**iuap移动平台Moli**

**移动开发快速入门**

# 概述

本文档通过一个实例从新建工程到MA Server端代码编写以及调整来说明如何快速了解MA Server开发。

MA Server快速入门开发分为如下步骤：

* 第一步、开发环境配置
* 第二步、创建移动summer工程和MA服务
* 第三步、编写summer工程端的代码和MA服务代码
* 第四步、运行本地MA Server进行前后端联调开发
* 第五步、部署到MA Server正式服务器

我们以下进行详细介绍。

# 开发环境配置

在开发之前，先确认iuapMobile Studio的BuildServer、MA等相关配置完成。如果MA没有配置，可按照如下进行配置。与MA相关的共有如下两个配置：

* 配置MA Home
* 配置Tomcat Home

## 1.1配置MA Home

* **第一步、启动iuap移动开发工具Studio**

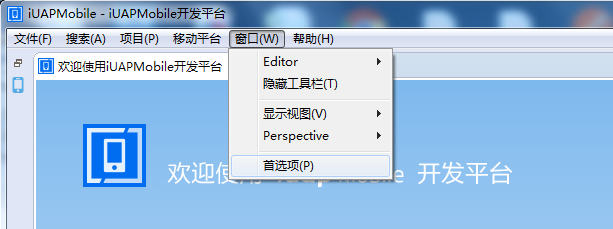
首先启动iuap移动开发工具Studio并登陆。

注：第一次需要指定一个工作空间，这些操作参考Eclipse的操作，因为开发工具studio是基于Eclipse的。

如果不登录会影响移动应用打包，因为移动应用打包需要在云移动应用构建服务器上进行。

* **第二步、配置MA Home**

在开发工具菜单栏中依次打开【窗口】下的【首选项】，如下图所示：



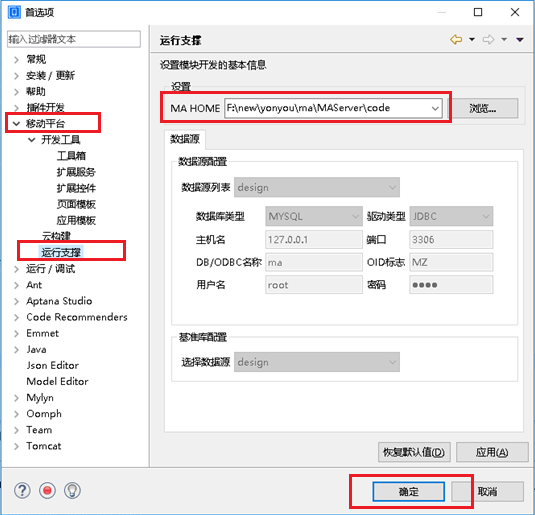
**注：**

**Windows版本的开发工具在【窗口】弹出菜单下找到【首选项】**

**Mac版的开发工具需要在工具栏的【iUAPMobile】弹出的菜单里找到【偏好设置】，如下图所示：**



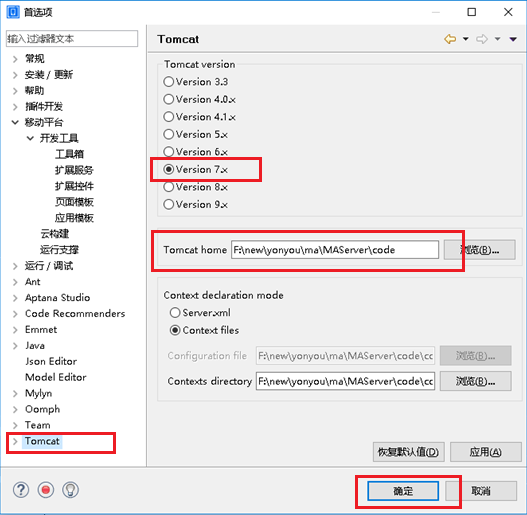
在弹出的【首选项】页面在左侧菜单树点击打开【移动平台】下的【运行支撑】，在右面进行MA HOME的配置，在输入框输入自己MAServer安装的路径。注意填写路径到的末级，完成后单击【确定】，如下图所示：



## 1.2配置Tomcat

MA Server是基于Tomcat的，因此需要在开发工具上通过配置Tomcat来实现本地运行调试，过程如下：

打开菜单栏中【窗口】下的【首选项】（mac的在【偏好设置】），点击选中左面菜单树【Tomcat】，右面进行Tomcat配置，因为本版本是基于Tomcat 7.x版本的，所以选择【Version 7.x】，完成后单击【确定】，如下图所示：



