# 概述

本文主要用于说明商户接入支付宝公众号demo(alipayDemo工程)流程及相关代码结构，希望能对准备接入支付宝公众号的商户提供帮助。

# 起步

## 环境准备

* Eclipse J2EE 版本:

<http://www.eclipse.org/downloads/packages/eclipse-ide-java-ee-developers/keplersr1>

* Eclipse M2eclipse插件：3.7开始直接可以在eclipse marketplace搜索安装
* Eclipse Tomcat插件：<http://www.eclipsetotale.com/tomcatPlugin.html>
* Apache maven: <http://maven.apache.org/download.cgi>
* Apache tomcat: <http://tomcat.apache.org/>

## 源代码导入

alipayDemo是使用maven进行工程构建，导入步骤如下：

* 下载附件alipayDemo，解压后在源代码路径输入mvn eclipse:eclipse;出现”BUILD SUCCESS” 表示工程构建完毕;
* 打开eclipse，”import” ->”Existing Projects into Workspace ”，导入构建成功的alipayDemo;
* eclipse导入工程成功后，会出现大概67个编译错误，别慌，主要是由于支付宝对外提供的SDK jar依赖导致；
* eclipse内选中工程，右键”Build Path” ->”Configure Build Path”,切换到”Libaries”->”Add Library”,在弹出列表里选择”Web App Libraries”，点击下一步，选中”alipayDemo”，点击”Finish”，编译工程，错误一扫而空~~

PS:部分同学可能遇到，导入alipayDemo后在eclipse呈现的是java project 而不是Dynamic Web Project。这个主要是由于project类型没有转换的缘故。解决方法是:选中”alipayDemo”右键”properties”，点击左侧”Project Facets”,选择右边配置，勾选”Dynamic Web Module”，版本勾选”2.5”，完成点击”OK”即可。

## 本地部署

提供两种本地部署方式：

* eclipse内部直接部署运行：安装eclipse tomcat插件成功后，在”server” 视图内直接添加”Apache”选中对应安装的tomcat,然后将alipayDemo添加进去，直接运行即可；
* 在alipayDemo源代码目录下，执行mvn package后，在target目录下会自动生成alipayDemo-1.0.0.war文件，复制放到tomcat安装目录下的webapps即可。

## 关于配置

支付宝体内存在两套环境：

* 沙箱环境：支付宝线下环境，只能用于联调和测试，所有的数据都是专用于测试，不能用于真实场景；
* 线上环境：支付宝线上环境，最终商户运行的环境，针对线上真实场景

为了方便的在两个环境进行切换，专门提供了两个配置和一个测试文件：

* SandBoxConfig.java: 用于配置支付宝线下环境，根据接口返回商户线下对应公众号ID，商户公钥，是否加签和验签，是否解密和加密等；
* OnlineConfig.java: 用于配置支付宝线上环境，根据接口返回商户线上对应公众号ID，商户公钥，是否加签和验签，是否解密和加密等；
* config.properties: 位于WEB-INF/config目录下，用于配置当前对应的支付宝环境配置及应用部署访问地址;

# 代码说明

## 类结构

整个alipayDemo工厂按照包名划分，主要分成两类:

* com.alipay.demo：demo所有业务相关代码；
* com.alipay.xml: xml格式解析工具类；

业务主要功能都在com.alipay.demo开头的相关包中实现，主要划分如下：

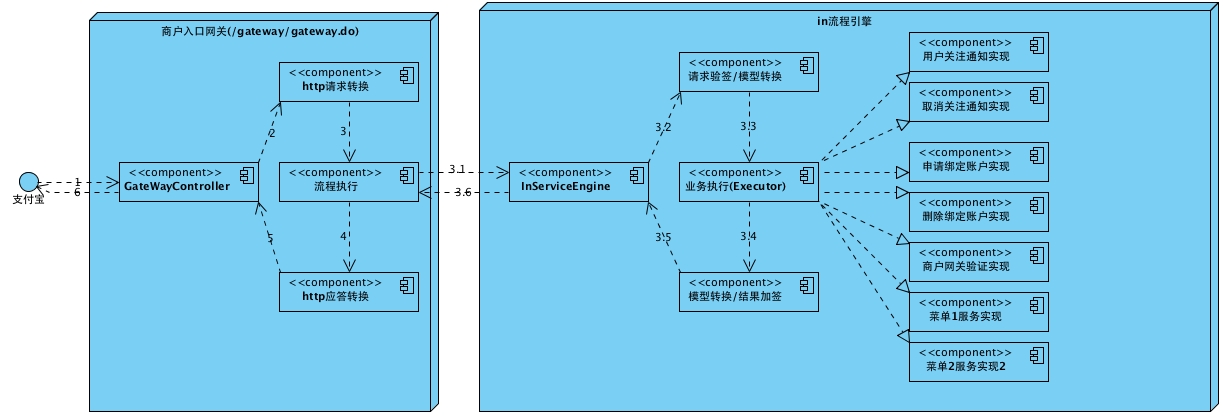
* in相关：支付宝主动请求商户相关；
* out相关：商户主动请求支付宝相关；
* controller： Web相关控制器实现：例如网关入口等;
* factory: 模型等相关工厂类;
* cache: 缓存实现
* config: 针对公众号的配置相关信息；针对支付宝沙箱和线上两套环境，提供不同的配置类；

