|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | |  |
|  | | |
|  | | |
| 方圆ok1 | | | |
|  | **SVA对接一本通** | | 附件1-16K |
|  | |
| **文档版本** | 01 |
| **发布日期** | **2018-04-15** |
| **华为技术有限公司** | |

|  |
| --- |
| 版权所有 © 华为技术有限公司2018。 保留一切权利。  非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。  商标声明  附件3-版权声明页图和其他华为商标均为华为技术有限公司的商标。  本文档提及的其他所有商标或注册商标，由各自的所有人拥有。  注意  您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或默示的声明或保证。  由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。 |

|  |  |
| --- | --- |
| 华为技术有限公司 | |
| 地址： | 深圳市龙岗区坂田华为总部办公楼 邮编：518129 |
| 网址： | <http://www.huawei.com> |
| 客户服务邮箱： | [support@huawei.com](mailto:support@huawei.com) |
| 客户服务电话： | 4008302118 |

目 录

[1 Service Anchor 对接一本通 1](#_Toc511832369)

[1.1 《Service Anchor 对接一本通》变更说明 1](#_Toc511832370)

[1.2 介绍 2](#_Toc511832371)

[1.2.1 SVA 2](#_Toc511832372)

[1.2.2 API接口 3](#_Toc511832373)

[1.2.3 开发者社区 4](#_Toc511832374)

[1.2.3.2 源码托管 4](#_Toc511832375)

[1.2.3.3 开发指导 4](#_Toc511832376)

[1.2.3.3.2 环境要求（以windows系统为例） 4](#_Toc511832377)

[1.2.3.3.3 将工程导入eclipse 4](#_Toc511832378)

[1.2.3.4 Github的相关操作 17](#_Toc511832379)

[1.2.4 OpenLab 18](#_Toc511832380)

[1.2.4.1 访问方式 18](#_Toc511832381)

[1.2.5 ICS Tester 18](#_Toc511832382)

[1.3 SVA 对接指导 18](#_Toc511832383)

[1.3.1 对接整体流程 18](#_Toc511832384)

[1.3.1.1 证书文件获取 18](#_Toc511832385)

[1.3.2 APP开发 19](#_Toc511832386)

[1.3.2.1 AMQP 19](#_Toc511832387)

[1.3.2.1.1 支持的语言 19](#_Toc511832388)

[1.3.2.1.2 开源代码 19](#_Toc511832389)

[1.3.2.2 数字证书 19](#_Toc511832390)

[1.3.2.2.1 原理 19](#_Toc511832391)

[1.3.2.2.2 生成和转换指导 19](#_Toc511832392)

[1.3.2.3 对接实例代码及解说 20](#_Toc511832393)

[1.3.2.3.1 通过https协议与SVA建立连接 20](#_Toc511832394)

[1.3.2.3.2 发送用户名密码获取Token 20](#_Toc511832395)

[1.3.2.3.3 根据上一步获取的Token去订阅（匿名化/非匿名化） 22](#_Toc511832396)

[1.3.2.3.4 通过双向证书认证获取AMQP数据 23](#_Toc511832397)

[1.3.2.3.5 Java版amqp对接代码示例 24](#_Toc511832398)

[1.3.2.3.6 数据解析（json）并录入数据库 27](#_Toc511832399)

[1.3.2.4 常见问题FAQ 29](#_Toc511832400)

[1.3.2.4.1 终端如何获取本机IP 29](#_Toc511832401)

[1.3.2.4.2 与SVA对接不成功，如何定位问题 30](#_Toc511832402)

[1.3.2.4.3 如何获取Java版证书 30](#_Toc511832403)

[1.3.2.4.4 为什么对接成功后，过一段时间连接会自动断开 30](#_Toc511832404)

[1.3.3 APP与SVA对接测试 30](#_Toc511832405)

[1.3.3.1 用例 30](#_Toc511832406)

[1.3.3.2 环境 30](#_Toc511832407)

[1.3.3.3 传输组网 31](#_Toc511832408)

[1.3.3.4 操作指导 31](#_Toc511832409)

[1.3.3.4.1 启动工程，访问demo首页，如下图 31](#_Toc511832410)

[1.3.3.4.2 选择菜单-信息管理-商场管理 31](#_Toc511832411)

[1.3.3.4.3 选择菜单-信息管理-sva管理 33](#_Toc511832412)

[1.3.3.4.4 选择菜单-信息管理-地图管理 35](#_Toc511832413)

[1.3.3.4.5 选择菜单-客流分析-客流实时热力图 39](#_Toc511832414)

[1.3.3.5 问题定位方式 41](#_Toc511832415)

[1.3.3.5.1 问题一：热力图没有数据显示 41](#_Toc511832416)

[1.3.3.6 常见问题FAQ 41](#_Toc511832417)

[1.3.3.6.1 Tomcat启动失败，提示 address already in use! 41](#_Toc511832418)

[1.3.3.6.2 登陆页面正常显示，输入用户名密码后登陆，页面跳转至错误信息页 41](#_Toc511832419)

[1.3.3.6.3 在SVA管理页面添加SVA信息时，点击确认后没有反应 41](#_Toc511832420)

[1.3.3.6.4 在地图管理页面上传地图失败 41](#_Toc511832421)

[1.3.4 对接案例 42](#_Toc511832422)

[1.3.5 人员及联系方式 42](#_Toc511832423)

# Service Anchor 对接一本通

概述

本文档描述了如何用第三方APP对接华为Service Anchor产品。

产品版本

与本文档相对应的产品版本如下所示。

| 产品名称 | 产品版本 |
| --- | --- |
| SVA3100AE |  |
| SVA3200AE |

读者对象

本文档的读者是需要对接华为Service Anchor产品应用系统开发者，本文档的读者需要熟悉HTTPS、AMQP协议，并且具备数字证书认证和加密的基本知识。

## 《Service Anchor对接一本通》变更说明

这里描述了《Service Anchor对接一本通》各个版本的变更信息。

01 (2018-04-15)

该版本为01版本

| 内容 | 修改项 |
| --- | --- |
|  |  |
|  |  |

## 介绍

### SVA

随着无线通信的发展，全球移动数据流量增长迅速，其中90%的流量来自于室内（例如商场、机场、企业），室内场景下终端用户对无线信号质量、业务体验有更高诉求，同时室内无线网络环境相比室外也变得更复杂。

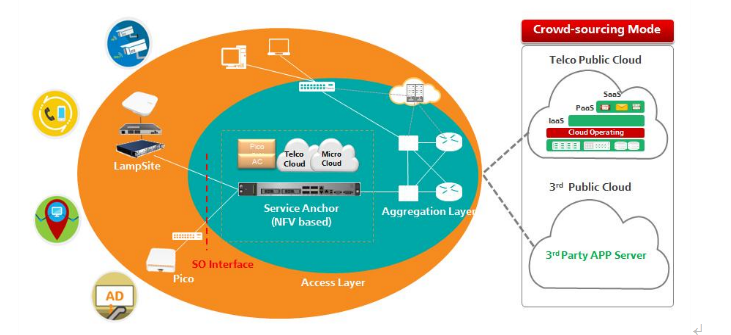
随着移动互联网的发展，室内企业市场也展现出更多的机遇。调查发现，超过40%的企业决策者选择MBB(Mobile Broadband)作为提升收入的首要方式，移动运营商也希望借助MBB开拓更多的业务，从而进一步提升营收。

在室内覆盖场景下，BYOD(Bring Your Own Device)逐渐在中型企业中流行。企业希望为员工提供无所不在的移动办公，同时室内大众消费场所也希望借助于移动互联网的发展为用户提供室内导航、业务推送等差异化业务，从而吸引更多客户、带来更多利润，赢取商业成功，并通过大数据分析提供商业决策依据。

针对以上需求，华为基于已有Small Cell室内覆盖提出了创新的Service AnchorTM解决方案。一方面小功率的Small Cell保证了运营商更好的覆盖和容量；另一方面，Service Anchor作为室内覆盖业务锚点，为Small Cell提供了控制节点功能，丰富无线网络功能，帮助运营商开拓室内蓝海。

Service Anchor是具有能力开放的室内网络控制器，向下连接Small Cell，向上可以连接电信公共云（传统核心网网元）和第三方应用云平台等。

网络结构图：



Service Anchor提供以下功能：

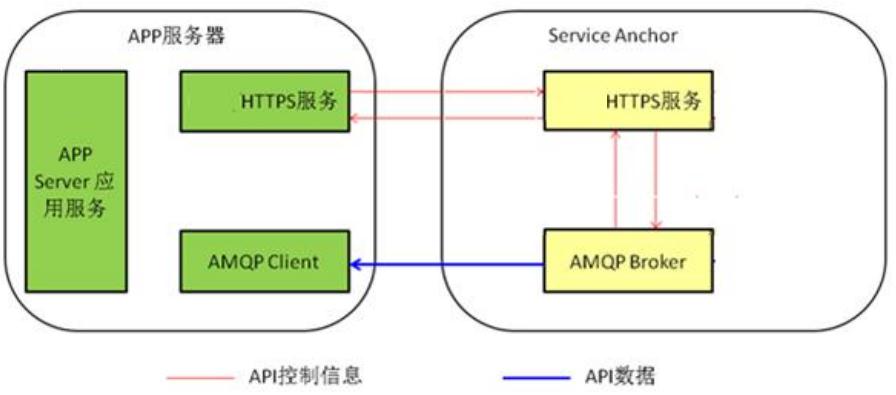
网络能力开放功能：在运营商的管理下对外提供位置API接口，满足企业、大众消费场所、OTT开发者的应用诉求。

室内无线网络控制节点功能：支持集成AE、AC、AG等功能组件，为Small Cell无线网络提供更丰富的功能。

### API接口

API接口是第三方APP对接AE的地址。

SVA API接口机制如下图所示。



设备对接信息表

| API IP | API Port | AMQP Broker IP | AMQP Broker Port | 位置信息 | 备注 |
| --- | --- | --- | --- | --- | --- |
| 192.168.10.2 | 9001 | 192.168.10.2 | 4703 | 新天府大道100号U9 | 如果是NAT组网，IP和Port要提供NAT后的与APP服务器能通的IP和Port。 |

SVA API主要由HTTPS服务模块和AMQP Broker组成。使用SVA API时，APP服务器需要先通过HTTPS的控制信息通道向SVA订阅API信息，SVA将响应信息中携带的AMQP Broker的IP地址和端口号以及SVA API数据队列ID等信息发送给APP服务器，APP服务器再和SVA建立AMQP连接，根据组网不同分为如下两种情况：

非NAT组网下，APP服务器可以选择使用SVA订阅响应信息中携带的AMQP Broker的IP地址和端口号或设备信息对接表中提供的AMQP Broker的IP地址和端口号向SVA发起AMQP建立连接请求，并绑定对应的队列ID，接收SVA发布的API数据。

NAT组网下，APP服务器只能使用设备信息对接表提供的AMQP Broker的IP地址和端口号向SVA发起AMQP建立连接请求，并绑定对应的队列ID，接收SVA发布的API数据。

更多关于使用API接口的指导可参考：http://support.huawei.com/carrier/docview?nid=DOC1000130726&path=PBI1-7851894/PBI1-21430741/PBI1-21430811/PBI1-21784974/PBI1-21481631