至此，开发环境配置完成了。

# 创建移动应用工程

一个移动工程总的来说包含两个部分：

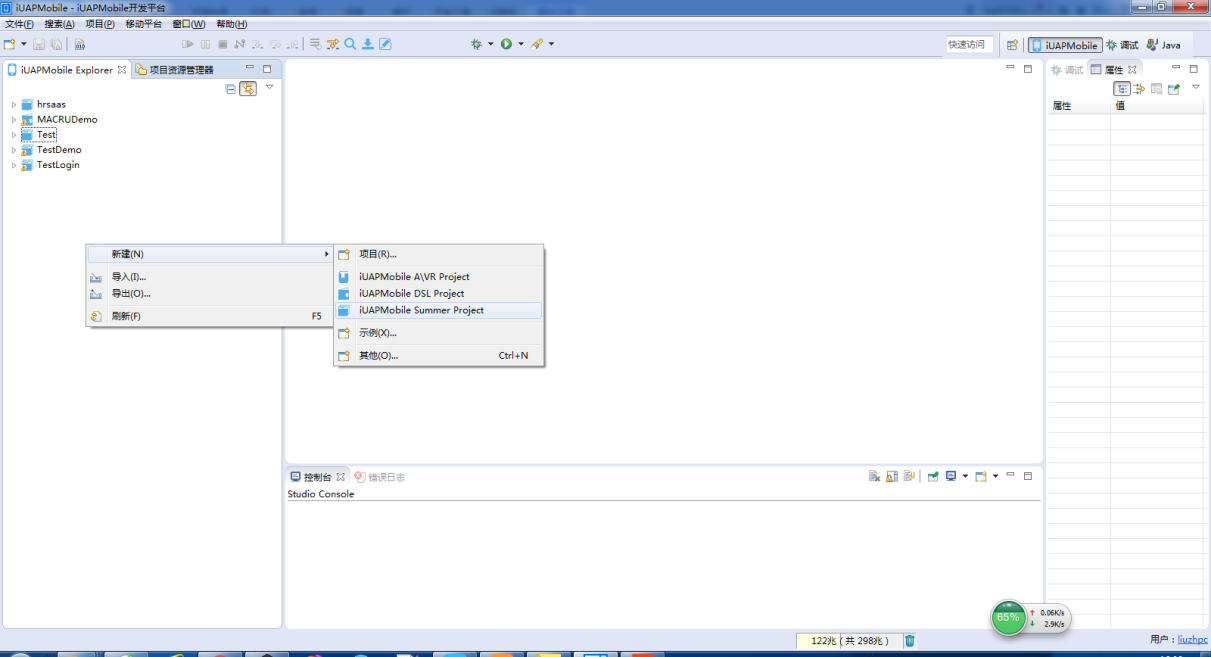
* APP端
* MA服务端

我们以summer工程来进行移动应用APP端的开发，通过APP端来请求MAServer的服务。

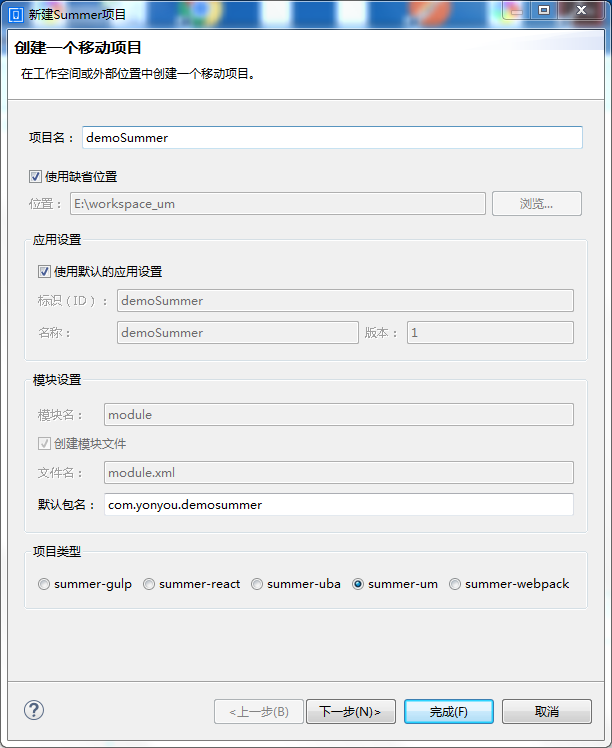
## 2.1新建summer工程

Summer开发框架是以HTML5技术进行移动开发的开发框架。我们以summer工程来介绍如何基于iuap移动开发平台进行快速开发。首先通过以下三步创建summer工程：

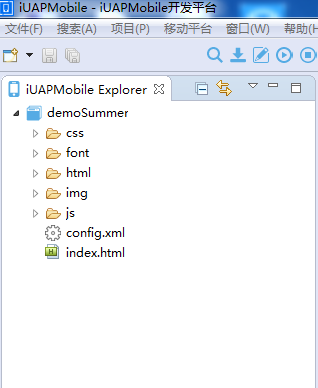
第一步、在iuapMobile视图下空白处鼠标右键->新建->项目->选择iuapMobile Summer Project，如图：



第二步、在弹出的【新建summer项目】向导页的【项目名】输入要创建的项目名，直接单击【完成】即可。本例子我们命名工程为demoSummer，如下图所示：



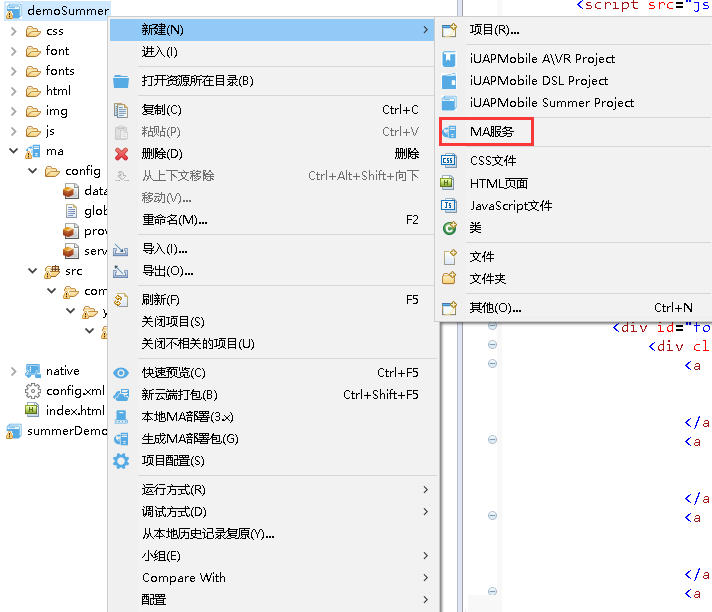
第三步、直接点击【完成】后，默认summer工程创建成功，目录结构如下：



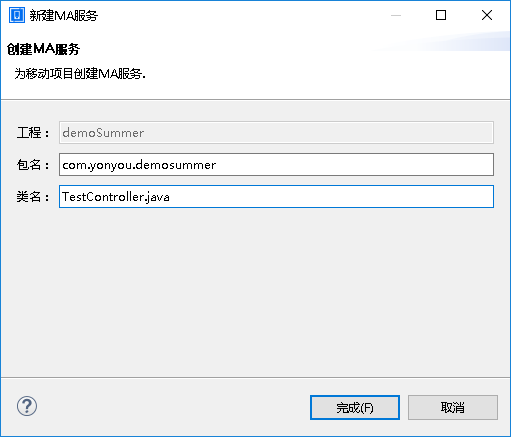
可以看出，开发工具自动会创建一个简单的工程模板，包含了html、css、js以及一些配置文件、资源文件等，至此，移动应用端的工程创建完毕。Summer工程详细的开发说明请参考开发工具的相关文档，这里不逐一赘述。

