### 1，工程的搭建

#### .环境搭建，准备好以下环境：

后端

java

redis

tomcat8

Maven

Idea

前端

ws

nodeJs

bower

#### 2).在命令提示符，输入命令，搭建一个新的项目:

Mvn org.apache.maven.plugins:maven-archetype-plugin:2.4:generate -D archetypeGroupId=com.hand -D archetypeArtifactId=hmap-webapp-archetype -D archetypeVersion=2.2-SNAPSHOT -D groupId=com.hand -D artifactId=HmapParent -D package=hmap.core -D archetypeRepository=http://mobile-app.hand-china.com/mavenrepo/nexus/content/repositories/snapshots

#### 3)进入项目根目录，在命令提示符输入以下命令:

mvn clean install

4)在mysql中创建数据库，然后在项目根目录输入以下命令，初始化数据库：</br>

mvn process-resources -D skipLiquibaseRun=false -D db.driver=com.mysql.jdbc.Driver -D db.url="jdbc:mysql://localhost:3306/hmaptest?useUnicode=true&characterEncoding=UTF-8" -D db.user=root -D db.password=root

#### 5)把项目导入ij编辑器中，在tomcat中设置jdni，然后启动即可

#### 前端

下载前端种子工程路径为：

<https://rdc.hand-china.com/gitlab/hmap-cloud-develop/hmap_front.git>

安装webstore，npm,bower

把种子工程导入到ws下，在根目录下执行以下命令

npm install

npm install -g gulp

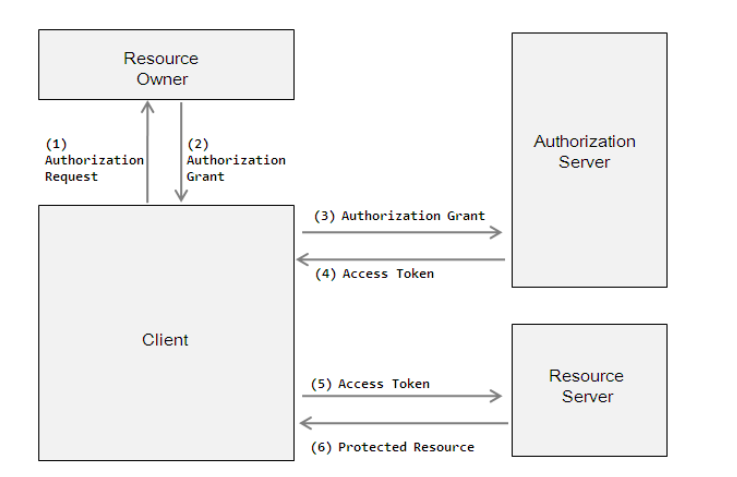
npm install -g bower

bower install

### /oauth/token接口的开发

#### 1）流程

如图1所示，协议的基本流程如下：



(1) Client请求RO的授权，请求中一般包含：要访问的资源路径，操作类型，Client的身份等信息。

(2) RO批准授权，并将“授权证据”发送给Client。至于RO如何批准，这个是协议之外的事情。典型的做法是，AS提供授权审批界面，让RO显式批准。这个可以参考下一节实例化分析中的描述。

(3) Client向AS请求“访问令牌(Access Token)”。此时，Client需向AS提供RO的“授权证据”，以及Client自己身份的凭证。

(4) AS验证通过后，向Client返回“访问令牌”。访问令牌也有多种类型，若为bearer类型，那么谁持有访问令牌，谁就能访问资源。

(5) Client携带“访问令牌”访问RS上的资源。在令牌的有效期内，Client可以多次携带令牌去访问资源。

(6) RS验证令牌的有效性，比如是否伪造、是否越权、是否过期，验证通过后，才能提供服务

#### 2）密码模式

http请求方式: POST <http://ip:port/hms/oauth/token?client_id=AppID&client_secret=AppSecret&grant_type=password&username=username&password=password>