### 开发者社区

#### 源码托管

使用Github托管源码，源码地址为：<https://github.com/SVADemoAPP/webServer.git>

#### 开发指导

##### 环境要求（以windows系统为例）

###### PC硬件：

内存：8G+ / 硬盘：500G+ / CPU：Core i5+

###### jdk

需要安装jdk和jre，并配置path环境变量； jdk的版本为1.8+；

下载地址：http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

###### tomcat

需要安装tomcat，版本为7+；

下载地址：http://tomcat.apache.org/download-80.cgi

###### maven

需要安装maven，版本为3+；

下载地址：http://maven.apache.org/download.cgi

###### mysql

需要安装mysql，版本为5+；

下载地址：https://www.mysql.com/downloads/

###### eclipse

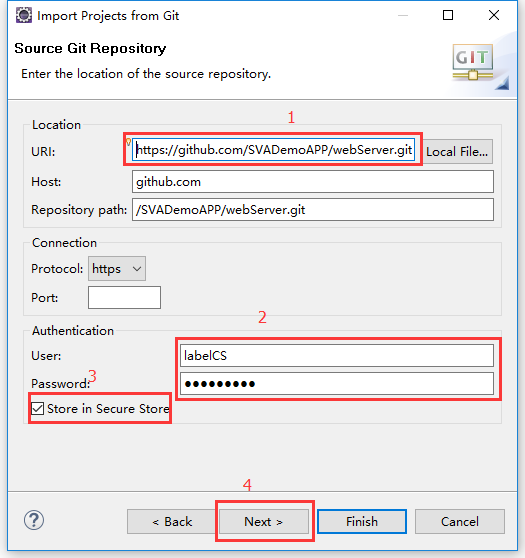
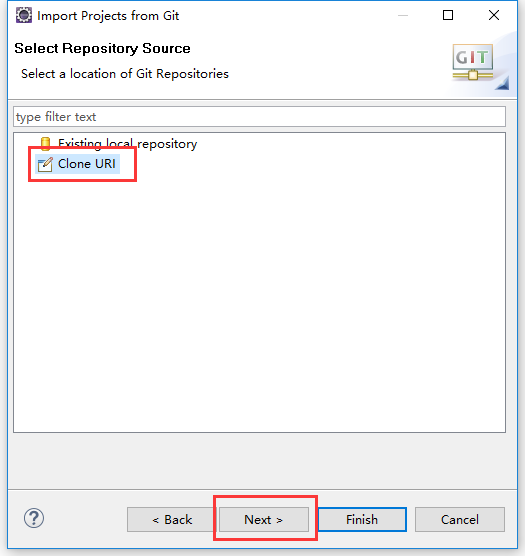
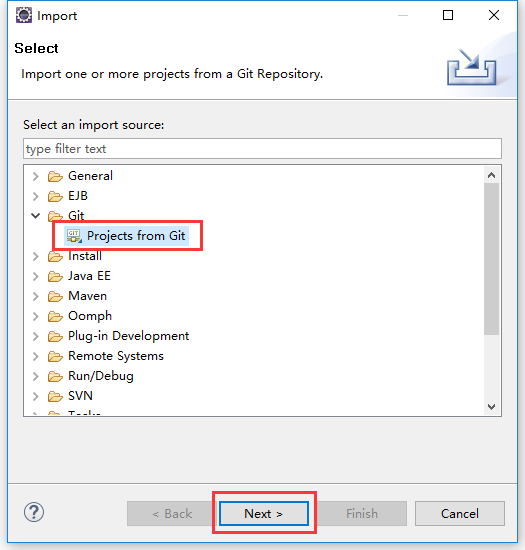
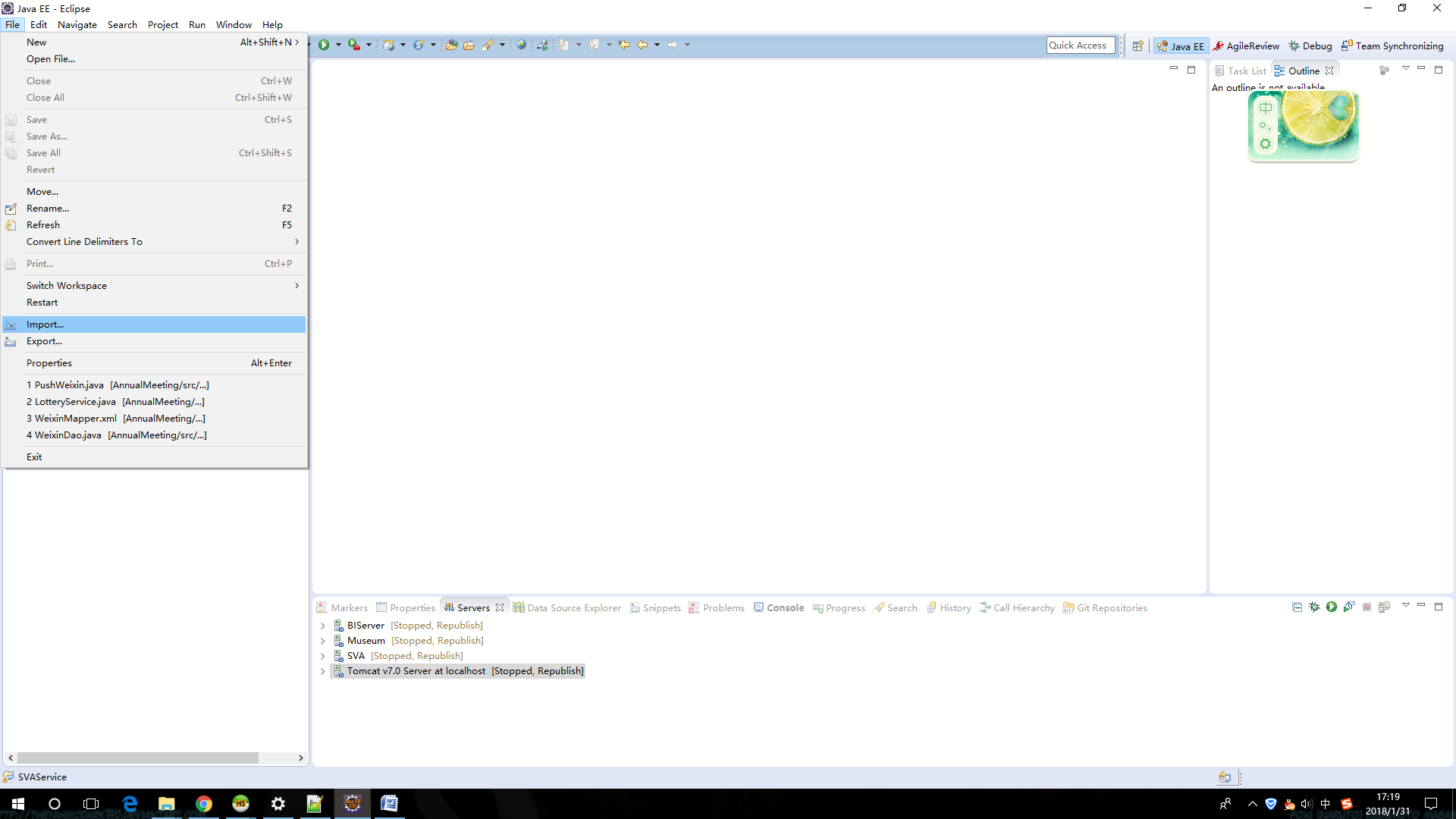
* 1. 需要安装eclipse，版本为Mars(4.5.2)+；
  2. 确保eclipse上已安装maven插件和git插件；
  3. 在eclipse里正确配置jdk、jre、tomcat和maven；
  4. 下载地址：https://www.eclipse.org/downloads/

以上均为基础操作，如有疑问可自行查阅相关资料

##### 将工程导入eclipse

###### 第一步，通过git插件导入demo，具体操作如图

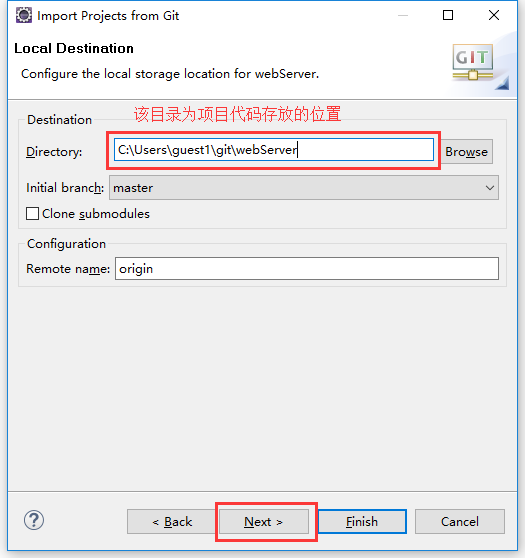
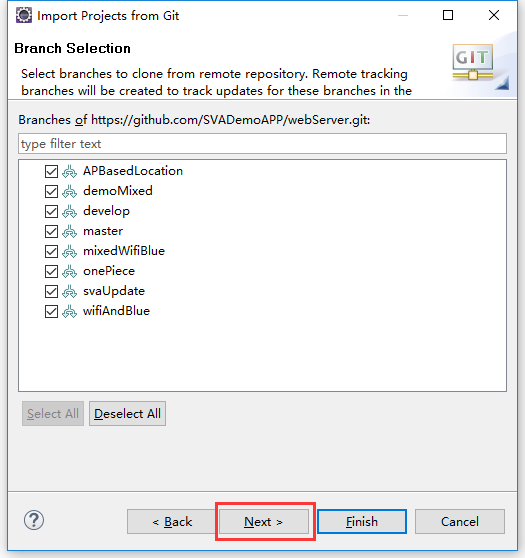
*File –> Import -> Git*



说明：

1->demo项目在github上的url，本项目为<https://github.com/SVADemoAPP/webServer.git>，如果是从demo项目fork下来的，请输入fork后的url；

2->输入你的github登录账户，并确保此账户有访问demo项目的权限，如没有权限，需要从demo项目fork出一个自己的项目，并使用该新项目开发（fork操作指导见文末github使用概要）；

3->勾选此选项避免每次git操作都要重新输入账户；

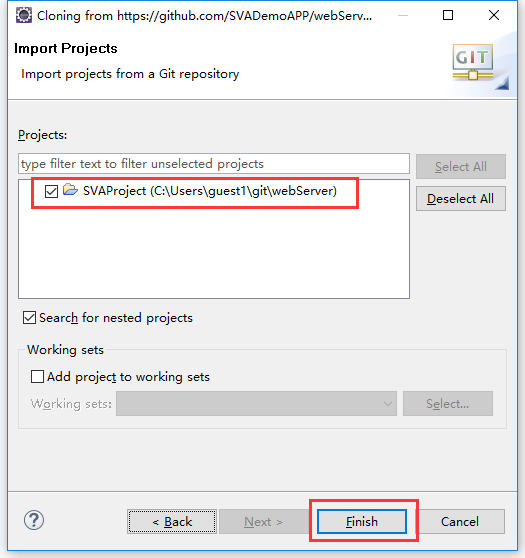
接下来开始从github上下载代码到本地，可能需要一些时间，完成后再进行下面的操作

###### 第二步，通过maven插件重新导入demo工程

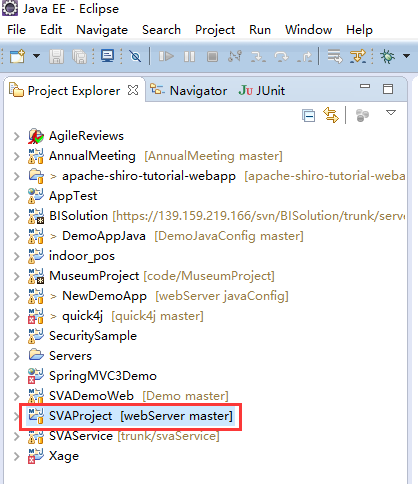
上一步操作完之后，eclipse出现如下界面：



勾选第一个选项，会出现上一步中代码下载至本地的位置，然后点击next



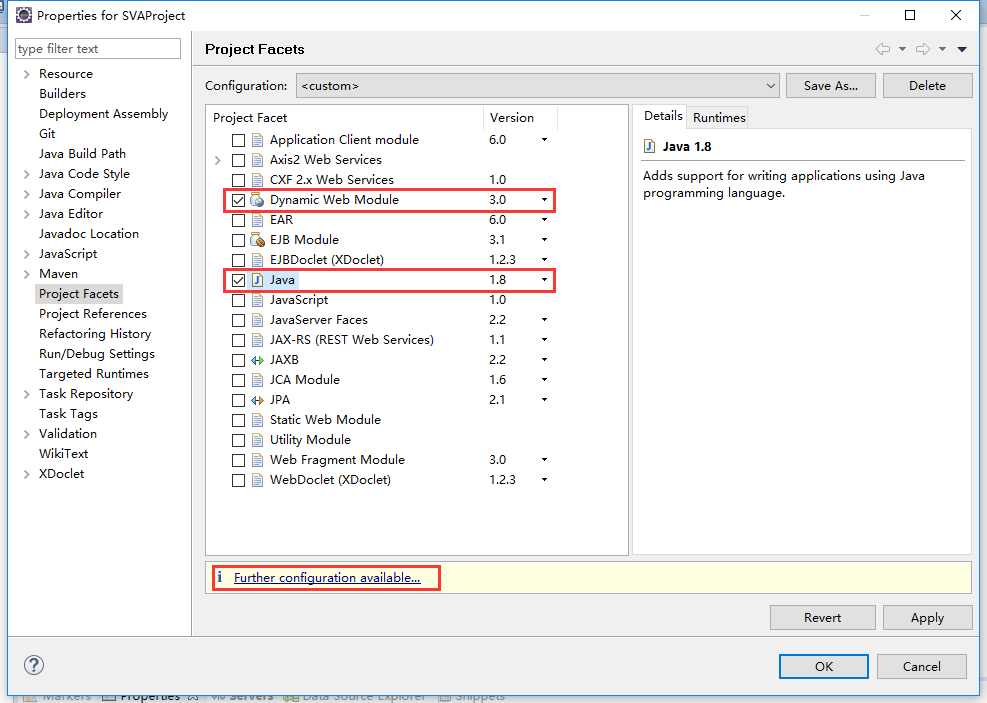
确保SVAProject已勾选（正常情况下默认勾选），然后Finish。稍等片刻，eclipse左侧的项目一览里会出现SVAProject项目，至此，demo项目导入完成。



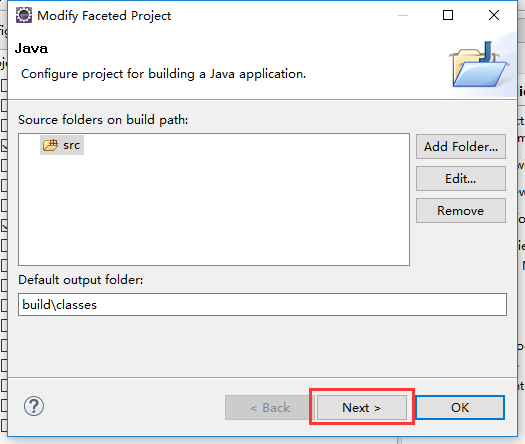
###### 第三步，Demo项目配置

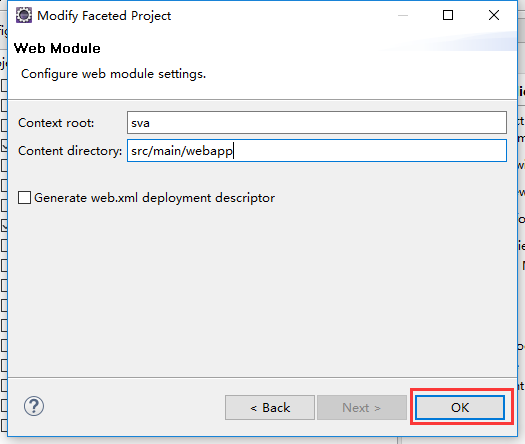
上一步导入完成后，eclipse会自动build，此过程会持续一段时间。Build结束后，需要对项目做一些配置，把项目变成Dynamic Web项目，以便能运行起来，具体操作如下：