## 2.2新建MA服务

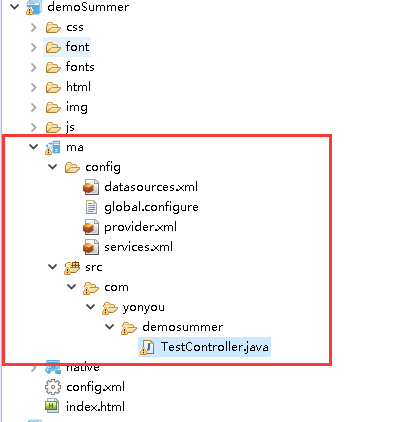
接下来我们来进行MA端的开发。首先鼠标点击选中工程，在鼠标右键弹出的菜单中依次点击选中【新建】下的【MA服务】，如下图所示：



点击【MA服务】后，弹出【新建MA服务】对话框，输入【类名】和【包名】，本例子中【包名】修改为“com.yonyou.demosummer”，【类名】修改为“TestController”，如下图所示：



点击【完成】按钮，MA服务创建完毕。此时工程出现目录结构会出现【ma】，展开如下图所示：



其中TestController.java是根据在创建MA服务器导中填写的包名和类名自动生成的MA Controoler的代码文件。打开java文件，可以看到包名和类名，这说明开发工具辅助我们完成了MA服务的创建。

接下来编写MA Controller的代码的过程在后面编写MAController代码一节进行讲解。

# 编写代码

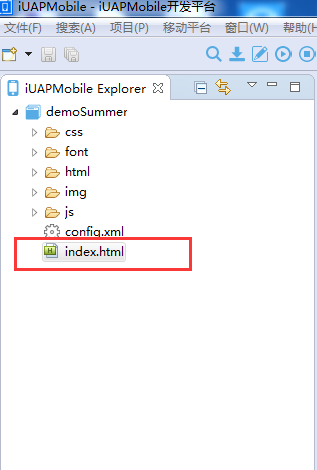
移动应用开发代码编写总的来说分为两部分：

* APP前端的代码编写
* MA服务端代码编写，即MA Controller代码编写

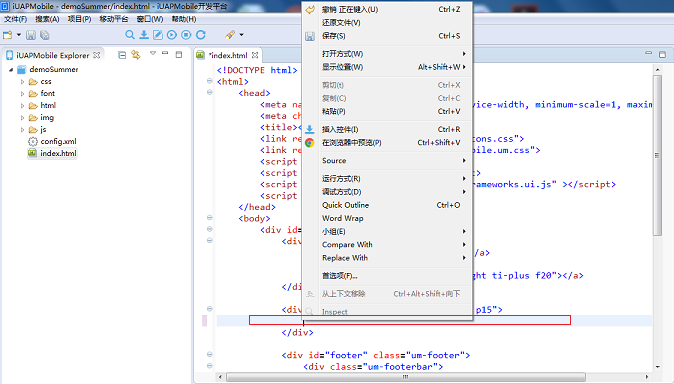
## 3.1APP端代码编写

### 3.1.1 编写index.html

在创建好的工程目录里，双击打开工程的index.html，位置如图：



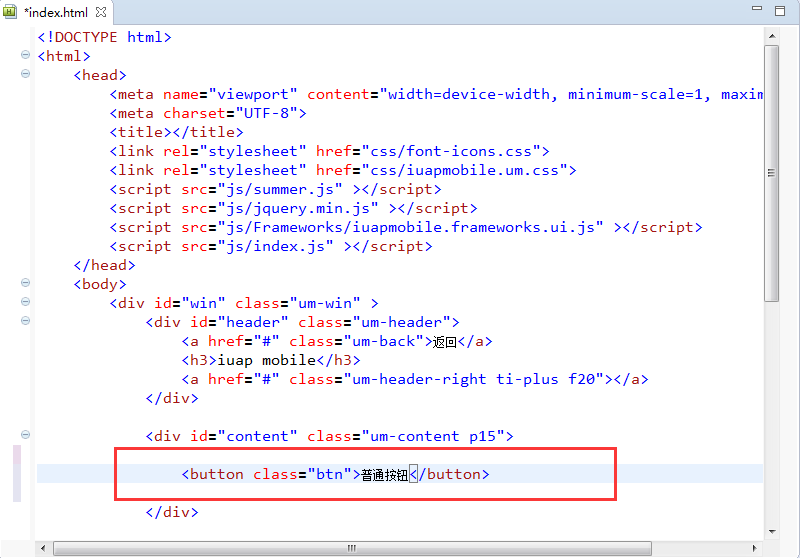
打开index.html后，在如下代码区域鼠标右键，菜单中选择【插入控件】，弹出【插入控件】页面。