这里的username和password可以是第三方接口也可以是中台页面的登录接口所传的用户名和密码所提供的。

以app接口为列，移动前端第一次登录的时候会携带username和password到中台获取接口凭证token

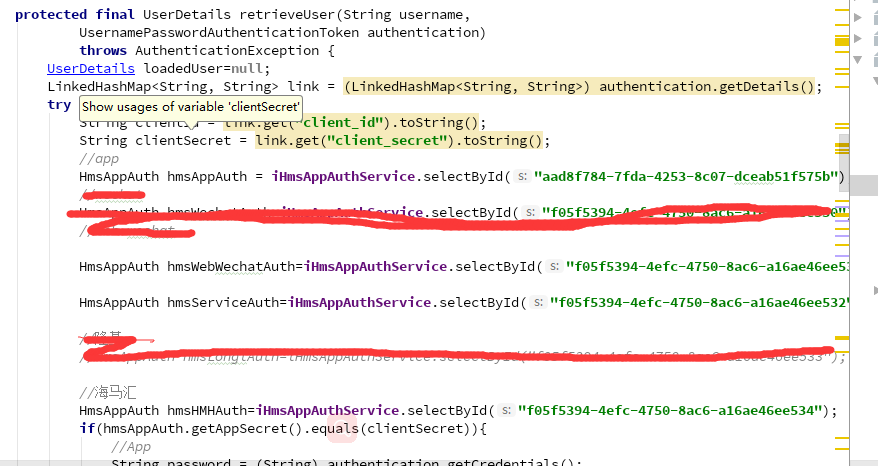
然后用token去调用中台开发的接口。

\*\*参数说明 \*\*

| **参数** | **是否必须** | **说明** |
| --- | --- | --- |
| grant\_type | 是 | password |
| client\_id | 是 | 应用唯一凭证 |
| client\_secret | 是 | 应用唯一凭证密钥 |
| username | 是 | 用户登录名 |
| password | 是 | 用户登录密码 |

##### 中台密码模式的实现

当移动端调用中台的/oauth/token接口的时候，会经过投票器等一系列的验证机制，一步一步的下来，中间会经过provide类，为了满足业务验证，此时我们必须重写provide，在里面去调用第三方的验证接口去验证是否app传来username和password是否正确，如果正确就把username和password放到UserDatail里面，这样验证机制就完成了。框架自动的返回一个token



#### 3）客户端模式

http请求方式: POST

<http://ip:port/hms/oauth/token?client_id=AppID&client_secret=AppSecret&grant_type=client_credentials>

不需要传用户名和密码，一般用于中台的一些配置，比如job定时机制

#### 4）授权码模式:authorization\_code

http请求方式: POST

<http://ip:port/hms/oauth/authorize?client_id=AppID&redirect_uri=YourUrl&response_type=code&scope=read>



移动前端调用这个接口返回一个授权码，返回授权码以后调用中台服务器的接口，中台根据前端带过来的授权码去调用海马汇的接口获取提前导入的用户信息，中台拿到用户信息以后调用第三方的接口验证是否通过/oauth/token中的验证机制从而获取token

格式:

使用获得的授权码向中台请求获得token

http请求方式: POST

http://ip:port/hms/oauth/token?client\_id=AppID&client\_secret=AppSecret&grant\_type=authorization\_code&redirect\_uri=YourUrl&code=code

请求参数:

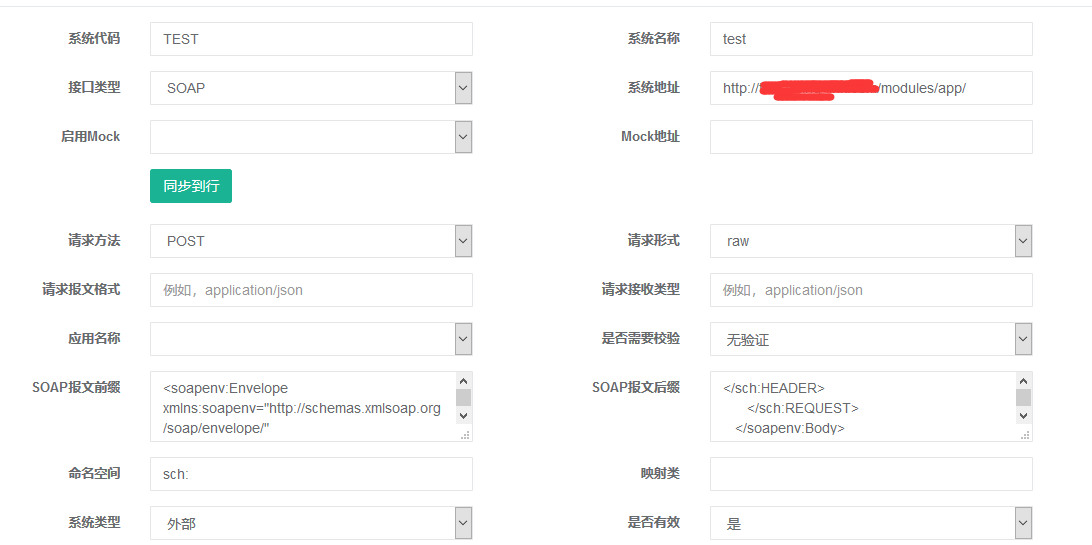


登录接口重写:http://eco.hand-china.com/doc/hmap-cloud/latest/part3/chapter9.html

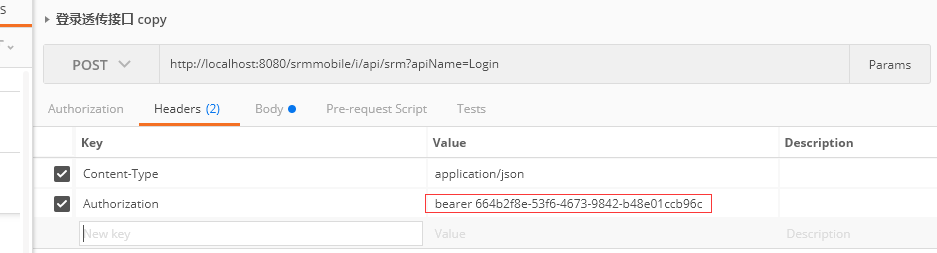
### 3,透传接口的开发

中台与第三方交互的时候用的数据格式是xml格式的，而移动前端传过来的数据是json格式的。所以需要中台转一下

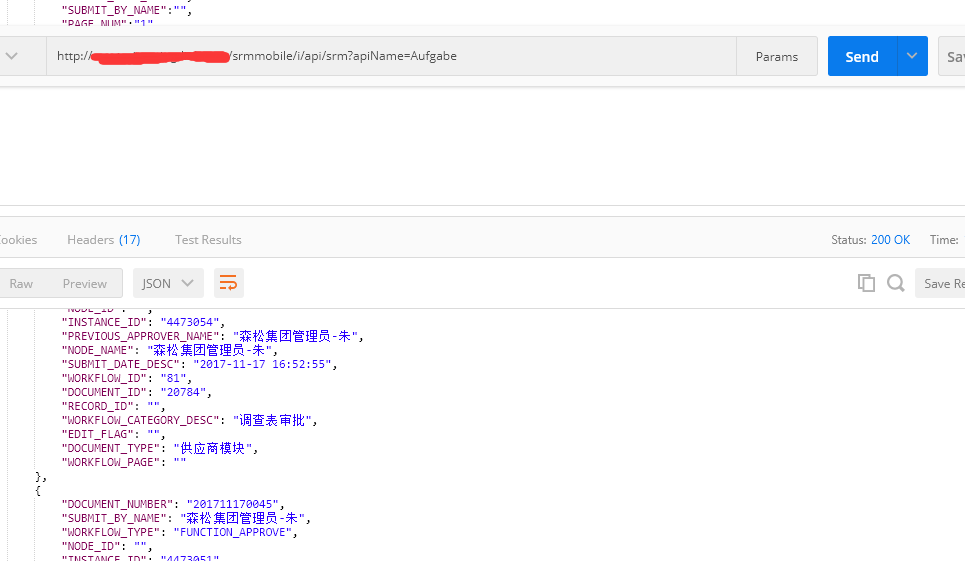
中台配置soap格式



然后通过http请求把第三方的数据带过来



与前端的交互如上图所示，就是前端把接口凭证token和请求参数传过来，即可，前端可以不传用户的账号和密码过来，因为token里面已经携带了这些信息



### 4,常用的海马汇功能

hmap-plugin-contact 组织架构插件  
hmap-plugin-wxqy 微信企业号插件  
hmap-plugin-wxfw 微信服务号插件  
hmap-plugin-appcenter 应用中心插件  
hmap-plugin-workflow 工作流插件  
hmap-plugin-article 企业频道  
hmap-plugin-statistics 统计分析插件  
hmap-plugin-rongyun 融云插件  
hmap-plugin-register 校验插件(二次校验,短信校验)  
hmap-plugin-message 消息中心插件  
hmap-plugin-job job平台插件  
hmap-plugin-feedback 服务插件  
hmap-plugin-edition 子应用、版本管理插件  
hmap-plugin-device 设备管理插件  
hmap-plugin-banner 轮播图、启动页插件  
hmap-plugin-api 接口透传、接口降级插件  
hmap-basic 基础功能项目(必选)