1. 右键项目->属性->Projects Facets



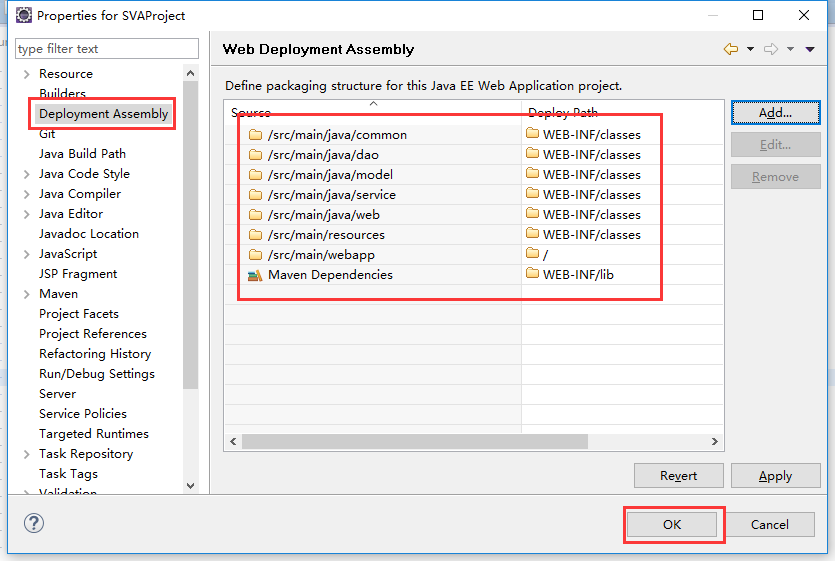
1. 点击*Further configuration available…*







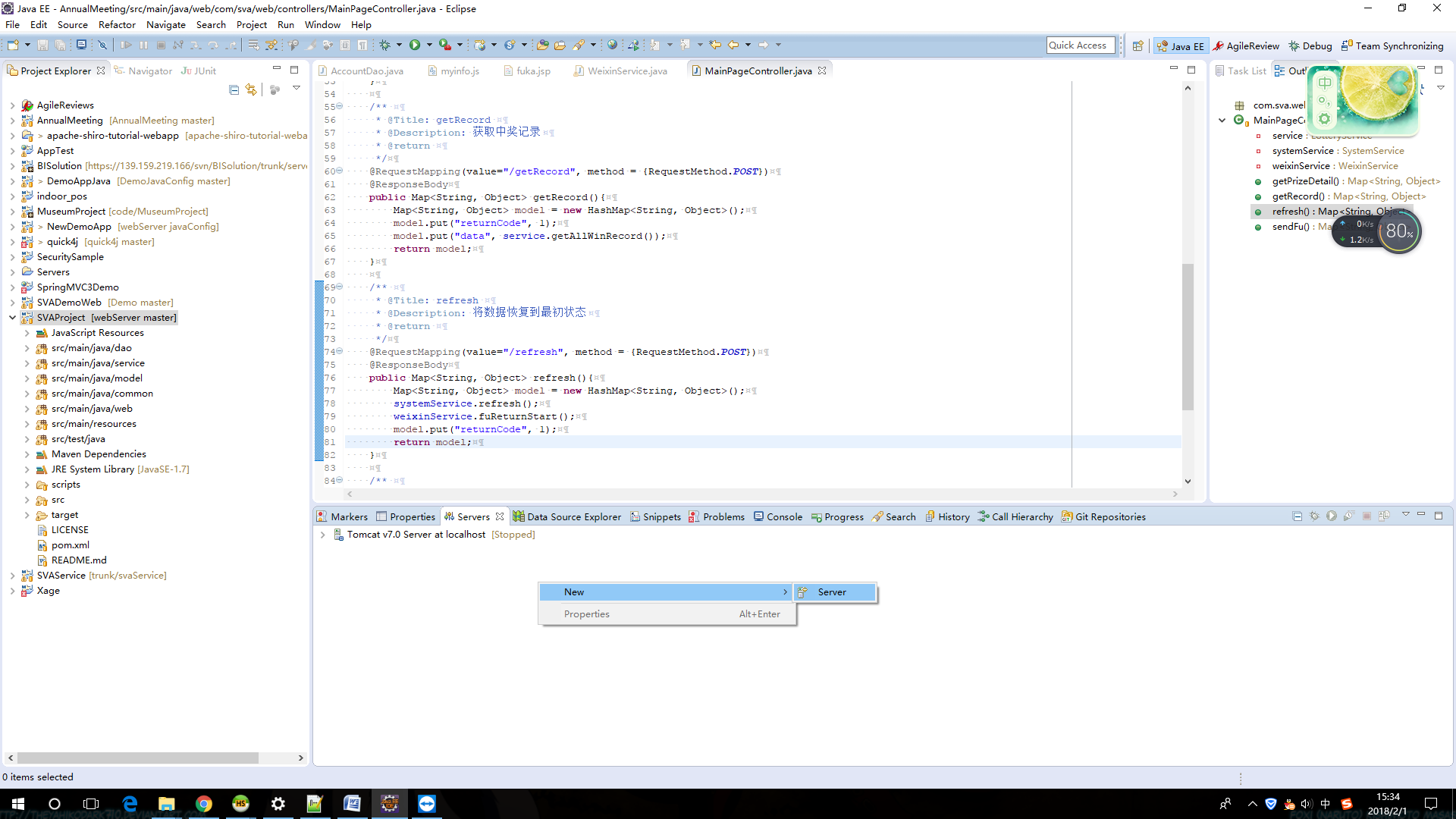
1. 设置部署程序集：再次右键项目->属性，打开属性窗口



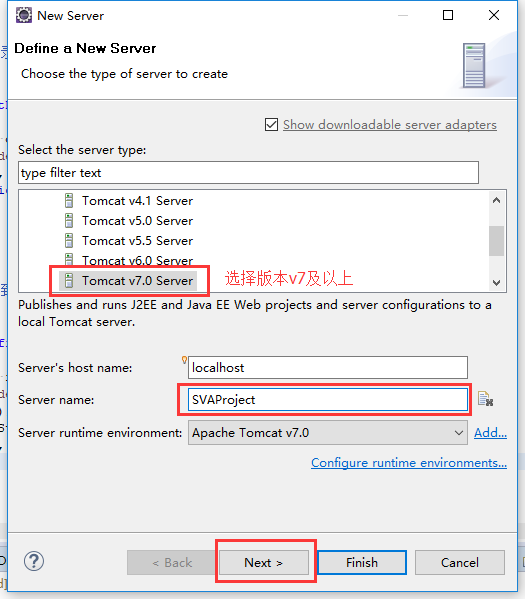
设置完成后的效果如上图，点击OK完成

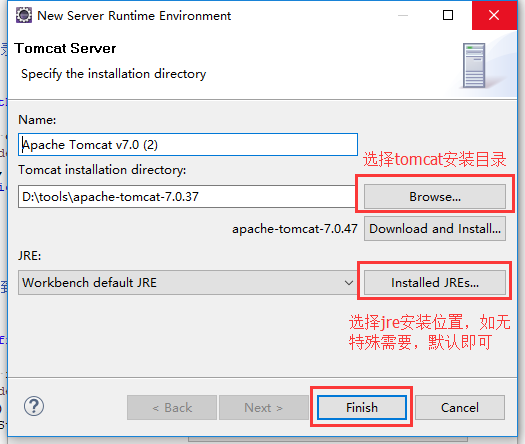
###### 第四步，在eclipse里通过tomcat启动项目

1. *打开Servers视图，右键空白区域，新建一个server*

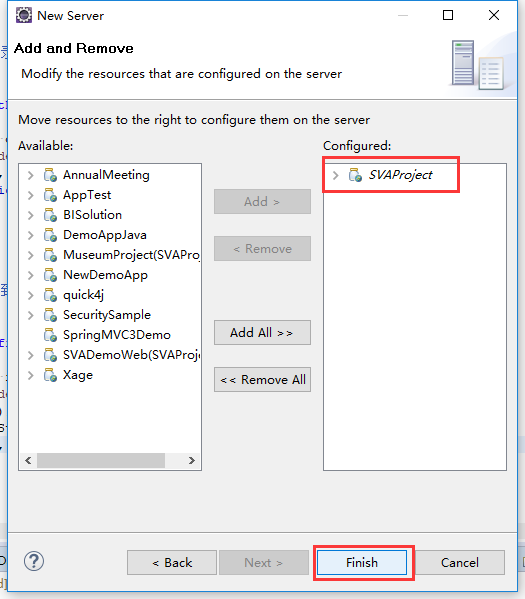
**

1. *选择电脑上已安装好的tomcat，进行发布；如果tomcat未配置好，点击Add配置*

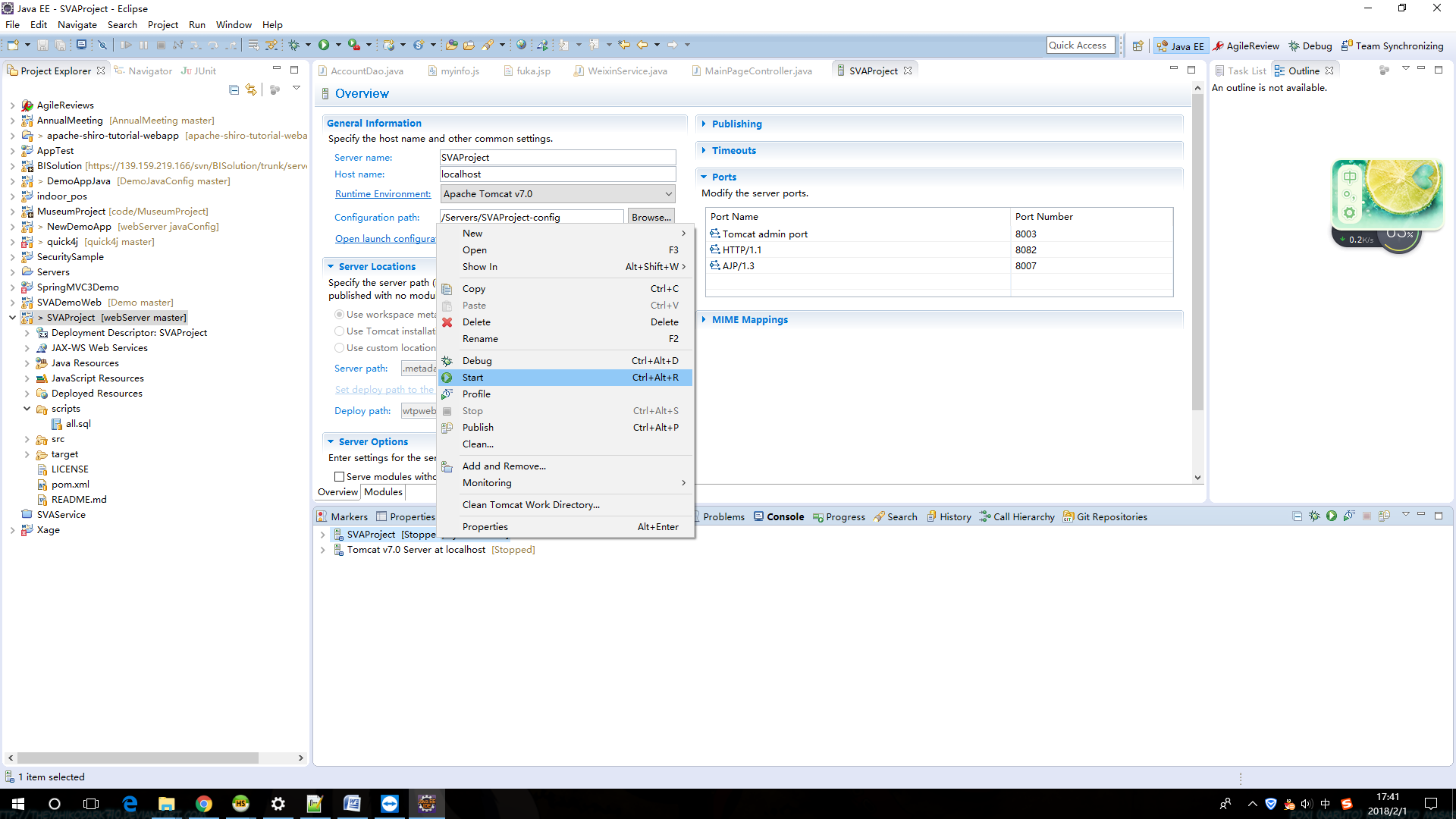
**

**

1. *选中demo项目并发布*

**

1. *完成上述步骤后，在Servers视图中出现了SVAProject服务，右键该服务->Start*

**

*（双击服务，可在弹出的页面查看服务发布的端口）*

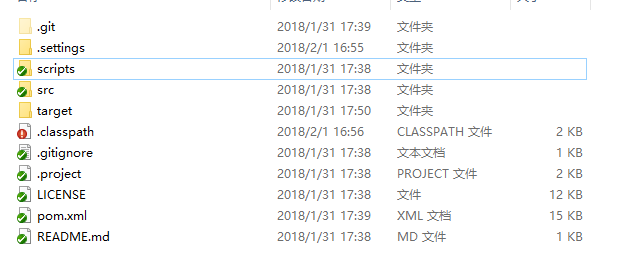
1. *在浏览器访问demo项目：打开google浏览器，输入localhost:8082/sva/account/login，出现如下页面，证明demo项目启动成功*

**

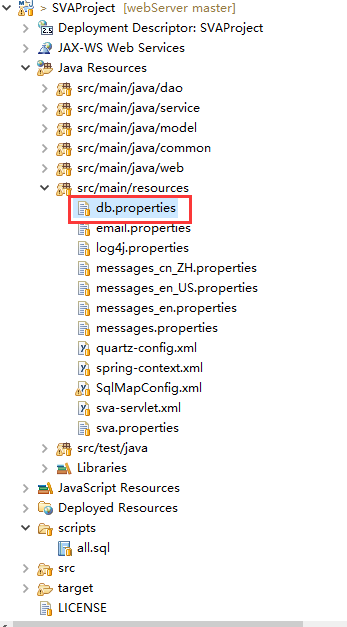
###### 第五步，数据库配置

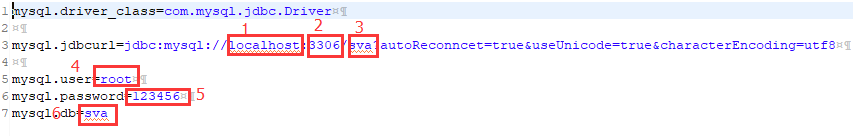
项目已启动成功，但若要正常使用相关业务，该需要对数据库进行配置。

1. 找到项目代码根目录的scripts文件夹



1. 已经安装好的mysql里，执行scripts里面的sql文件（可以使用相关mysql管理工具，如heidisql、navicat等），将数据表和初始数据导入数据库。
2. 根据自己的配置，修改demo项目的数据库配置文件，位置如下：