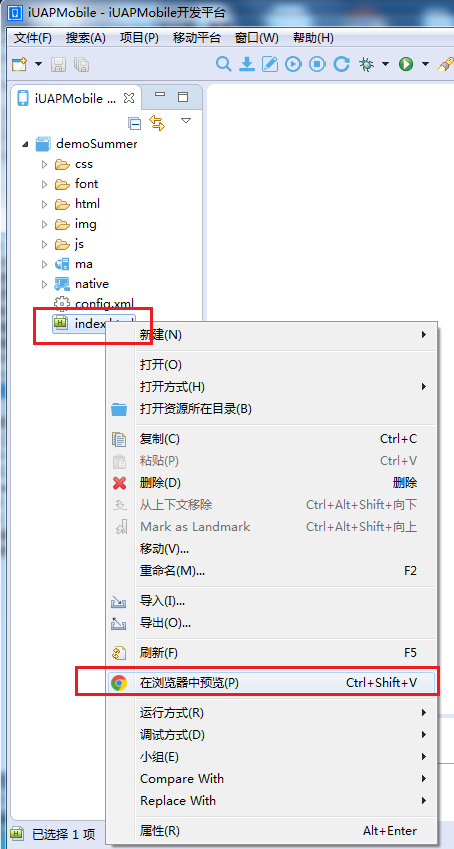
在弹出的页面里面左侧选择【按钮】，选中【普通按钮】，单击【确定】，如下图所示：



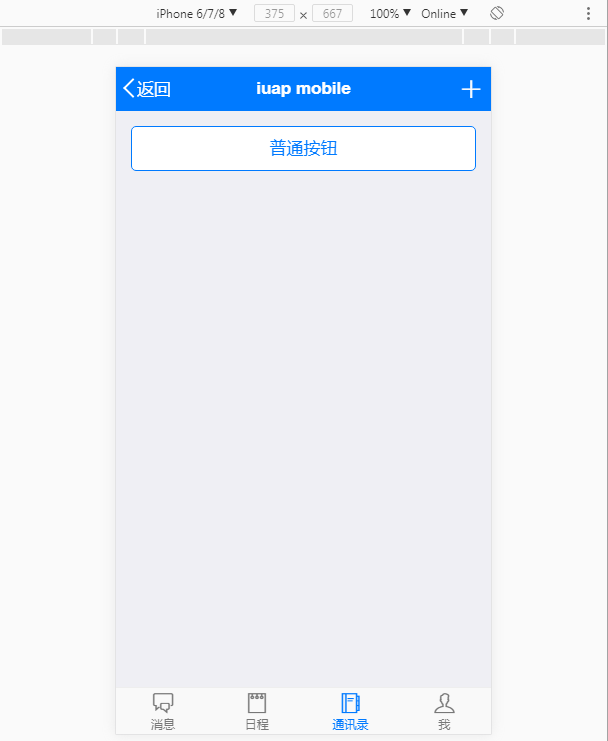
在代码编辑器会自动出现如下代码：



此时可以在浏览器预览一下当前HTML。具体操作为：鼠标单击选中index.html，在鼠标右键弹出的菜单中，点击【在浏览器中预览】，如下图所示：

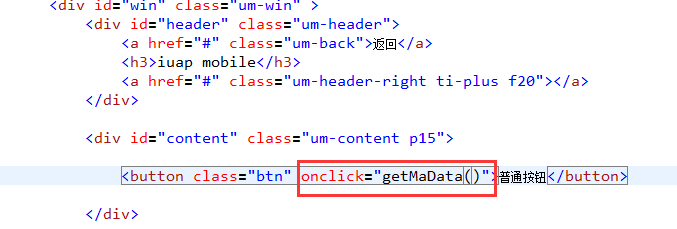


开发工具会自动启动Chrome浏览器（前提是安装Chrome浏览器）来加载index.html，将Chrome浏览器切换至开发者模式（按F12），选择合适的模拟器来预览页面，这里的详细介绍可以参考开发工具summer开发的相关文档，如下图所示：



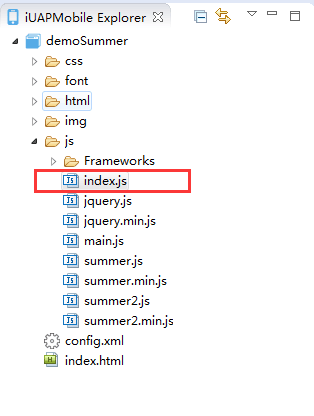
如果能正确在浏览器里显示index.html页面，并已经存在我们插入的【普通按钮】元素，说明我们的操作正确无误。如果不能正确显示，可以借助于浏览器来辅助我们排查错误。

接下来，我们进行点击【普通按钮】来实现我们请求MA的代码逻辑，在index.html中的<button>标签中增加onclick事件为getMaData()，如下图所示：

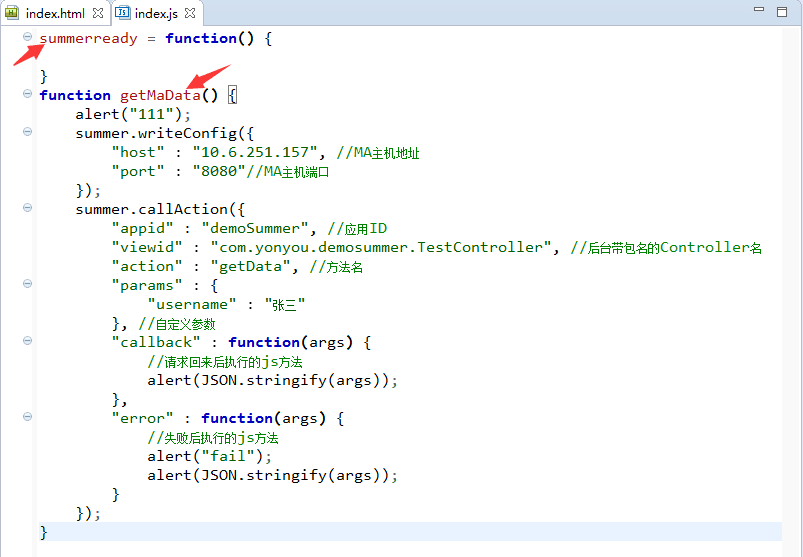


### 3.1.2 编写index.js

（1）在工程里双击打开index.js，位置如图：



（2）清空index.js中的summerready里的所有代码（务必删除），增加js方法 getMaData（）并保存，如下图所示：



getMaData方法的具体代码如下

|  |
| --- |
| **function** getMaData**()** {  alert("开始调用ma");  summer.writeConfig({  "host" :"10.6.251.157", //MA主机地址  "port" :"8080"//MA主机端口  });  summer.callAction({  "appid" :"demoSummer", //应用ID  "viewid" :"com.yonyou.demosummer.TestController", //后台带包名的Controller名  "action" :"getData", //方法名  "params" :{  "username" :"张三"  }, //自定义参数  "callback" :**function(**args**)** {  //请求回来后执行的js方法  alert(JSON.stringify(args));  },  "error" :**function(**args**)** {  //失败后执行的js方法  alert("fail");  alert(JSON.stringify(args));  }  });  } |

特别注意，注意，注意（重要的事情说三遍），summer的writeConfig方法的参数host和port一定是你本地MA的地址，一定确保手机能访问到，否则手机无法访问MA，关键代码如下：

summer.writeConfig({

"host" :"10.6.251.157", //MA主机地址,手机一定要能访问到

"port" :"8080"//MA主机端口

});