以上功能都是海马汇框架写好的功能，我们只需要在pom.xml中导入相关的maven包，在中台页面上面配置对应的参数，然后直接调用接口就可以实现

### 5,与前端的交互

#### 前端框架

Ionic+vuejs

#### 概要

需要知道子应用，在PC工作台如何上传，以及如何设置。

子应用目前可以支持在线H5，即将你的H5页面放到web容器中（tomcat、weblogic、nginx等），将你的H5访问链接配置到子应用上；也支持将客户已经有的H5页面链接配置成一个子应用。

子应用目前也支持离线H5，即将你的H5页面上传到服务器上。用户使用你的子应用之前会要求先下载你的H5子应用到手机本地，再在本地运行H5代码。（推荐使用这种方式，体验要好很多，并且支持离线应用场景）

因为用户在使用APP过程中，进入到APP登录页面后，就已经登录了。用户点击进入子应用，不应该再让客户进行登录。所以需要每个子应用单点登录获取到当前用户的登录token。

因为离线H5是打包到APP中的，所以可以通过调用原生函数的方式获取到token

#### 在线H5获取token的方式

在线H5可以理解为额外开发一个单独的系统，对接到海马汇上（与第三方业务系统将H5页面挂到海马汇上是一个道理），所以需要在海马汇工作台上为系统创建第三方账号，获得设置生成的client\_id和client\_secret(secret需要在调用解析code的接口时使用)。在设置子应用跳转链接时，需要为子应用构造如下的链接： <http://ip:port/hms/oauth/authorize?client_id=AppID&redirect_uri=YourUrl&response_type=code&scope=read> ，请求发起后会直接跳转到YourUrl，并在最后追加code，形如YourUrl?code=xxxxx。即员工点击后，页面将跳转至 YourUrl?code=xxxxx，在你的链接中需要接收产生的code授权码，并且调用授权码解析接口，获得token。下面的例子是如何在你开发的vue2.0页面中，调用授权码解析接口获得token。

配置链接类似:http://wechat.hand-china.com/hmapshowcase220/oauth/authorize?

client\_id=xxxxxx&

client\_secret=xxxxxx&

grant\_type=authorization\_code&

response\_type=code&redirect\_uri=http://wechat.hand-china.com/hmap\_doc/

其中client\_id,client\_secret在企业信息中查看,redirect\_uri即为要跳转的链接

跳转后链接类似:http://wechat.hand-china.com/hmap\_doc/?code=xxxxxx

使用code请求子应用服务端(第三方系统)请求token

第三方系统根据code调用海马汇用户信息接口获取用户信息,根据用户信息匹配到子应用服务端(第三方系统)的token

#### 离线H5获取token的方式

· 用户下海马汇APP下载并且打开离线应用

· 子应用通过桥接获取到海马汇提供的token

· 使用token请求子应用服务端(第三方系统)请求token

· 第三方系统根据token调用海马汇用户信息接口获取用户信息,根据用户信息匹配到子应用服务端(第三方系统)的token

·

Vue2.0开发

Vue是新一代的轻量级的JavaScript前端框架，注重渲染性能，虚拟Dom，MVVM架构以及令人开发畅快的Web Component，非常适合海马汇的需求和前端快速渲染的需求。下图为第三方的js前端框架性能对比，轻量和性能的优势使vueJS特别适合做一些轻量级应用的场景和页面的开发。

ionic1.x开发

Ionic是一个兼容性，稳定性，性能非常好的前端框架；

Ionic 具有Angular的所有JavaScript代码设计的优秀特性，易读性，扩展性很强；

Ionic提供了漂亮的UI，以及酷炫的CSS3动画

### 6，代码展示