说明： 1->固定为localhost

2->mysql服务端口，3306

3->数据库名称，sva

4->mysql连接用户名，root

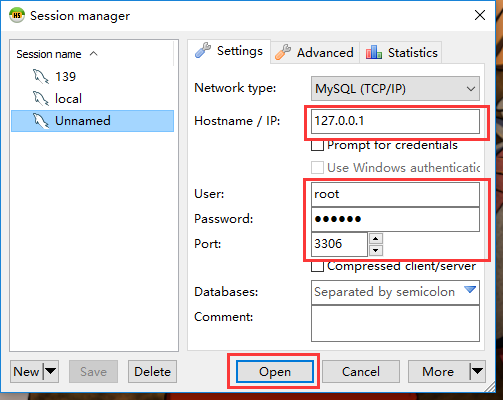
5->mysql连接密码，123456

6->同3

改了配置后：（Reloading Context with name [/sva] is completed）

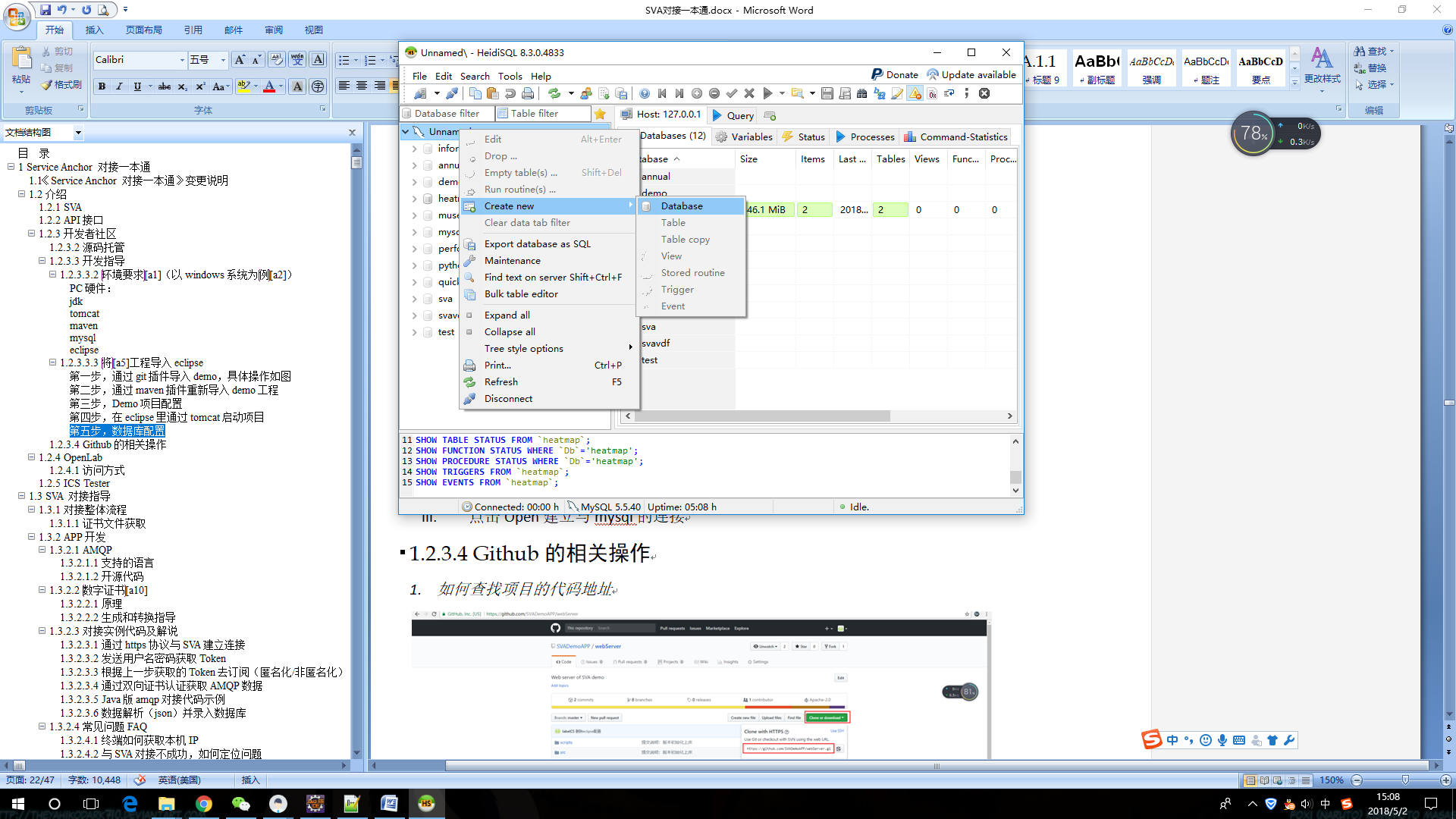
至此，demo项目已经正常运行，并通过web使用相关功能。

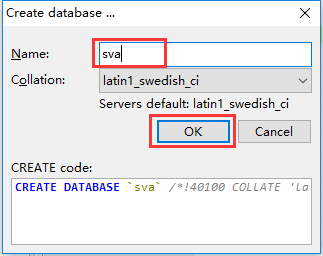
1. HeidiSQL工具使用说明
2. 下载地址：<https://www.heidisql.com/download.php>
3. 按照引导程序安装完成后打开，如图：



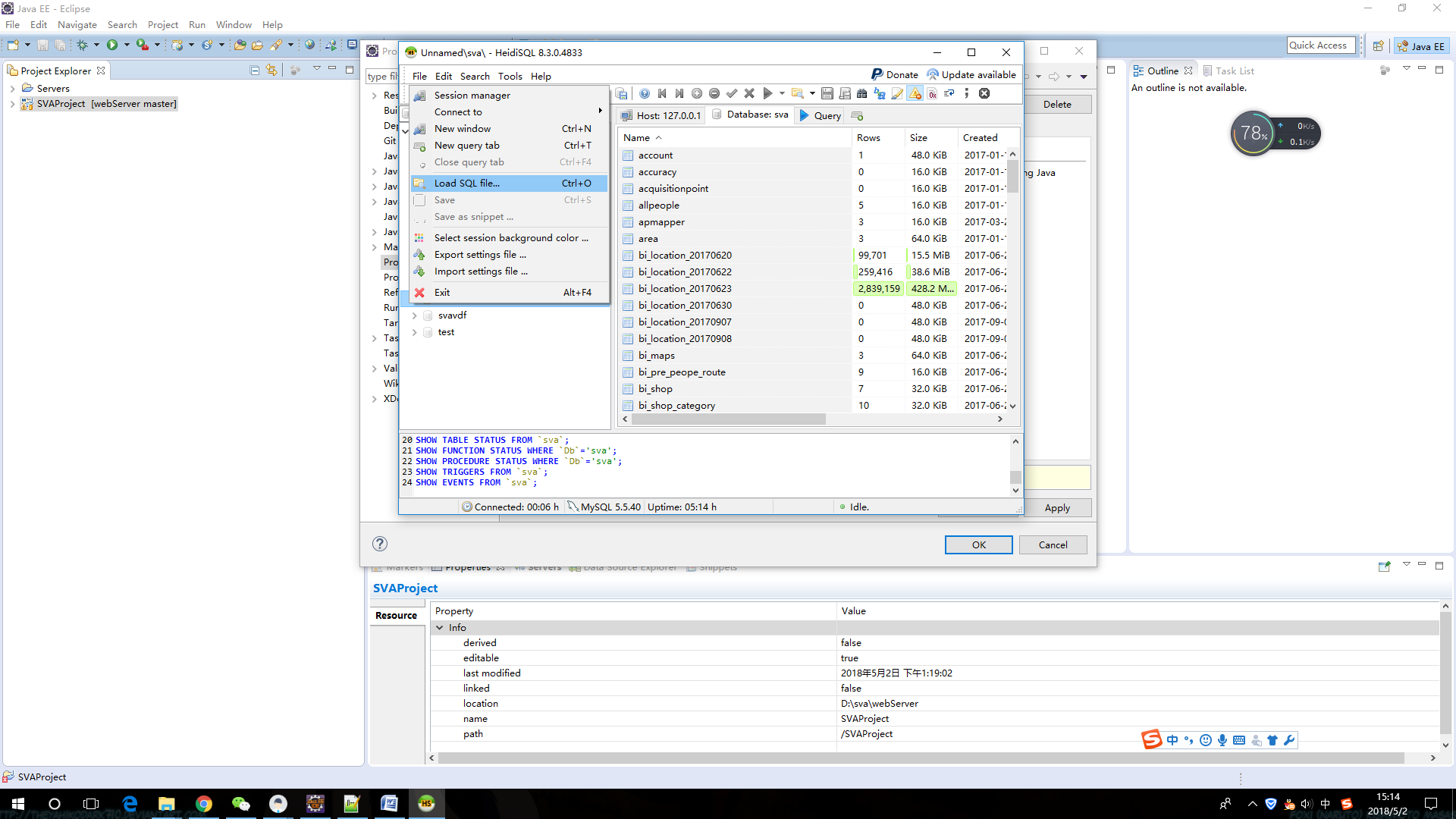
IP：127.0.0.1/ User：root/ Password：123456/ Port：3306