注：

其中callAction方法中的参数含义如下：

appid，默认为项目名字(如果项目名称没有被再次修改)，是app的唯一ID，对应在config.xml中的app\_name；

viewid为ma处理请求的后端controller类，即带包名的类名；

action为controller中处理请求的具体方法的名字。

以上完成了移动app端的调用MA的代码。接下来我们开发编写MA端的代码。

## 3.2编写MA Controller代码

双击工程目录下MA下的TestController.java代码，在代码编辑器进行代码编写如下代码后保存。代码如下：

|  |
| --- |
| **package**com.yonyou.demosummer;  **import**org.json.JSONObject;  **publicclass**TestController {  **public** String getData(String args) **throws** Exception {  JSONObjectjson = **new**JSONObject(args);  JSONObjectjsonResult = **new**JSONObject();  jsonResult.put("status", 1);  jsonResult.put("msg", "hello，" + json.optString("username"));  **return**jsonResult.toString();  }  } |

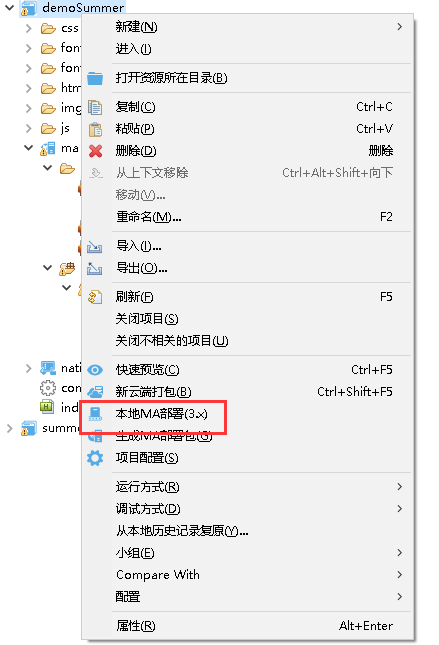
本例子中我们实现简单的MA服务的编写。

# 开发工具启动MA

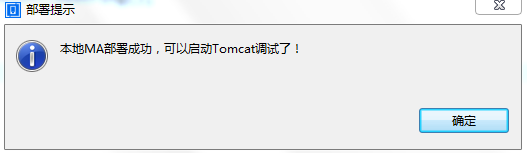
MA服务的代码编写完毕后，我们开始启动运行MA，共需要三步来完成。

## 4.1第一步、本地MA部署

操作过程：鼠标点击选中所在工程，在鼠标右键弹出菜单中选择【本地MA部署】如下图所示：



