id

"tableName":"" 主表名称

}

}



##### 4.4.2 已办事务

###### 4.4.2.1 查询列表

查询列表入口：platform/mobile/mobileTask/myAttend.ht

controller对应方法：

**public** Object myAttend(

HttpServletRequest request,HttpServletResponse response)

对应service接口：

filter.addFilter("userId", userId) // 实例ID

filter.addFilter("subject", subject) // 任务标题

processRunService.getMyAttend(filter)

返回格式：

{

runId: 10000011380027, // 实例id

subject: 手机url测试-admin-2012-12-27 15:46:02 // 标题

}

ProcessRun.map.xml对应查询语句：*getMyAttend*

###### 4.4.2.2 查看表单

查看表单入口：platform/mobile/mobileTask/getMyAttendForm.ht

controller对应方法：

**public** Object getMyAttendForm(

HttpServletRequest request,HttpServletResponse response)

对应service接口

mobileTaskService.getProcessData(

runId, 实例id

ContextUtil.*getCurrentUserId*(), 用户id

ctxPath, 项目根目录

taskId 任务id

)

返回数据格式：

{

taskId=10000011380027, 任务id

form={ 表单

"emptyForm":true, 是否空表单

"extForm":true, 是否url表单

"fields":"", 主表字段

"formDetailUrl":"", url明细表单

"formEditUrl":"", url编辑表单

"options":{}, 审批意见

"signTask":false, 是否会签任务

"subTableList":[], 子表字段

"tableDesc":"", 主表描述

"tableId":0, 主表id

"tableName":"" 主表名称

}

}

##### 4.4.3 交办给我的任务

###### 4.4.3.1 查询列表

查询列表入口：platform/mobile/mobileTask/myAssign.ht

controller对应方法：

**public** Object myAssign(

HttpServletRequest request,HttpServletResponse response)

对应service接口：bpmTaskAssigneeService.getAllMyTask

对应service接口：

filter.addFilter("assigneeId ", assigneeId) // 收到任务用户id

filter.addFilter("subject", subject) // 任务标题

bpmTaskAssigneeService.getAllMyTask(filter)

返回格式：

{

runId: 10000011380027, // 实例id

taskId: 10000010660009, // 任务id

subject: 手机url测试-admin-2012-12-27 15:46:02 // 标题

taskStatus: 1 // 任务状态，1：待办，2： 已办

}

BpmTaskAssignee.map.xml对应查询语句：getAllMyTask

以assigneeId 作为查询条件

###### 4.4.3.2 查看表单

查看表单入口：platform/mobile/mobileTask/getMyAssigneeForm.ht

controller对应方法：

**public** Object getMyAssigneeForm(

HttpServletRequest request,HttpServletResponse response)

对应service接口

mobileTaskService.getProcessData(

runId, 实例id

ContextUtil.*getCurrentUserId*(), 用户id

ctxPath, 项目根目录

taskId 任务id

)

返回数据格式：

{

taskId=10000011380027, 任务id

form={ 表单

"emptyForm":true, 是否空表单

"extForm":true, 是否url表单

"fields":"", 主表字段

"formDetailUrl":"", url明细表单

"formEditUrl":"", url编辑表单

"options":{}, 审批意见

"signTask":false, 是否会签任务

"subTableList":[], 子表字段

"tableDesc":"", 主表描述

"tableId":0, 主表id

"tableName":"" 主表名称

}

}

##### 4.4.4 我交办的任务

###### 4.4.4.1 查询列表

查询列表入口：platform/mobile/mobileTask/myAttend.ht

controller对应方法：

**public** Object myAttend(

HttpServletRequest request,HttpServletResponse response)

对应service接口：

filter.addFilter("userId", userId) // 用户id

filter.addFilter("subject", subject) // 任务标题

bpmTaskAssigneeService.getAllMyTask(filter)

返回格式：

{

runId: 10000011380027, // 实例id

subject: 手机url测试-admin-2012-12-27 15:46:02 // 标题

}

BpmTaskAssignee.map.xml对应查询语句：getAllMyTask

以userId 作为查询条件

###### 4.4.4.2 查看表单

查看表单入口：platform/mobile/mobileTask/getMyAttendForm.ht

controller对应方法：

**public** Object getMyAttendForm(

HttpServletRequest request,HttpServletResponse response)

对应service接口

mobileTaskService.getProcessData(

runId, 实例id

ContextUtil.*getCurrentUserId*(), 用户id

ctxPath, 项目根目录

taskId 任务id

)

返回数据格式：

{

taskId=10000011380027, 任务id

form={ 表单

"emptyForm":true, 是否空表单

"extForm":true, 是否url表单

"fields":"", 主表字段

"formDetailUrl":"", url明细表单

"formEditUrl":"", url编辑表单

"options":{}, 审批意见

"signTask":false, 是否会签任务

"subTableList":[], 子表字段

"tableDesc":"", 主表描述

"tableId":0, 主表id

"tableName":"" 主表名称

}

}

##### 4.4.5 查看流程图

UserImage.js 查看流程图，使用iframe嵌入jsp的方式

###### 4.4.5.1 查询列表处查看

Util.js 中提供了查看方法

userImage(

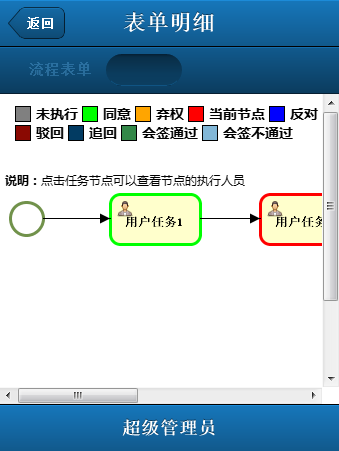
type: 查看流程图的id类型

id: 对应流程图的id

)