## 流程说明

alipayDemo实现了两个最主要的流程：

### 支付宝主动请求商户

具体功能包括：商户网关验证，用户关注通知，用户取消关注通知，申请绑定外部户请求，删除绑定外部户通知，服务类菜单点击等。实现流程如下：



alipayDemo提供统一网关入口(实现类GatewayController,url是/gateway/gateway.do),业务逻辑由InServiceEngine提供实现。

InServiceEngine提供针对支付宝请求的实现流程模板，提供四个方法：

* doBefore

提供针对支付宝请求的验签和解密功能(根据com.alipay.demo.config下的具体配置执行),这部分商户可以不用扩展；

* doAction

支付宝不同请求是根据其请求参数内的msgType,eventType和actionParam组合来区分，因此根据这三个参数，找到对应具体业务实现类执行;

* doFail

当流程执行中出现异常时的统一处理逻辑，商户可根据自己身需要进行扩展;

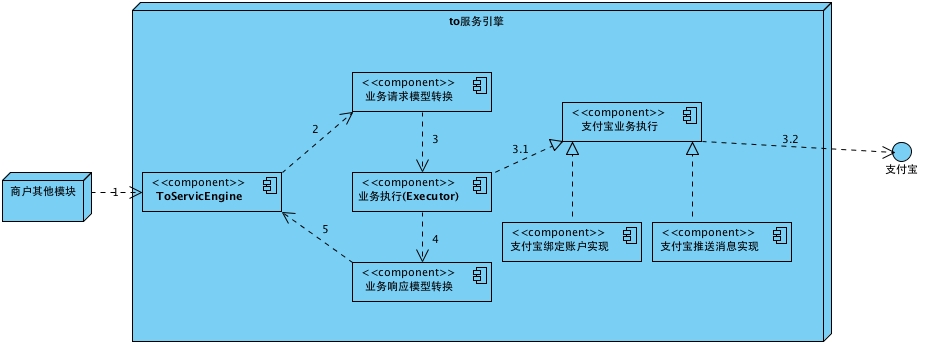
* doFinal

无论是正常执行or异常执行，统一最终处理响应操作；商户可根据自身进行扩展；

InServiceExecutor类封装不同业务的执行方法，根据支付宝请求的不同业务，继承该类进行实现；

### 商户主动请求支付宝

具体功能包括：添加绑定商户会员号接口，更新绑定商户会员号接口，菜单创建/更新/删除接口，商户推送消息接口，获取GIS地理位置接口，免登授权码获取接口，用户信息获取接口等。实现流程如下：



与支付宝请求商户类似，商户主动请求支付宝也采用流程模板机制实现，实现类为ToServiceEngine,其内部提供的方法与InServiceEngine一致，相应的执行器基类是ToServiceExecutor，但与InServieExecutor扩展方式略有不同:由于商户请求支付宝实际逻辑都是调用SDK内提供的接口实现，所以商户主动请求支付宝的Executor实现比较简单，只需要实现getExecutorName接口返回唯一实现标识。比较复杂的实现反而是根据不同业务实现对应请求和响应模型转换(对应InsServiceEngine内doBefore和doFinal方法)。

# 最后

如本文开始说明的一样，alipayDemo是能商户接入支付宝公众号的样例代码，希望能给快速接入支付宝公众号的商户提供帮助。

您的使用是对我莫大的荣幸，支付宝公众号的未来缺你不可！