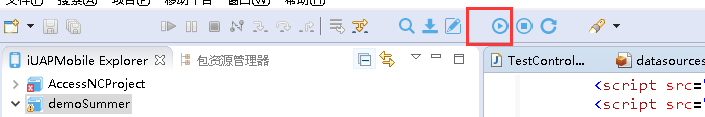
会弹出本地MA部署成功的弹出确认框，如下图所示：



当开发工具弹出上图部署成功提示后，说明本地MA服务已经部署成功。点击【确认】即可。接下来就可以启动MA Server了。

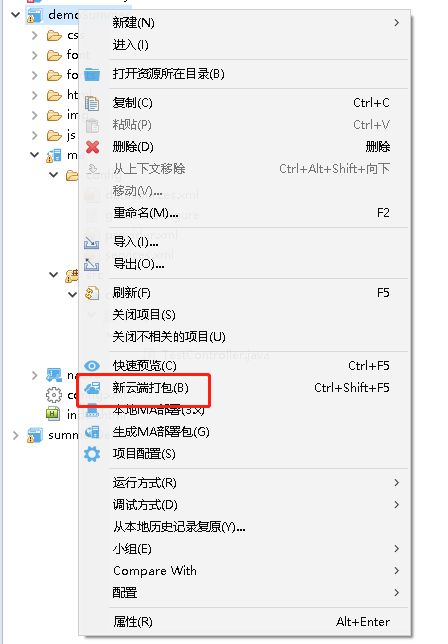
## 4.2第二步、启动MA Server

单击工具栏中按钮可以启动MA Server。如下图所示：



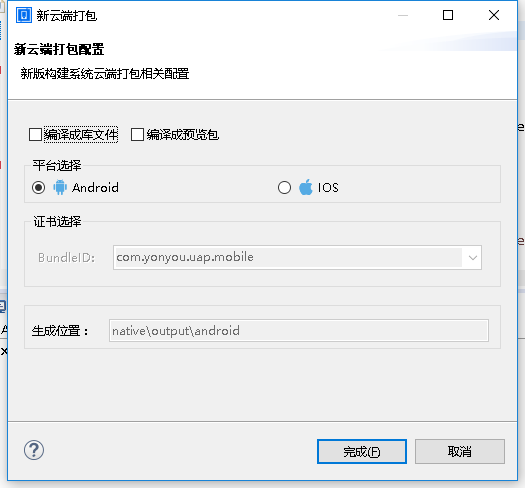
## 4.3 summer工程编译打包运行至手机（或模拟器）

1、右键summer项目，选择【新云端打包】

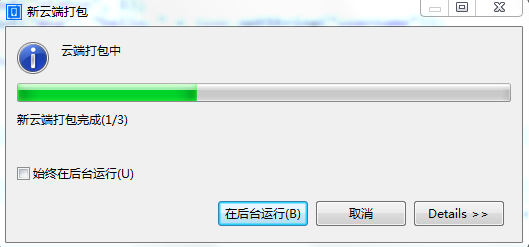


1. 新云端打包配置

默认选择Android平台，本例也以Android打包为例。



点击【完成】，会自动弹出打包过程页面，如下所示：



提示：不想每次打包的时候都弹出这个等待框，可以点击【在后台运行】。下次打包的时候便不会再弹出这个等待框了。

1. 生成二维码，扫描下载app

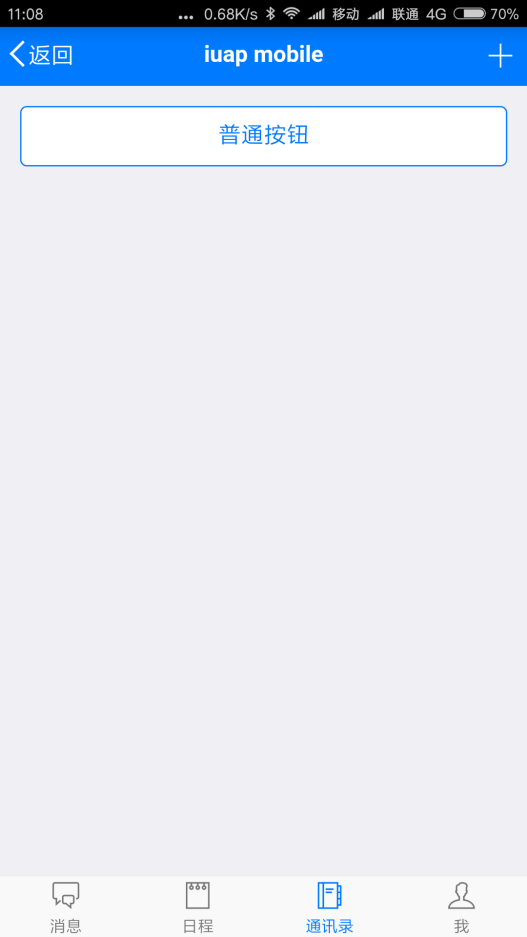
构建服务器构建完毕后，studio会自动弹出打包成功后的二维码，如下所示：



用手机扫描二维码安装app，或者根据二维码对应的地址下载应用包（安卓适用）后使用模拟器安装。

1. 验证结果

启动app后看到如下所示：



单击【普通按钮】。出现alert信息，则代表代码已经运行到例子index.js的getMaData方法中alert处。



点击手机上的alert确定，后app会自动弹出另一个alert，如下图所示：



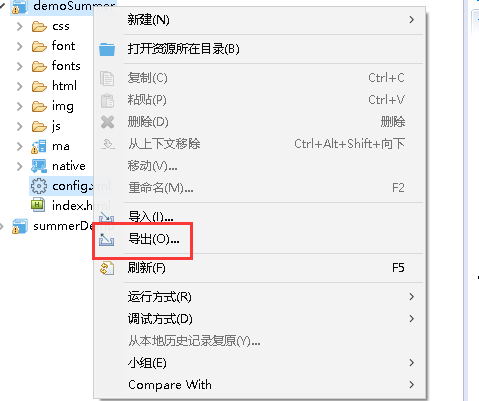
此时alert弹出的是ma返回app的数据，这些数据就是我们MA Controller里的代码。开发者也可以重复以上步骤，修改js和controller代码，改写成自己需要的数据，再次运行结果来验证自己的代码运行情况。

