

---

# 电信物联网平台对接教程

## 目录

- 一.准备工作..... 2
  - 1.1 电信物联网平台账号申请..... 2
  - 1.2 https 级别 IP..... 2
  - 1.3 获取 APPID 以及密钥..... 2
- 二、程序对接..... 3
  - 2.1 鉴权对接..... 3
  - 2.2 订阅服务通知..... 3
  - 2.2 创建设备下发命令..... 4
  - 2.3 接收电信物联网平台推送..... 4
  - 2.4 注意事项..... 4

---

# 一.准备工作

## 1.1 电信物联网平台账号申请

在进行与电信平台的对接之前需要在以下的网站中申请物联网开发者平台的账号，申请环节将不做赘述。

超链接：

<https://develop.local.ct10649.com:8843/login?service=https%3A%2F%2Fdevelop.ct10649.com%3A8093%2F#/developer-overview>

## 1.2 https 级别 IP

在获取电信物联网开发者平台帐号之后，需要准备一个拥有 https 级 ip 的服务器，这里我使用的是阿里云的云服务器，如果使用非 https 级别 ip 即需要与电信签署白名单协议。

## 1.3 获取 APPID 以及密钥

在电信平台中需要保存 APPID 以及其密钥，以下是一个获取的例子(不可用)。

APPID OB1tsCOc05HBFgLXXXxxxQTYnywa ZPizGJWDmdm87CXXXxxxflLzLjRsa
--

## 二、程序对接

### 2.1 鉴权对接

查看 java 程序中的 Authentication.java 文件中代码，修改部分 Constant.java 中的参数并使用使用初次登陆 FirstLogin 函数。

对接电信云平台登陆成功之后需要进行周期性的鉴权刷新，此处建议使用 Spring 框架中的 Scheduled 写定时刷新。详细代码可参考 src/com/util 文件下的 timingService.java 文件中的 RefreshToken 函数，并且需要将刷新生成的 accessToken 更新至 Constant.java 中(此处做一个解释，在进行初次对接的时候 Authentication.java 中已经进行了该静态数据的生成与赋值)。

如下图为定时刷新鉴权效果图：



```
管理员: C:\Windows\System32\cmd.exe - java -jar demo-0.0.1-SNAPSHOT.jar
RefreshToken: accessToken: e08d61ee37f57f9394b872f1756963
RefreshToken: accessToken: 83cd3f52b873f9193dab6715b9744930
RefreshToken: accessToken: 7db13d61411ebb4ebcea6229a8b1fd11
RefreshToken: accessToken: 9c8f6f3ee4b72374decf30aafe4c21ca
RefreshToken: accessToken: 783e9b16bea012938061d8f9511ceaab
RefreshToken: accessToken: a0dec3647bcb9c1265addf36be6392
RefreshToken: accessToken: ac6a41528456d5c3c13a4b4f8647eb5
RefreshToken: accessToken: 2fcf6a3a55f7cfeb6e5665c23f0a0bb
RefreshToken: accessToken: faeda9a2bfe183af3f3d91381e4f75
RefreshToken: accessToken: a177aae70feaa15de3ff2b2a4a9c33
RefreshToken: accessToken: 6990ab805ef254a6e3b57fde523f77f
RefreshToken: accessToken: 1a985c89287fcc34ba6e26bbfec59b3f
RefreshToken: accessToken: 68c146b859e1e2f1e5ed5f206bf5a834
RefreshToken: accessToken: 22eddd235d7bdec79122e872a5e8ab0
RefreshToken: accessToken: 8535c128190e7863d753b96176a78ea
RefreshToken: accessToken: 374abb8b5fd3ec1ef481e45bd84393c
RefreshToken: accessToken: 2f6da0bcb5bdf54dca6dec338a0c224
RefreshToken: accessToken: 37d378ceb139b5b55ca7592138e6694
RefreshToken: accessToken: 73ccc554f8ef39defcf49578ccff3c3
RefreshToken: accessToken: 82c3ae05ae0fd7cfd9be72e275285d
RefreshToken: accessToken: 8c49559bcabee118ccd4e2f01b5428
RefreshToken: accessToken: 3952b890b52d2754108860678775a63c
RefreshToken: accessToken: f7865279b93e299cd64a65846dc13d2b
RefreshToken: accessToken: 627e9c2db8c02753fc6bc36e575abbbb9
```

### 2.2 订阅服务通知

在接收到物联网平台的推送订阅服务通知之前需要准备 https 的 ip 以及需要使用 src\main\java\com\IOT\service\subscriptionManagement 文件夹中的 SubscribeServiceNotification.java 文件，在该文件只需要运行一次即可，并且可以在电信物联网平台观察到这个订阅服务已经生效。

使用教程：使用 idea 或者 eclipse 编辑器的 CTRL+ 点击方式进入 SubscribeServiceNotification.java 中的 Constant.java 中的参数，详细的参数在注释中已经提示。修改完毕只有点击运行即可，不需要进行任何操作，此时你可以看

---

见在电信平台的订阅服务中看见我们刚刚修改的订阅服务，例如设备状态改变：  
devcieDataChanges:https:192.168.1.1:8080/api/devcieDataChanges。

## 2.2 创建设备下发命令

创建设备下发命令需要在 java 工程中的 resources 文件夹下放置 cert/ca.jks 以及 outgoing.CertwithKey.pkcs12 两个文件，例如在 TaskController.Java 文件中 CreateDeviceCommand 函数进行我们命令的下发，此时我们需要传入若干参数，但是必须注意的是，在进行设备下发的时候需要获取我们上面刷新到的 Constant.accessToken 以免造成下发失败。详细可以根据硬件厂商定制的 profile 文件进行命令的定制

## 2.3 接收电信物联网平台推送

接收电信物联网平台的推送之前确保消息服务已经订阅成功，使用电信平台时候需要使用 Spring 框架中的 RESTful 接口进行推送消息的接收，如以下代码所示。

```
//电信传回的数据
@RequestMapping(value = "/IoT/updateDeviceData", method = RequestMethod.POST,
produces = MediaType.APPLICATION_JSON_VALUE)
public static final String DEVICE_DATA_CHANGED_CALLBACK_URL = CALLBACK_BASE_URL
+ "/IoT/updateDeviceData";
```

从上述代码我们可以通过看出，上面为我们自定义的电信接口，下面为推送订阅服务中设定的接口，这里的联动性需要注意。

其余的接口大同小异，逻辑不做过多的赘述。可以参考 src\main\java\com\IOT\IOTController 中的接口进行修改。

## 2.4 注意事项

在与电信物联网平台的对接过程中需要注意以下几点：

1. 项目运行开始时需要进行鉴权，并且需要进行周期刷新鉴权
2. 鉴权获取的 accessToken 可以使用静态变量进行数据的存取
3. 下发需要获取上述的鉴权信息，并且需要参考硬件提供商的 profile 文件

- 
4. 消息订阅以及消息推送具有整体性，两者关系紧密
  5. 当单设备推送消息较多可以使用组推送
  6. 一次性数据可以使用 redis 技术，提高程序运行效率
  7. 使用 SpringBoot 框架请多借鉴本程序尤其是 application.yml 与 pom.xml 中配置
  8. Spring boot 打包的 jar 与直接在运行程序是不一样的，尤其注意 cert 文件夹(在 demo 中我在 src 中也保留了一份)
  9. 需要注意跨域，可以直接仿照本 demo 进行跨域
  10. 本 demo 使用 spring-boot+mybatis+mysql 主要 jar 都在 pom.xml 文件中体现