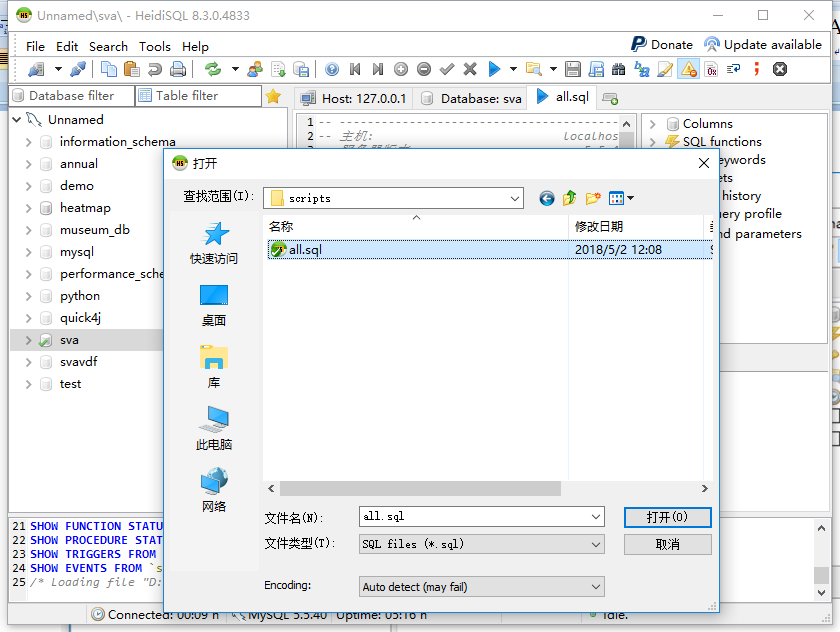
1. 点击Open建立与mysql的连接，并建立一个数据库sva，如图：



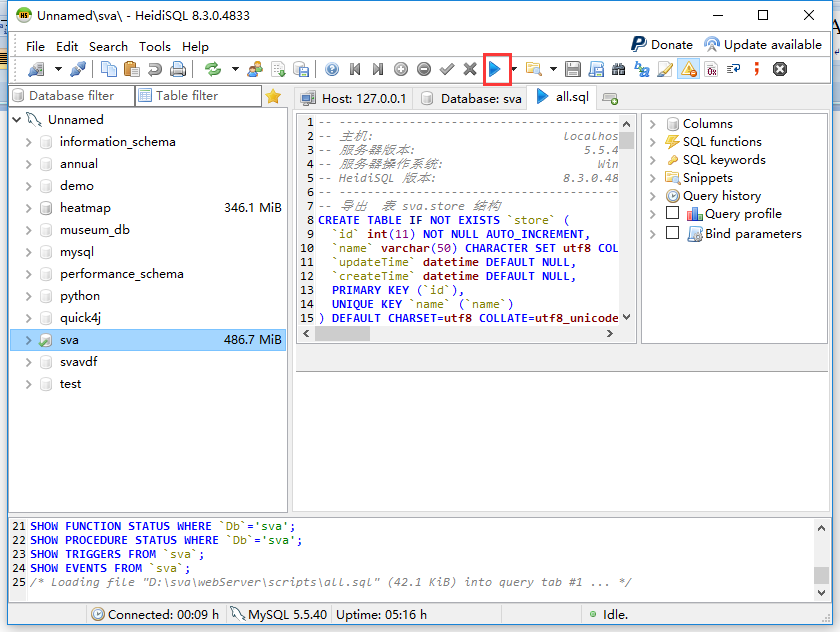


1. 点击选中新建的sva数据库，依次点击File — Load SQL File…，加载demoApp的sql文件（all.sql），如图：



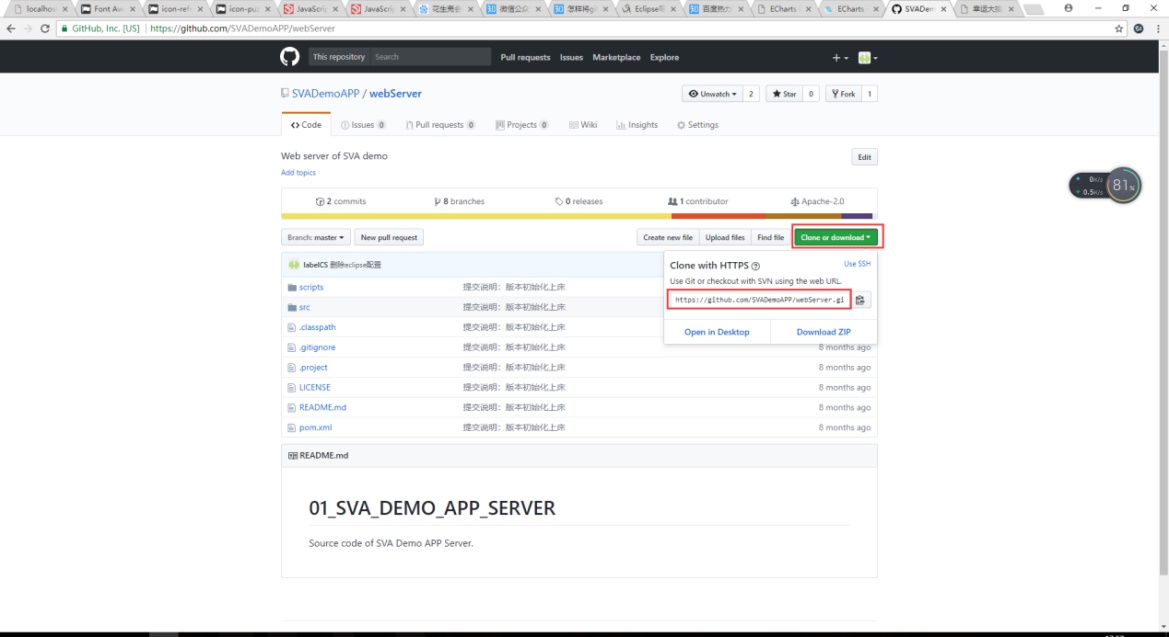


1. 点击蓝色三角符号，执行sql文件，至此demoapp的数据库配置完成。

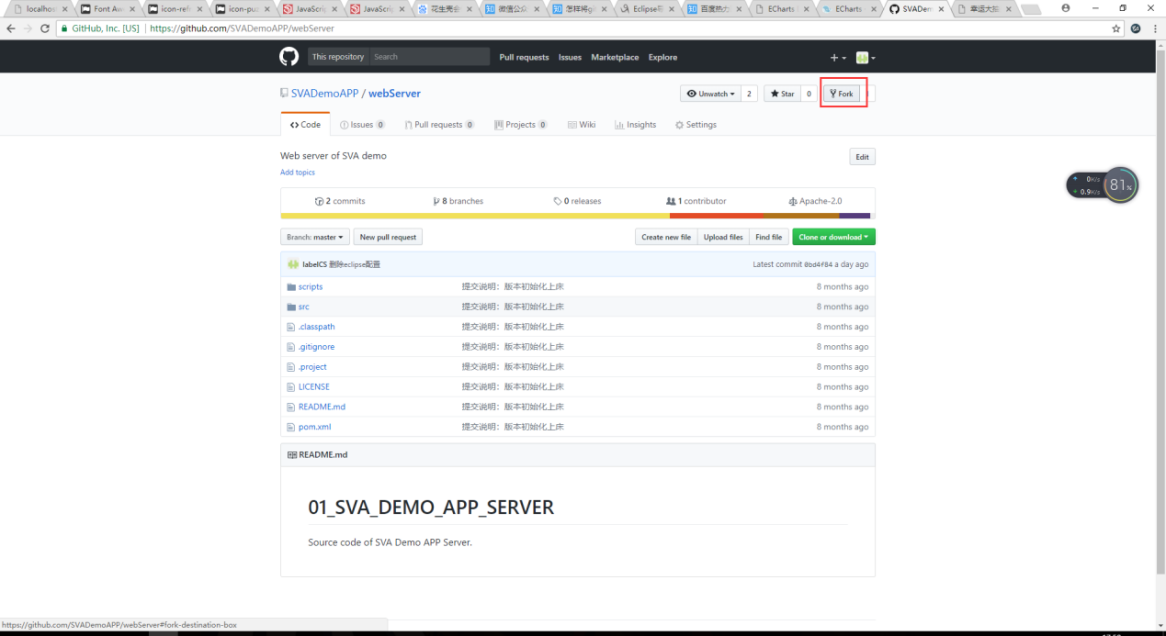


#### Github的相关操作

1. *如何查找项目的代码地址*

**

1. *如何fork项目*

**

*按照提示一步步进行操作即可*

### OpenLab

#### 访问方式

OpenLab是一套开放的测试环境，在真实环境尚未搭载时，他作为一个仿真的原生态环境和第三方APP进行对接测试。使用OpenLab减少了硬件需求，大大加快了第三方APP的对接效率，

目前在OpenLab上开放的API地址为218.4.33.216，开放端口为9001（获取token，订阅），4703（连接qpid服务器）

### ICS Tester

华为SVA（Service Anchor）通过自有知识产权的定位算法，为网络下的用户提供位置定位服务，从而将室内MBB网络变成一张室内GPS系统。

SVA提供定位数据服务，ICS LBS Tester V200R001C00工具作为SVA配套工具，提供相应的定位显示，客流统计分析，精度测试等功能。主要为SVA运维管理人员提供终端手机接入SVA的订阅服务及订阅后定位数据的实时监控及人流统计等功能展示，帮助SVA设备的客户及运维人员了解SVA定位的部分功能特性。

获取路径：

[**http://support.huawei.com/carrier/navi?coltype=software#col=software&from=product&detailId=PBI1-22517701&path=PBI1-7275736/PBI1-9855721/PBI1-7275869/PBI1-22517696**](http://support.huawei.com/carrier/navi?coltype=software#col=software&from=product&detailId=PBI1-22517701&path=PBI1-7275736/PBI1-9855721/PBI1-7275869/PBI1-22517696)

## SVA对接指导

### 对接整体流程

Restful接口：基于HTTP/HTTPS协议进行信息交换，数据接口采用AMQP协议进行数据交换，数据对接步骤：

1. 通过https协议与SVA建立连接

2. 发送用户名密码获取Token

3. 根据上一步获取的Token去订阅（匿名化/非匿名化）

4. 通过双向证书认证获取AMQP数据

5. 数据解析（json）并录入数据库

#### 证书文件获取

证书下载地址：https://github.com/SVADemoAPP/Release/tree/master/SVA%E5%AF%B9%E6%8E%A5%E4%B8%80%E6%9C%AC%E9%80%9A/Certificate

### APP开发

#### AMQP

Service Anchor通过AMQP协议向第三方提供数据

##### 支持的语言

Java/Python/C++

##### 开源代码

Github开源代码地址：<https://github.com/SVADemoAPP/webServer.git>

#### 对接实例代码及解说

##### 通过https协议与SVA建立连接

在HttpsService.java代码中看到，对接过程在SSL验证中要设置https.protocols为”sslVersion”，sslVersion为ssl的版本号（默认为”TLSv1”），由参数传进来，要与SVA上的设置一致，开发前需确认。具体可以参考HttpUtil.java代码。

// SSL验证

TrustManager[] tm = {xtm};

System.setProperty("https.protocols", sslVersion);

SSLContext ctx = SSLContext.getInstance(sslVersion);

ctx.init(null, tm, null);

con.setSSLSocketFactory(ctx.getSocketFactory());

con.setHostnameVerifier(hv);

##### 发送用户名密码获取Token

在获取Token过程中host\_ip为SVA的IP地址，还需要提供app\_name和app\_password，在响应中，获取token方法：

String token = con.getHeaderField("X-Subject-Token");

接口说明：

方法 URL 协议 内容描述格式 参数内容

POST https://host\_ip:9001/v3/auth/tokens HTTP/HTTPS（RESTful） JSON Body:

app\_name, app\_password

内容描述样例：

{

"auth": {

"identity": {

"methods": ["password"],

"password": {

"user": {

"domain": "Api",

"name": app\_name,

"password": app\_password

}

}

}

}

}

返回样例：{"token": {"methods": ["password"], "user": {"prompt\_day": null, "ip": "", "expire\_day": null, "domain": 3, "name": "app1", "pwd\_expired": 0, "id": "e79561f5141f43d19b58cd40649b881a", "pwd\_prompt": 0}, "level": 0, "roles": [{"id": "operators"}], "tenant": {"id": "default\_tenant", "name": "Default\_Tenant"}, "expires\_at": "2015-01-27 20:04:11", "issued\_at": "2015-01-26 20:04:11"}}

请求样例：

public static void main(String[] args)

{

String url = "https://192.168.200.164:9001/v3/auth/tokens";

String content = "{\"auth\":{\"identity\":{\"methods\":[\"password\"],\"password\": {\"user\": {\"domain\": \"Api\",\"name\": \"app2\",\"password\": \"User@123456\"}}}}}";

String charset = "UTF-8";

HttpsService capi = new HttpsService();

try

{

String to = capi.httpsPost(url, content, charset, "POST", null, "TLSv1");

Logger.getLogger(HttpsService.class).info(to);

}

catch (KeyManagementException e)

{

// TODO Auto-generated catch block

Logger.getLogger(HttpUtil.class).info(e.getStackTrace());

}

catch (NoSuchAlgorithmException e)

{

// TODO Auto-generated catch block

Logger.getLogger(HttpUtil.class).info(e.getStackTrace());

}

catch (IOException e)

{

// TODO Auto-generated catch block

Logger.getLogger(HttpUtil.class).info(e.getStackTrace());

}

}

##### 根据上一步获取的Token去订阅（匿名化/非匿名化）

通过上一步的获取的token值，去订阅，通过订阅获取QUEUE\_ID，订阅分为两种：

A） 匿名化订阅

B） 非匿名订阅

上面两种主要区分在接口，locationstreamreg为“非匿名化”订阅，locationstreamanonymousreg为“匿名化”订阅，主要区别可以查看返回样例中的userid。

https://host:9001/enabler/catalog/locationstream[anonymous]reg/json /v1.0

接口说明：

方法 URL 协议 内容描述格式 参数内容

POST https://host:9001/enabler/catalog/locationstream[anonymous]reg/json /v1.0 HTTP/HTTPS（RESTful） JSON

locationstreamreg为非匿名化订阅

locationstreamanonymousreg为匿名化订阅 Header:

token

Body:

app\_name

useridlist

内容描述格式样例：{"APPID":app\_name, "useridlist":["a1b1c1d1","a2b2c2d2"] }

返回样例：{"SubscribeInfos": [{"APPID": "app1", "APINAME": "location", "BROKER\_PORT": 4703, "QUEUE\_ID": "app1.1bc314b652924f088c644263ab135e1d.6f079ab0", "BROKER\_IP": "10.177.192.238"}], "result": {"error\_code" : 0, "error\_desc": "Success"}}

说明：useridlist为可选参数，当订阅消息中携带useridlist时，RAN Enabler按照useridlist指定的用户来采集和上报信息。当订阅消息中不携带useridlist时，RAN Enabler上报所有用户的位置信息。目前，useridlist中的userid只能是用户的IP地址。

取消订阅Un Subscription

方法 URL 协议 内容描述格式 参数内容

DELETE https://host:9001/enabler/catalog/locationstream[anonymous]unreg/json/v1.0 HTTP/HTTPS（RESTful） JSON

locationstreamreg为非匿名化订阅

locationstreamanonymousreg为匿名化订阅 Header:

Token

Body:

app\_name

useridlist

内容描述格式样例：{"APPID":app\_name, "useridlist":["a1b1c1d1","a2b2c2d2"]}

返回样例：NA

说明：useridlist为可选参数，当取消订阅消息中携带useridlist时，RAN Enabler停止useridlist指定的用户的信息采集和上报。当订阅消息中不携带useridlist时，RAN Enabler停止上报所有用户的位置信息。目前，useridlist中的userid只能是用户的IP地址。

##### 通过双向证书认证获取AMQP数据

信息发布采用AMQP协议格式进行发布，用户位置信息作为AMQP的净荷进行发布，信息内容采用JSON格式进行描述，信息内容最大长度为1024。发布的内容示例如下

“匿名化”订阅(红色部分区别与“非匿名化”订阅)：

'{"locationstreamanonymous":[{"IdType":"user id","Timestamp":1428030491000,"datatype":"coordinates","location":{"x":81.0,"y":-15.0,"z":1},"userid":["8589934849"]},{"IdType":"user id","Timestamp":1428030491000,"datatype":"coordinates","location":{"x":81.0,"y":-15.0,"z":1},"userid":["8589934850"]},{"IdType":"user id","Timestamp":1428030491000,"datatype":"coordinates","location":{"x":81.0,"y":-15.0,"z":1},"userid":["8589934851"]},{"IdType":"user id","Timestamp":1428030491000,"datatype":"coordinates","location":{"x":81.0,"y":-15.0,"z":1},"userid":["8589934852"]},{"IdType":"user id","Timestamp":1428030491000,"datatype":"coordinates","location":{"x":81.0,"y":-15.0,"z":1},"userid":["8589934853"]},{"IdType":"user id","Timestamp":1428030491000,"datatype":"coordinates","location":{"x":81.0,"y":-15.0,"z":1},"userid":["8589934854"]}]}'

“非匿名化”订阅

'{"locationstream":[{"IdType":"IP","Timestamp":1428030491000,"datatype":"coordinates","location":{"x":81.0,"y":-15.0,"z":1},"userid":["0a0a0a0a"]}]}'

对接代码在AmqpThread.java文件中，每一次对接单独起一个线程，通过GlobalConf.java文件里的线程池管理，调用代码如下：

AmqpThread at = new AmqpThread(sva,amqpDao,queueId);

GlobalConf.addAmqpThread(sva.getId(), at);

at.start();

##### Java版amqp对接代码示例

public void run()

{

String ip = sva.getIp();

String id = sva.getId();

String userId = sva.getUsername();

String port = sva.getBrokerPort();

// 获取keystore的路径

String path = System.getProperty("user.dir");

path = path.substring(0, path.indexOf("bin"))+"webapps/SVAProject/WEB-INF";

path = path + File.separator + "java\_keystore" + File.separator;

// 设置系统环境jvm参数,证书设置，用于amqp对接

System.setProperty("javax.net.ssl.keyStore", path + "keystore.jks");

System.setProperty("javax.net.ssl.keyStorePassword", "importkey");

System.setProperty("javax.net.ssl.trustStore", path + "mykeystore.jks");

System.setProperty("javax.net.ssl.trustStorePassword", "importkey");

// 地址变量

String brokerOpts = "?brokerlist='tcp://"+ip+":"+port+"?ssl='true'&ssl\_verify\_hostname='false''";