# 部署

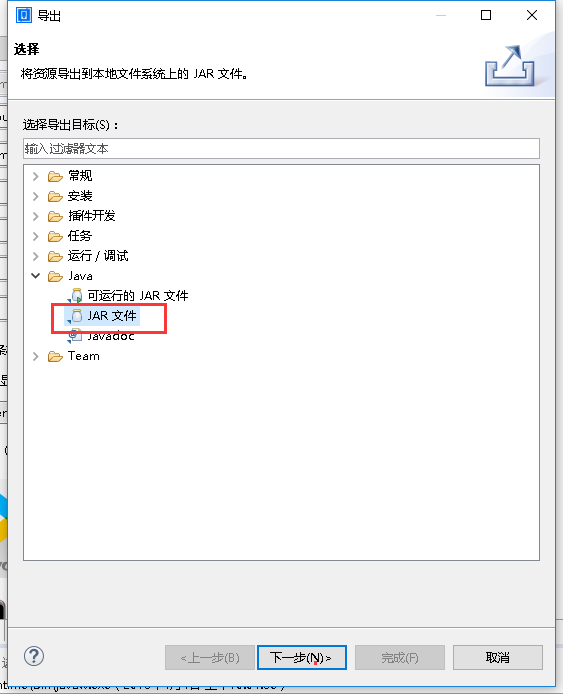
在生产环境下，请按照以下步骤部署mp包：

1. 利用开发工具生成mp包

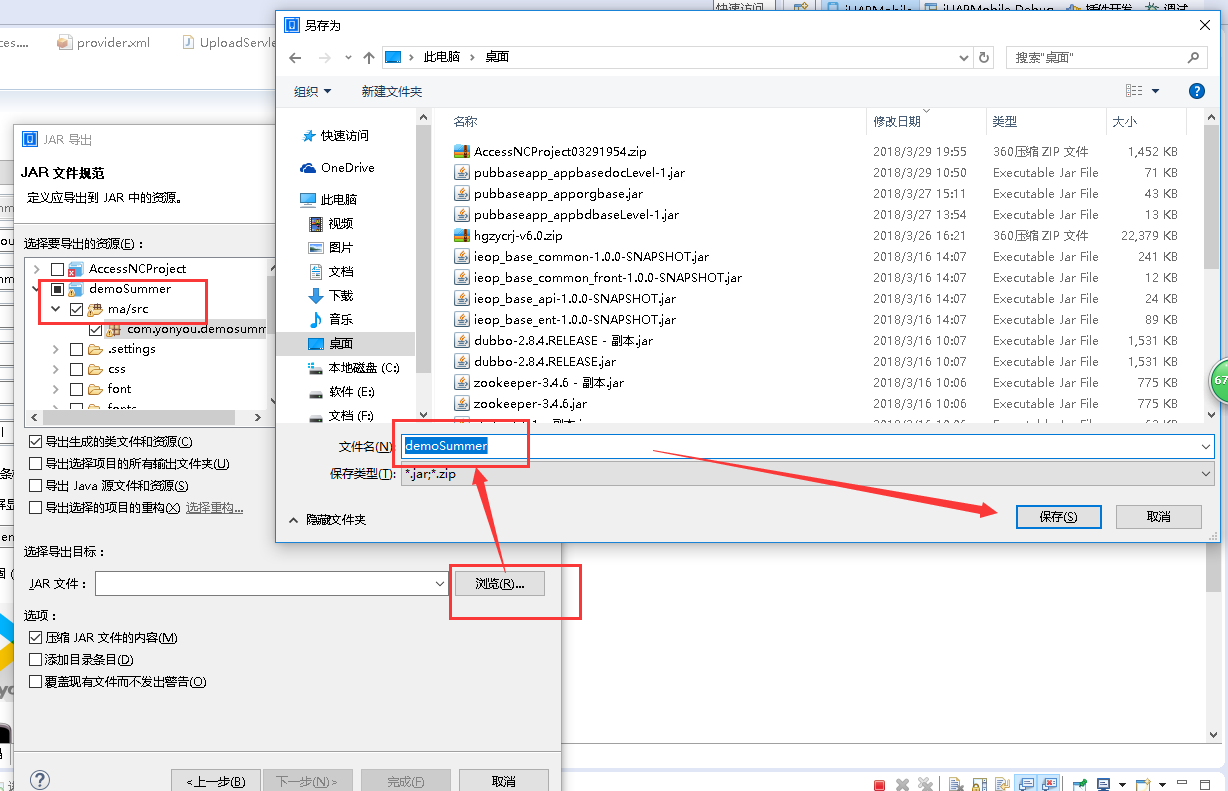
邮件项目，在菜单中选择【导出】



选择Java-JAR文件，单击【下一步】

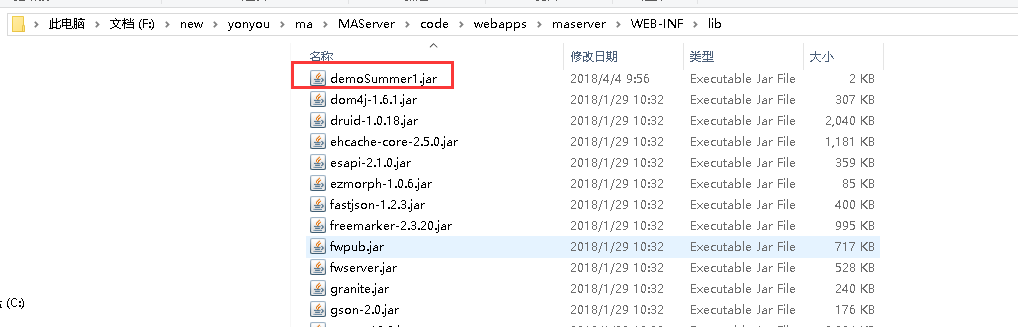


选择导出的工程，单击【浏览】，保存到本地。



2、将jar包放到

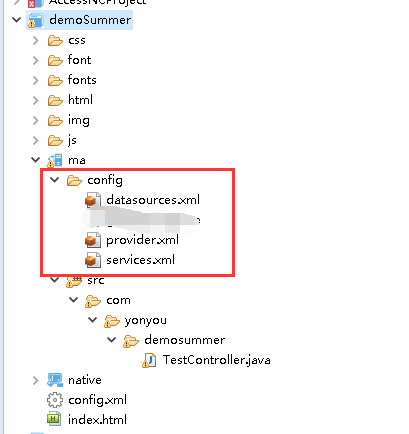
F:\new\yonyou\ma\MAServer\code\webapps\maserver\WEB-INF\lib位置下。



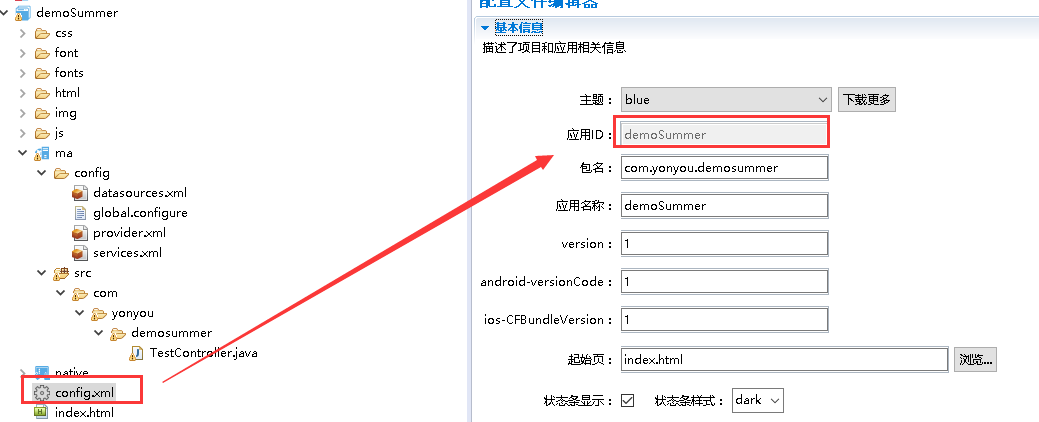
1. 将项目中的xml拷贝到

F:\new\yonyou\ma\MAServer\code\webapps\maserver\WEB-INF\classes\conf\configure\demoSummer文件夹下。