###### 4.4.5.2 明细表单处查看



##### 4.4.6 提交审批意见及执行任务

执行路径：platform/mobile/mobileTask/submitForm.ht

**Controller对应方法**

**public** Object submitForm(

HttpServletRequest request,HttpServletResponse response)

接口方法实现：

mobileTaskService.complete(

taskId, 任务id

opinion, 审批意见

userAccount, 用户id

voteAgree

)

###### 4.4.6.1 普通任务

当取得表单signTask 标记返回 false 为普通任务



###### 4.4.6.2 会签任务

当取得表单signTask 标记返回 true 为会签任务

标记voteAgree，1：完成；2：反对；0：弃权；5：直接完成结束；6：直接反对结束



###### 4.4.6.2 url表单

"extForm":true, 是否url表单

"formDetailUrl":"", url明细表单

"formEditUrl":"", url编辑表单

如果该表单为 url 表单，查看表单处则直接读取url

### 4.5 自定义开发表单

以下做一个定义表单的demo

建表脚本：

create table mobile\_user\_info(

userid NUMBER(18) NOT NULL,

username VARCHAR2(127) NOT NULL,

idCard VARCHAR2(40) NOT NULL,

CONSTRAINT PK\_mobile\_user\_info PRIMARY KEY (userid)

)

代码生成器添加url表单的两个文件

一个是明细查看

一个表单编辑



代码生成器配置 codegenconfig.xml

<templates>

. . .

<template id=*"mobileForm"* path=*"mobileform.ftl"*/>

<template id=*"mobileDetail"* path=*"mobiledetail.ftl"*/>

. . .

</templates>

<files baseDir=*"D:/dev/hotent/bpmx31"* >

. . .

<file refTemplate=*"mobileForm"*

filename=*"{class}Form.js"* dir=*"src\main\webapp\js\hotent\{system}\{package}"* />

<file refTemplate=*"mobileDetail"*

filename=*"{class}Detail.js"* dir=*"src\main\webapp\js\hotent\{system}\{package}"* />

. . .

</files>

<table tableName=*"MOBILE\_USER\_INFO"* >

<variable name=*"class"* value=*"MobileUserInfo"* />

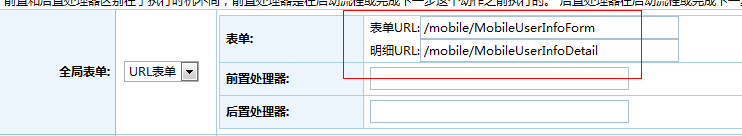
<variable name=*"classVar"* value=*"mobileUserInfo"* />

<variable name=*"package"* value=*"mobile"* />

<variable name=*"path"* value=*"mobile"* />

</table>

表单设置配置路径



用代码生成器生成相关文件

修改表单文件 MobileUserInfoForm.js

在items 处添加控件





点击 完成，反对，弃权 相关动作。会提交到MobileUserInfoController里的save方法

@RequestMapping("save")

@Action(description="添加或更新MOBILE\_USER\_INFO")

@ResponseBody

**public** Object save(HttpServletRequest request, HttpServletResponse response) **throws** Exception

{

Map responseMap = **new** HashMap();

responseMap.put("success", **true**);

**return** responseMap;

}

方法头要声明 @ResponseBody 代表返回json格式

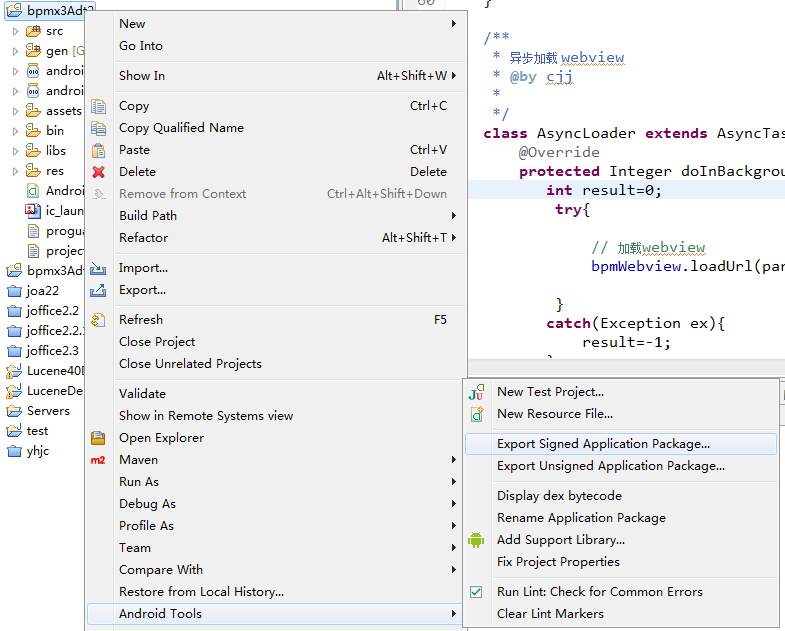
方法的返回类型是 Object 里面用一个 HashMap 存储

便于 ext 接收数据

## 打包

利用了android 的打包插件进行打包

项目根目录 – android tools – export application package

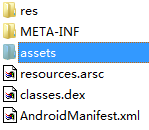


## 改变访问路径及显示图标

打包后会生成后缀名为 \*.apk 的android 安装文件。



把后缀名改为 rar，我们可以修改，访问系统的路径和自定义图标



assets目录下有 config.properties 修改其中url 可以改变手机端访问路径

res 目录下可以修改显示图标

手机程序名要在源程序修改