String connectionString = "amqp://"+userId+":"+"xxxx@xxxx/"+brokerOpts;

LOG.debug("connection string:" + connectionString);

// 建立连接

try {

conn = new AMQConnection(connectionString);

conn.start();

LOG.debug("connection started!");

// 获取session

Session session = conn.createSession(false, Session.AUTO\_ACKNOWLEDGE);

LOG.debug("session created!");

// 获取队列

Destination queue = new AMQAnyDestination("ADDR:"+queueId+";{create:sender}");

MessageConsumer consumer = session.createConsumer(queue);

LOG.debug("consumer created!");

while(!isStop && !this.isInterrupted())

{

Message m = consumer.receive(60000);

// message为空的情况,

if(m == null){

LOG.debug("Get NULL message, pause for 1 miniute!");

continue;

}

// message格式是否正确

if(m instanceof BytesMessage){

BytesMessage tm = (BytesMessage)m;

int length = new Long(tm.getBodyLength()).intValue();

if(length > 0){

byte[] keyBytes = new byte[length];

tm.readBytes(keyBytes);

String messages = new String(keyBytes);

// 数据解析并存入数据库

saveSvaData(messages, Integer.parseInt(sva.getStoreId()), sva.getType(), sva.getIp(), sva.getId());

LOG.debug("SVA Data:"+messages);

}else{

LOG.debug("Get zero length message");

}

}else{

LOG.warn("Message is not in Byte format!");

}

}

} catch (URLSyntaxException e) {

LOG.error(e);

} catch (QpidException e) {

LOG.error(e);

} catch (JMSException e) {

LOG.error(e);

} catch (URISyntaxException e) {

LOG.error(e);

} catch (Exception e){

LOG.error(e);

} finally{

try {

if(conn != null)

{

conn.close();

}

} catch (JMSException e) {

LOG.error(e);

}

LOG.error("[AMQP]No data from SVA,connection closed!");

}

}

##### 数据解析（json）并录入数据库

根据获取的数据类型，调用不同的方法进行解析并入库；

private void saveSvaData(String jsonStr, int storeId, int type, String ip, String svaId){

if(StringUtils.isEmpty(jsonStr)){

LOG.warn("No data from SVA!");

mylog.location("No data from SVA!");

}else{

JSONObject result = JSONObject.fromObject(jsonStr);

// 非匿名化订阅

if(result.containsKey("locationstream")){

saveLocationstream(result, storeId, type, svaId);

}

// 匿名化订阅

else if(result.containsKey("locationstreamanonymous")){

saveLocationstreamAnonymous(result, storeId, svaId);

}

// 网络信息订阅

else if(result.containsKey("networkinfo")){

saveNetworkinfo(result, ip);

}

// 电子围栏订阅

else if(result.containsKey("geofencing")){

saveGeofencing(result);

}else{

LOG.warn("Unknow message from SVA!");

mylog.location("Unknow message from SVA!");

}

}

}

针对非匿名化订阅查看解析入库的代码，其他类型的解析入库方式基本一致

private void saveLocationstream(JSONObject result, int storeId, int type, String svaId)

{

JSONArray list = result.getJSONArray("locationstream");

for(int i = 0; i<list.size();i++){

LocationModel lm = new LocationModel();

JSONObject loc = list.getJSONObject(i);

if(!parseLocation(loc, storeId, lm, svaId)){

continue;

}

// 全量订阅

if(type == 0){

// 获取当天的日期字符串

String tableName = "location" + ConvertUtil.dateFormat(loc.getLong("Timestamp"), "yyyyMMdd");

// 执行数据库保存逻辑

dao.saveAmqpData(lm,tableName);

}

// 指定用户订阅

else

{

//检查该用户是否已经存在

int count = dao.checkPhoneIsExisted(lm.getUserID());

if(count > 0){

dao.updatePhoneLocation(lm);

}else{

dao.saveAmqpData(lm, "locationphone");

}

}

}

}

private boolean parseLocation(JSONObject loc, int storeId, LocationModel lm, String svaId){

// 当前时间戳

long timeLocal = System.currentTimeMillis();

lm.setTimestamp(new BigDecimal(timeLocal));

// 设置LocationModel

JSONObject location = loc.getJSONObject("location");

lm.setSvaId(svaId);

lm.setIdType(loc.getString("IdType"));

lm.setTimestamp(BigDecimal.valueOf(System.currentTimeMillis()));

lm.setTimestampSva(BigDecimal.valueOf(loc.getLong("Timestamp")));

lm.setDataType(loc.getString("datatype"));

lm.setX(BigDecimal.valueOf(location.getInt("x")));

lm.setY(BigDecimal.valueOf(location.getInt("y")));

int z = location.getInt("z");

JSONArray useridList = loc.getJSONArray("userid");

// 用户存在多个的情况，目前只取第一个；若用户为空则不作处理

if(useridList.size()>0){

lm.setUserID(useridList.getString(0));

}else{

return false;

}

// 楼层号转换

if(z > 0){

z += storeId\*10000;

}else{

z = Math.abs(z) + 5000 + storeId\*10000;

}

lm.setZ(BigDecimal.valueOf(z));

return true;

}

#### 常见问题FAQ

##### 终端如何获取本机IP

安卓手机：设置— 关于手机 — 状态信息 — ip地址

##### 与SVA对接不成功，如何定位问题

参照1.3.3.5.1章节，对日志里的信息进行分析，查找原因

##### 为什么对接成功后，过一段时间连接会自动断开

从SVA获取token是有时间限制的，SVA侧可以设置有效时长，有效期结束后，订阅会被取消，此时需要重新获取token进行订阅。

关于有效时长的信息，可以从获取token的返回值中查看。如果需要修改有效时长，可以联系SVA负责人。

### APP与SVA对接测试

#### 用例

用户使用demo app与SVA对接，并查看热力图

#### 环境

操作系统：Window 7 Service Pack 1、Window Service 2008

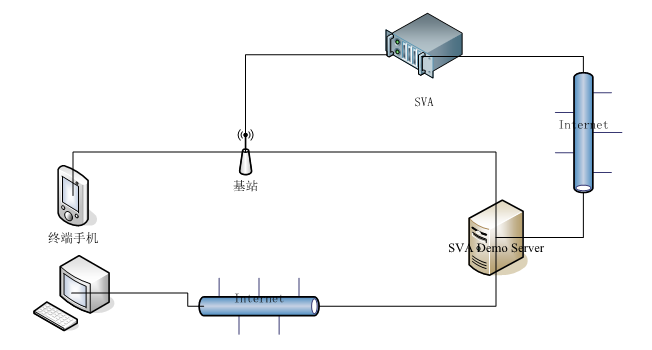
内存：4G以上

CPU：2.0GHz

硬盘：60G以上

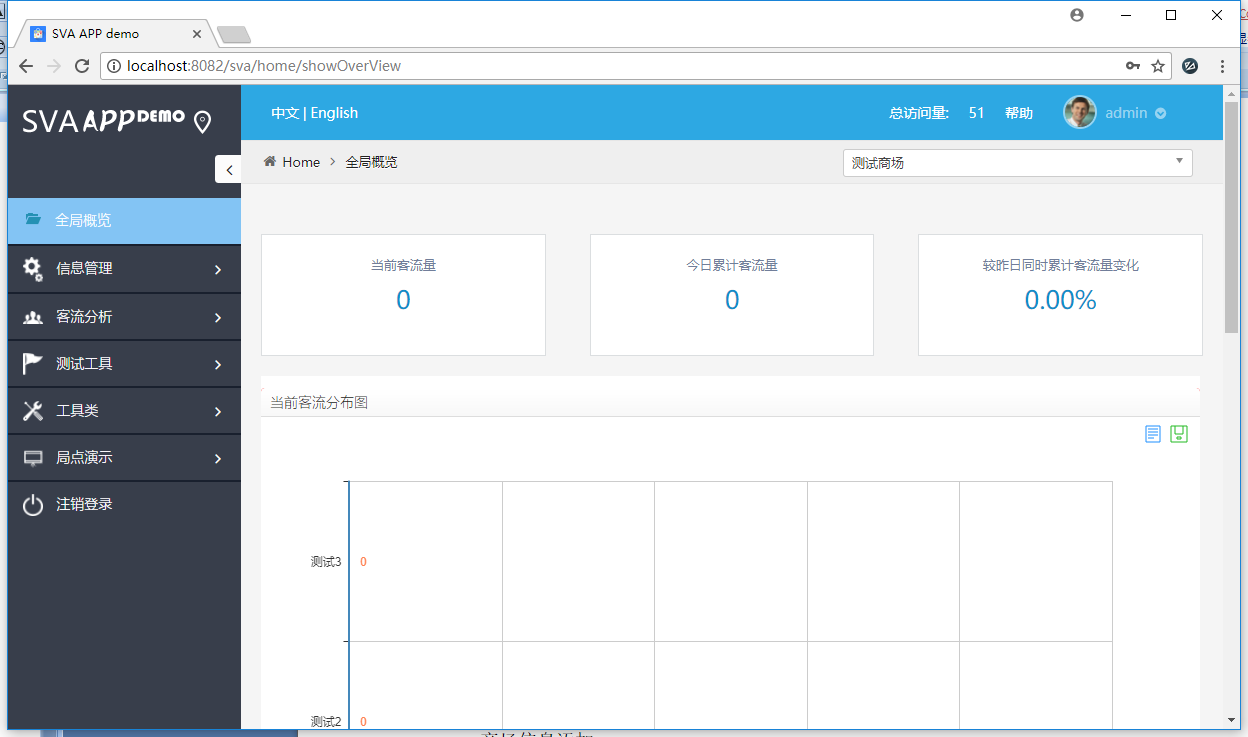
浏览器：IE9、Chrome 40.0、Firefox 36及以上版本

#### 传输组网



#### 操作指导

##### 启动工程，访问demo首页，如下图



##### 选择菜单-信息管理-商场管理

该模块主要用于商场信息的添加、查看、修改和删除。

商场信息添加：

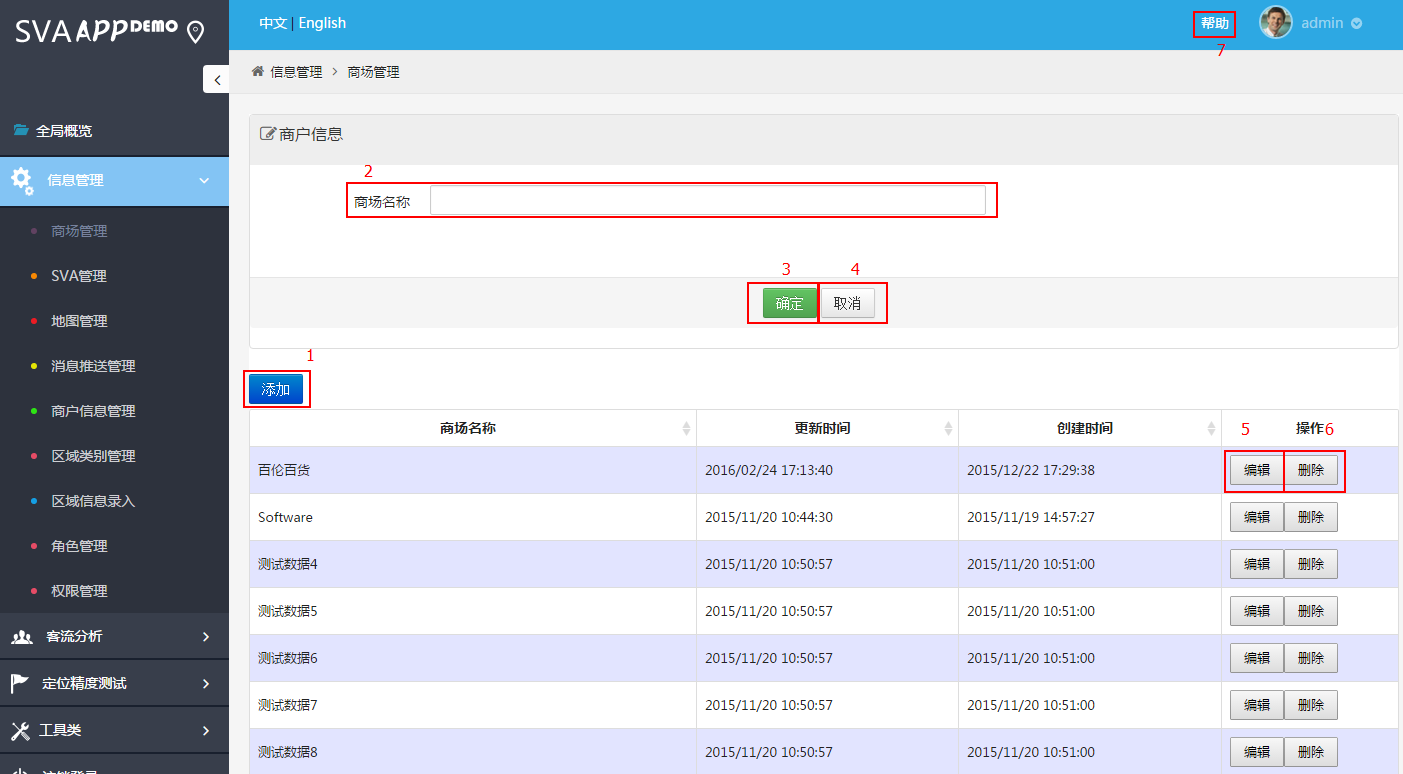
1. 进入【信息管理】 -- 【商场管理】页面；
2. 点击【添加】按钮，输入商场名称；
3. 点击【确定】按钮，保存商场信息；
4. 保存成功，商场信息会自动添加到下面的列表中；

商场信息删除：

1. 进入【信息管理】 -- 【商场管理】页面；
2. 点击要删除的商场信息对应的【删除】按钮；
3. 在弹出的确认删除对话框中点击【确定】按钮；
4. 删除成功后，被删除商场信息将在下面的列表中删除；