具体xml为：



注：标红的目录需要新建，名字为应用的ID，如图：

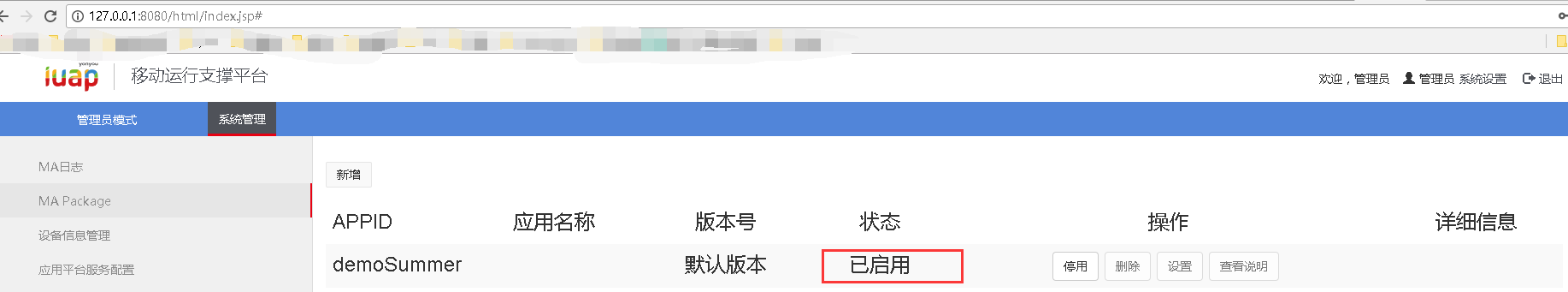


1. 启动ma server

执行F:\new\yonyou\ma\MAServer\code\bin\startup.bat

1. 验证

访问ma 后台管理页面http://ip:port/ 默认账号密码：admin/admin

如果默认包已启动则mp包部署成功。  


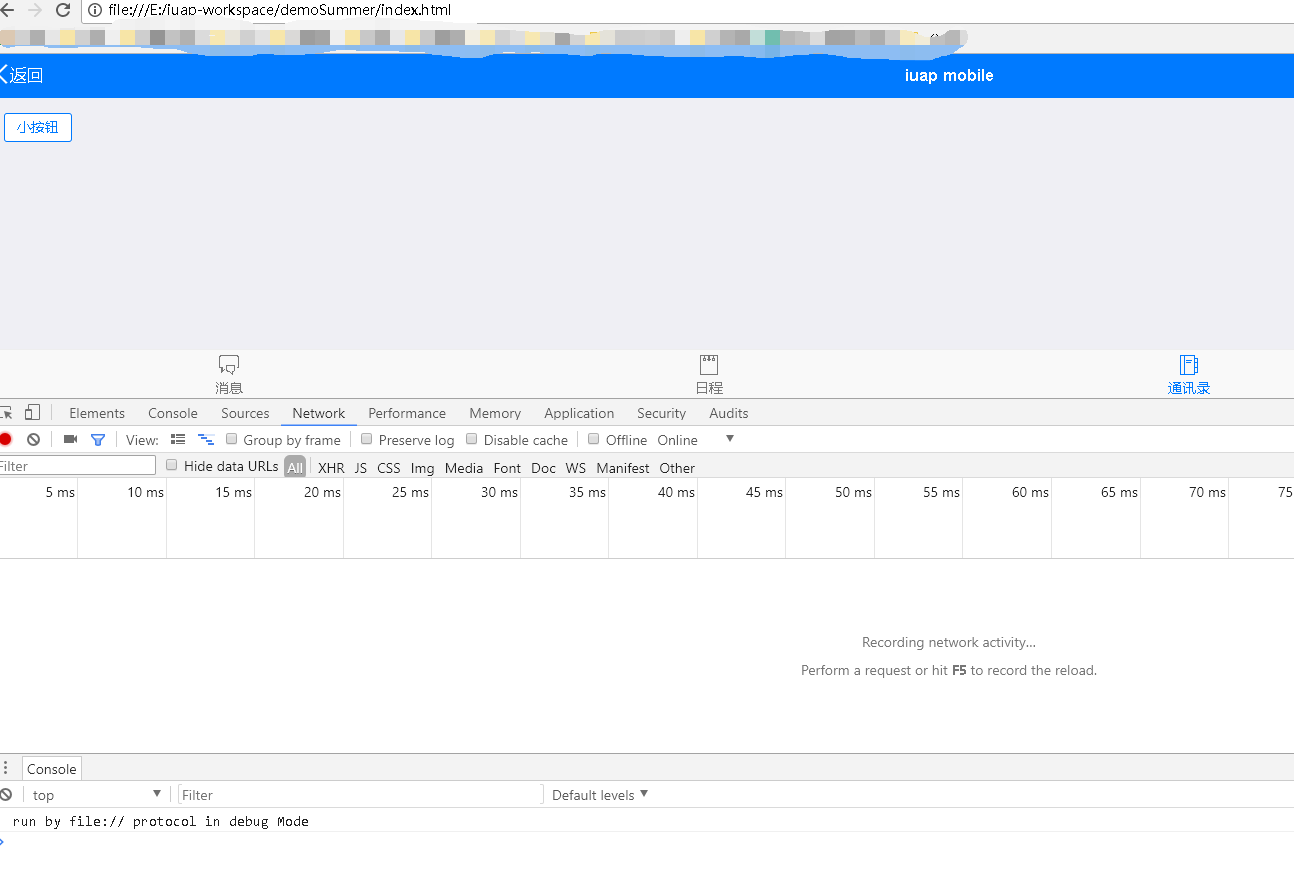
# 调试

不管是调没调用MA服务，如果出现错误，建议都从前端调试。最好在运行前预览界面，看是否有错。

## 5.1前端JS 调试

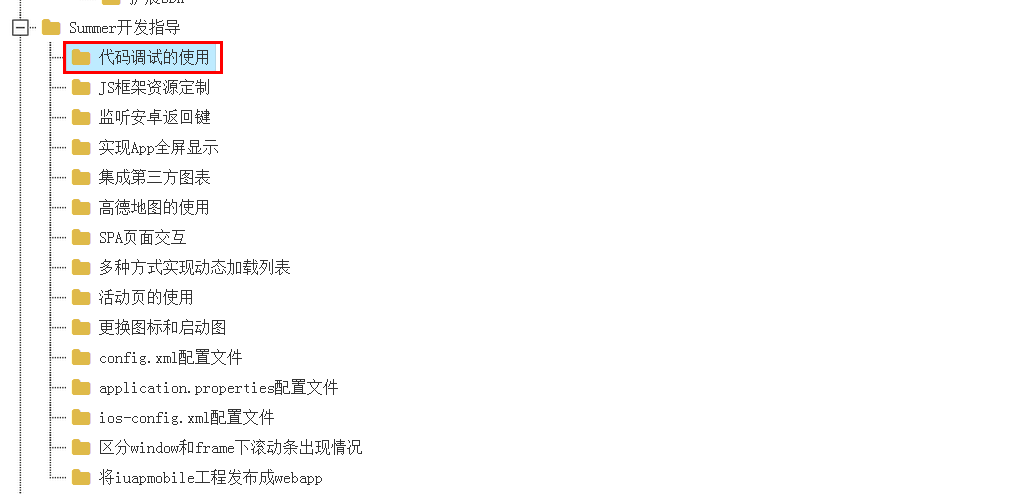
1. Chrome预览调试

预览，在Chrome浏览器中按F12键，显示如下图，根据相关提示解决问题。



1. Chrome inspect调试

在官网如下图所示处有介绍



## 5.2服务器端调试

1. 调试方法见如下连接

<http://www.csdn.net/article/2012-09-03/2809495-Java-Debugging-Tips-with-Eclipse>