商场信息修改：

1. 进入【信息管理】 -- 【商场管理】页面；
2. 点击【编辑】按钮，编辑要修改的信息；
3. 修改后，点击【确定】按钮，保存修改后的商场信息；
4. 保存成功，修改后的商场信息会将原有的商场信息覆盖，从而添加到下面列表中。



商场管理

说明：

1. 添加操作：点击【添加】按钮，弹出新增配置页面；
2. 商场名称：可以由中文、字母、数字、特殊字符（~、!、@、#、^、\*、()、-、\_、+、|、\、[]、{}、:、.、/、<>）组成，最大长度为50位；
3. 确定操作：点击【确定】按钮，对添加的商场信息进行保存。保存后商场信息存放在数据库中；
4. 取消操作：点击【取消】按钮，取消对已添加的商场信息进行保存。取消添加后，列表中无新增商场信息；
5. 编辑操作：点击【编辑】按钮，对已存在的商场信息进行修改；
6. 删除操作：点击【删除】按钮，确认删除后，则该条数据将从列表中和数据库中删除；并刷新页面展示最新的数据信息；
7. Help操作：点击【帮助】按钮，页面跳转到帮助文档页面。

##### 选择菜单-信息管理-sva管理

该模块主要用于SVA服务器对接的配置，输入：IP、SVA名称、Token端口号、Broker端口号、用户名、密码，通过选择商场名称、状态、订阅类型来添加。

SVA信息添加：

1. 进入【信息管理】 -- 【SVA管理】页面；
2. 点击【添加】按钮，输入所有信息；
3. 点击【确定】按钮，保存SVA信息；
4. 保存成功，SVA信息会自动添加到下面的列表中；

SVA信息删除：

1. 进入【信息管理】 -- 【SVA管理】页面；
2. 点击要删除的SVA信息对应的【删除】按钮；
3. 在弹出的确认删除对话框中点击【确定】按钮；
4. 删除成功后，被删除SVA信息将在下面的列表中删除；

SVA信息修改：

1. 进入【信息管理】 -- 【SVA管理】页面；
2. 点击【编辑】按钮，编辑要修改的信息；
3. 修改后，点击【确定】按钮，保存修改后的SVA信息；
4. 保存成功，修改后的SVA信息会将原有的SVA信息覆盖，从而添加到下面列表中。



SVA管理

说明：

1. 添加操作：点击【添加】按钮，弹出新增配置页面；
2. IP：SVA服务器的IP（由SVA服务器维护人员提供）；
3. SVA名称：可以由中文、字母、数字、下划线组成，最大长度为20位；（test）
4. Token端口号：由SVA服务器维护人员提供；（9001）
5. Broker端口号：由SVA服务器维护人员提供；（4703）
6. 用户名：可以由中文、字母、数字、下划线组成，最大长度为20位（由SVA服务器维护人员提供）；（app1）
7. 密码：最大长度为20位（由SVA服务器维护人员提供）；（User@1234）
8. 商场名称：通过下拉框选择（商场名称由商场管理模块提供），例如U6食堂、王府井；
9. 状态：开启/关闭功能；选择是否需要接收处理SVA服务器发过来的数据。选择开启后，则接收处理SVA服务器发过来的数据；选择关闭后，则不接收处理SVA服务器发过来的数据；
10. 订阅类型：匿名化（服务端接收到的信息是由用户的非真实ID上传的）、非匿名化（服务端接收到的信息是由用户用真实ID上传的）、指定用户（服务端接只接收指定用户ID上传的信息）；
11. 确定操作：点击【确定】按钮，对添加的SVA信息进行保存。保存后SVA信息存放在数据库中；
12. 取消操作：点击【取消】按钮，取消对已添加的SVA信息进行保存。取消添加后，列表中无新增SVA信息；
13. 恢复/停用操作：点击【停用】按钮，对应的SVA服务器将不能使用，同时按钮的名字会修改为【恢复】，SVA状态改为【关闭】；点击【恢复】按钮，将对应的不能使用的SVA服务器启动，同时按钮的名字修改为【停用】， SVA状态将会变为【开启】；
14. 编辑操作：点击【编辑】按钮，对已存在的SVA信息进行修改；
15. 删除操作：点击【删除】按钮，确认删除后，则该条数据将从列表中和数据库中删除；并刷新页面展示最新的数据信息；

##### 选择菜单-信息管理-地图管理

该模块主要用于添加、修改和删除地图，导入地图文件支持20M以下大小的图片，图片格式支持.jpg、.png。

地图下载地址：https://github.com/SVADemoAPP/Release/tree/master/SVA%E5%AF%B9%E6%8E%A5%E4%B8%80%E6%9C%AC%E9%80%9A/map

地图信息添加：

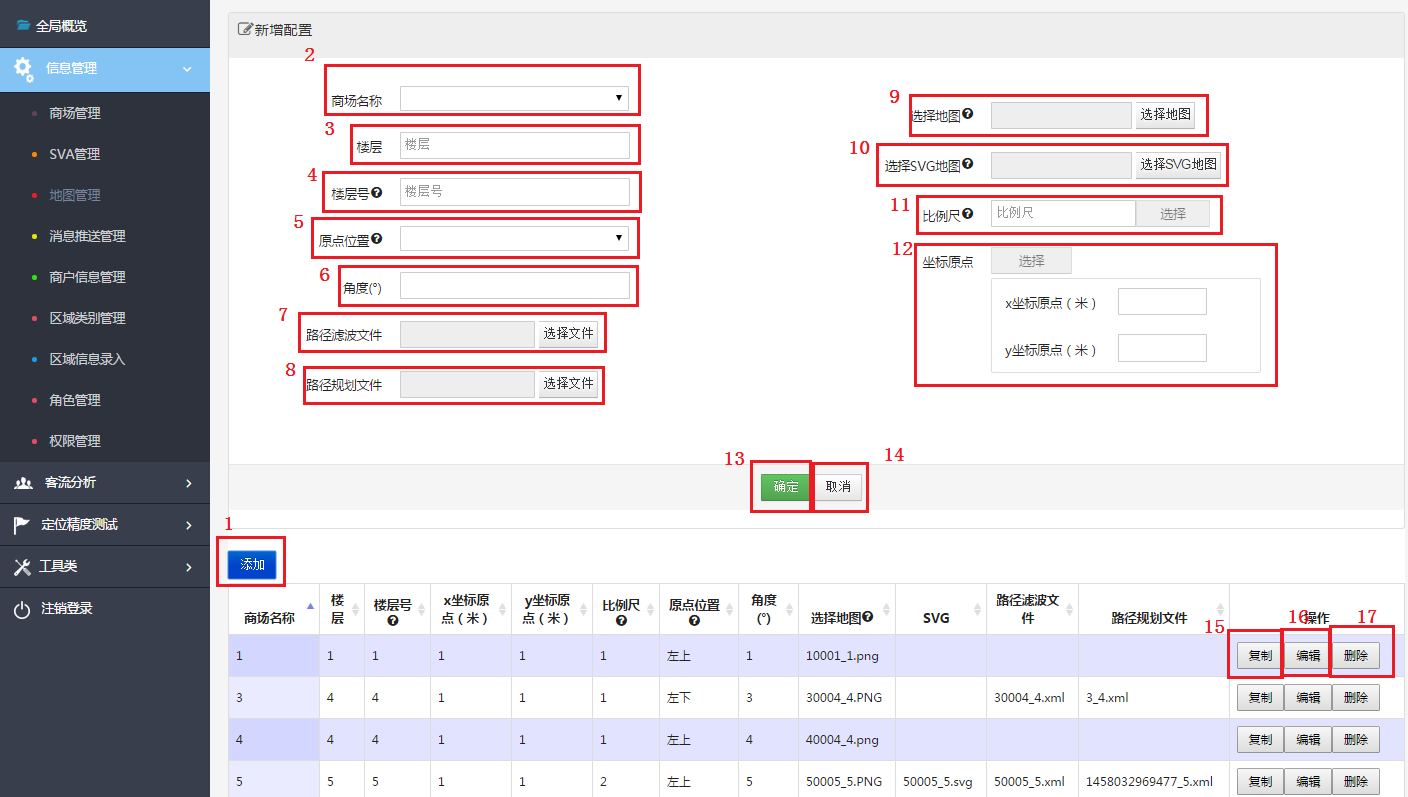
1. 进入【信息管理】 -- 【地图管理】页面；
2. 点击【添加】按钮，录入所有信息，并选择要导入的地图文件；
3. 点击【确定】按钮，保存地图信息；
4. 保存成功，楼层地图信息会自动添加到下面的列表中；

地图信息删除：

1. 进入【信息管理】 -- 【地图管理】页面；
2. 点击要删除的楼层地图对应的【删除】按钮；
3. 在弹出的确认删除对话框中点击【确定】按钮；
4. 删除成功后，被删除楼层地图信息将在下面的列表中删除；

地图信息修改：

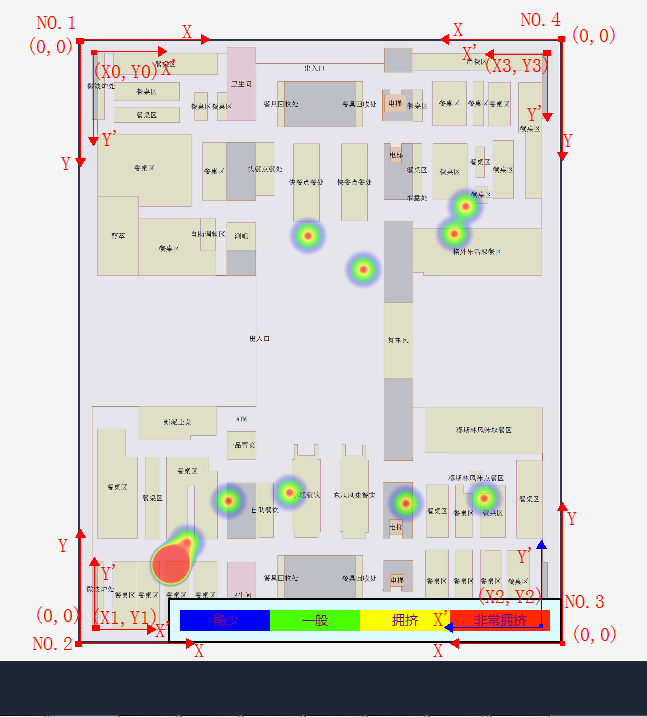
1. 进入【信息管理】 -- 【地图管理】页面；
2. 点击【编辑】按钮，编辑要修改的信息；
3. 修改后，点击【确定】按钮，保存修改后的地图信息；
4. 保存成功，修改后的地图信息会将原有的地图信息覆盖，从而添加到下面列表中。



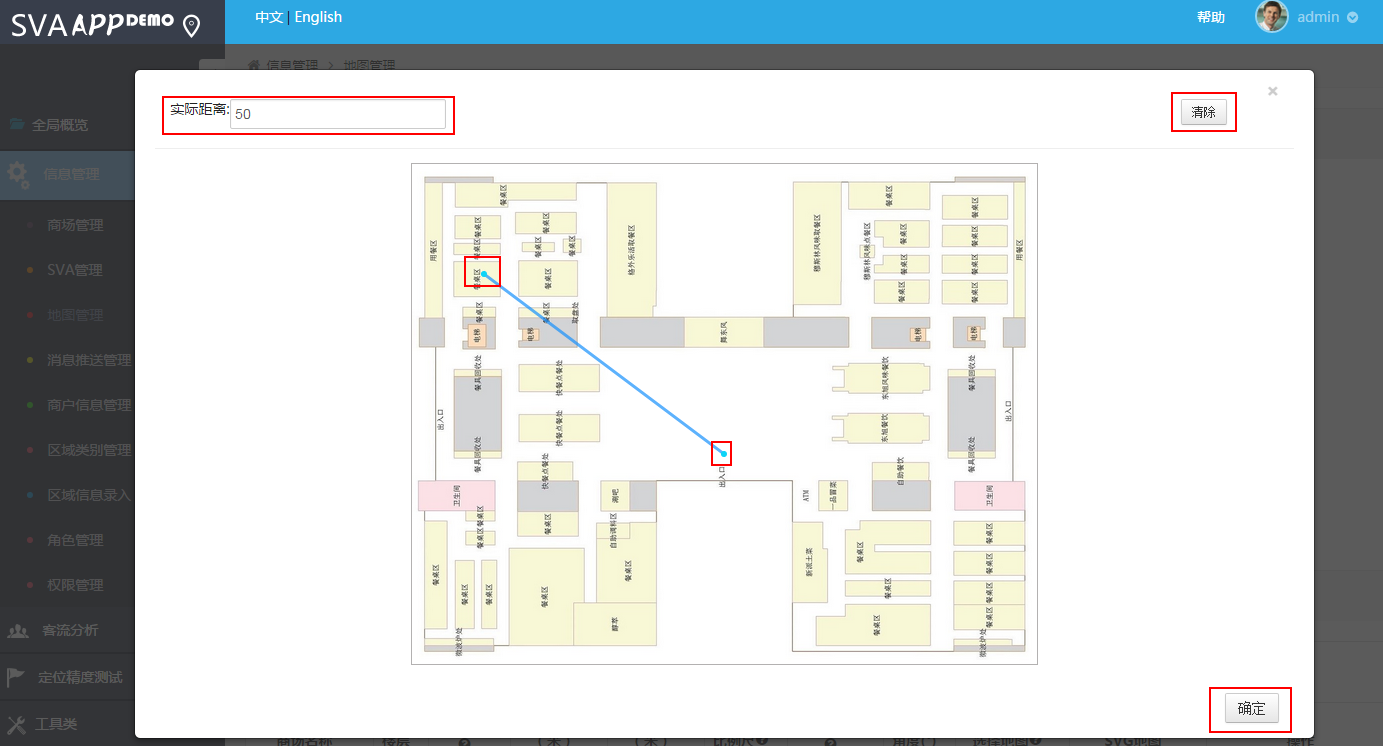
地图管理

说明：

1. 添加操作：点击【添加】按钮，弹出新增配置页面；
2. 商场名称：通过下拉框选择（商场名称由商场管理模块提供），例如U6食堂、王府井；
3. 楼层：可以由中文、字母、数字、下划线组成，最大长度为10位；
4. 楼层号：数字，支持附属，最大长度为8位，例：3或-3；
5. 原点位置：分别可以选择地图的4个角作为坐标原点（包含地图边框）。在地图原点位置选取图中，4个原点（0，0）分别对应为：左上角NO.1,左下角NO.2，右下角NO.3，右上角NO.4。左上角NO.1中（x0，y0）对应地图管理页面的x坐标原点和y坐标原点，为地图除去边框后的实际坐标原点。x0代表地图实际坐标原点到Y轴的距离，y0代表地图实际坐标原点到X轴的距离。通俗的解释为x0和y0分别代表Y轴方向的地图边框厚度和X轴方向的地图边框厚度。其他三个角的原理与NO.1的原理一样；

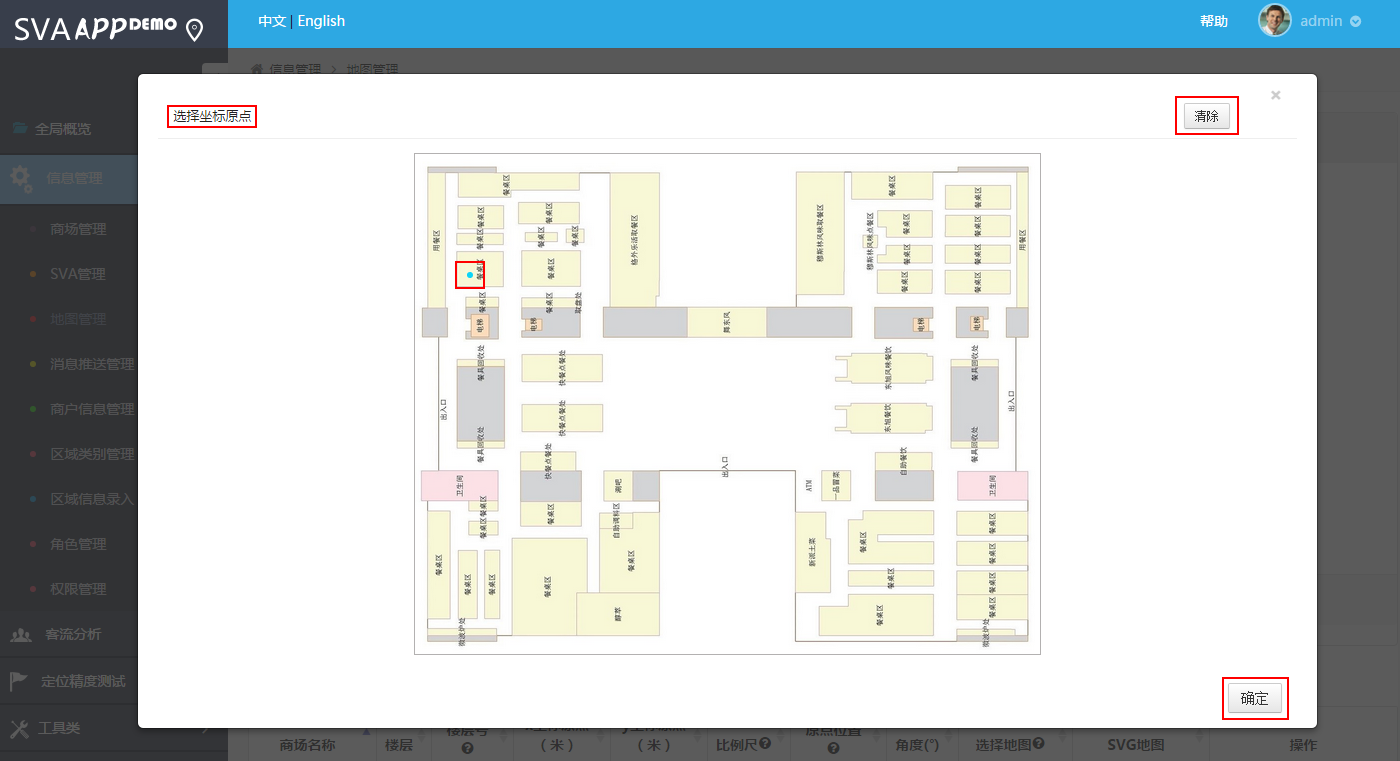


地图原点位置选取

1. 角度（非必填项）：可以输入0°- 360°，默认为0°；
2. 路径滤波文件：支持上传路径滤波文件，只能上传.xml格式的文件；
3. 路径规划文件：支持上传路径规划文件，只能上传.xml格式的文件；
4. 选择地图控件：只能上传.jpg和.png格式的文件；
5. 选择SVG地图控件：只能上传.SVG格式的文件；
6. 比例尺：像素大小与实际距离的比值。只能由数字和小数点组成，最大长度为15位（包括小数点）；比例尺的值：（1）手动输入；（2）通过后台计算得出：点击【选择】按钮，弹出地图页面，输入实际距离、在地图上选取两个点，点击确定得到比例尺的值；

比例尺图形化选择

1. 坐标原点：添加的地图周围会出现空白的无效区域，这里的坐标原点为实际地图的坐标原点，如：添加的地图原点选择在左上角，X轴方向地图边框厚度为1米，y轴方向厚度为2米，那么实际地图的坐标就为（2，1）。只能由数字组成，最大长度为10位（提供SVA地图方在提交地图时把x、y坐标原点在图中标出）。坐标原点值：（1）手动输入；（2）通过后台计算得出：点击【选择】按钮，弹出地图页面，在地图上选取坐标点，点击【确定】按钮得到坐标原点的值；



选取坐标原点

1. 确定操作：点击【确定】按钮，对添加的地图信息进行保存。保存后地图存储在服务器端，地图信息存放在数据库中；
2. 取消操作：点击【取消】按钮，取消对已添加的地图信息进行保存。取消添加后，列表中无新增地图信息；
3. 复制操作：点击【复制】按钮，复制选择的数据信息到添加页面中。并可在原基础上修改部分信息内容添加数据。
4. 编辑操作：点击【编辑】按钮，对已存在的地图信息进行修改；
5. 删除操作：点击【删除】按钮，确认删除后，则该条数据将从列表中和数据库中删除；并刷新页面展示最新的数据信息。

##### 选择菜单-客流分析-客流实时热力图

该模块主要用于实时监控某一楼层的客流量，某个位置的客流量越大，颜色越深。通过选择商场（支持下拉框）、楼层（支持下拉框）和输入时间（图中的1所示，单位为秒，由用户手动输入，控制热力图显示为当前时间的前几秒的热力图），点击【确定】按钮，可以统计当前地图上的总人数。

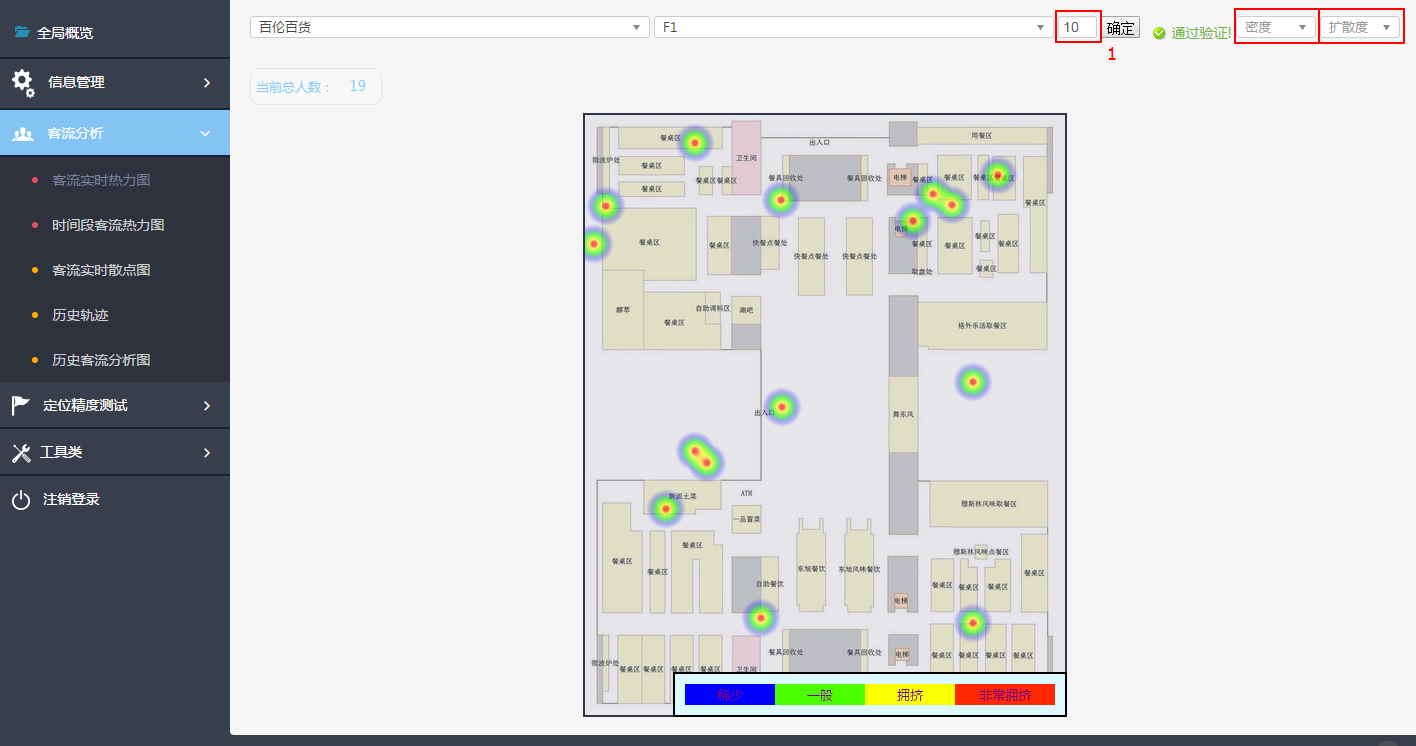
密度：分为1、2、3、4、5、10、15、20，8个等级，数值代表了每平方米有多少人，如：“1”代表了每平方米有一个人。当密度选择最低时，每个点的颜色都为红色。在点的密度不变的情况下，将密度选择为20，图中密度小于20的点颜色由深变浅。在选择的密度不变的情况下，点的密度增大，颜色会逐渐变深。

扩散度：分为20、30、40、50、100、150、200，7个等级，数值代表了描点时画出的圆的半径，如：“20”代表了地图中以手机传回的定位点的坐标为圆点，规划出来的范围半径为20像素值。当选择的扩散度增大时，图中的点的范围就增大。

地图右下角的为图例。图例是客流对应所选密度的颜色变化。

*注：*

*地图右下方的色块。如选择密度为10，那么稀少（蓝色）显示：人数小于1；一般（绿色）：人数小于4；拥挤（黄色）：人数大于7；非常拥挤（红色）：人数大于10。热力图每隔4s刷新一次，从数据库中获取最新的点进行展示。*



客流实时热力图

#### 问题定位方式

##### 问题一：热力图没有数据显示

第一步，查看sva对接是否成功

在SVA管理页面，对表格里目标SVA行点击编辑按钮，然后不做修改直接点击确定。如果不能成功并有错误信息提示，说明目标SVA出现问题，无法连接，需要联系对应的SVA负责人进行确认；如果编辑成功，说明SVA链路是通的，需要进行下一步排查确认。

第二步，查看订阅是否成功

打开app后台日志，在软件包路径\apache-tomcat-7.0.37\logs\run文件夹里的sva\_debug.log文件。找到最新一次的订阅，如果出现“[DEBUG] subscription result:{"result": {"error\_code": 4912322123, "error\_desc": "Param is not correct!"}}”的日志，说明订阅失败，根据里面的错误码和错误信息，向SVA负责人确认；如果出现“[DEBUG] subscription result:{"Subscribe Information": [{"APINAME": "locationstream", "APPID": "APP12", "BROKER\_IP": "59.36.10.38", "BROKER\_PORT": 4703, "ECHO\_TIMER": "0", "QUEUE\_ID": "APP12.36812b258b644eae977ac4d3ce4a9a48.27f06602"}], "result": {"error\_code": 0, "error\_desc": "Execution succeeded."}}”的日志，说明订阅成功，需要进行下一步的排查确认。

第三步，查看数据是否正常接收

还是上一步中的日志sva\_debug.log，在订阅成功之后的日志中，如果出现“[ERROR] [AMQP]No data from SVA,connection closed!”的日志，说明SVA没有数据上报，向SVA负责人确认是否有定位数据上报；如果出现类似“[DEBUG] SVA Data:{"locationstream":[{"IdType":"IP","Timestamp":1503575261000,"datatype":"coordinates","location":{"x":270,"y":546,"z":1},"userid":["0adecb76"],"map":{"mapid":"1025"}},]}”的数据上报日志，说明SVA上报正常，并且app接收也正常，但接收的数据没有正确存入数据库，需要确认下数据库的连接是否正确，数据库是否启动，数据库中的表结构是否正确。

#### 常见问题FAQ

##### Tomcat启动失败，提示address already in use!

端口冲突，先确认app使用的端口号，然后查看该端口号是否已被占用。若已被占用，考虑修改app的端口号或是冲突程序的端口号。

##### 登陆页面正常显示，输入用户名密码后登陆，页面跳转至错误信息页

数据库连接失败，请进行以下确认：

1.数据库是否启动；

2.app的数据库连接设置是否正确，端口、密码等信息是否正确；

3.app启动后是否很长时间没有使用（闲置超过一个月），此时需要重启app

##### 在SVA管理页面添加SVA信息时，点击确认后没有反应

SVA配置信息不正确，或是对应的SVA未处于工作中，此时需要和SVA负责人确认

##### 在地图管理页面上传地图失败

有以下可能：

1. 地图格式不对，仅支持png或是jpg格式的地图；
2. 地图尺寸过大，地图尺寸不超过5000\*5000像素，大小不大于20mb

### 对接案例



### 人员及联系方式

SVA：李学义 手机：

APP：范彬彬 手机：