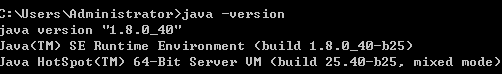
# 软件安装过程

1. JDK1.8安装

并设置好环境变量

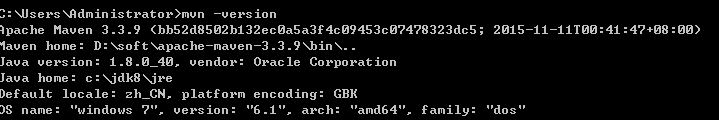


<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

下载。 看机器是32还是64位的， JDK和eclipse等需要都下载32或64的。

1. apache-maven-3.3.9安装

设置好环境变量。Setting.xml自己配置，下面的仅是参考。Conf目录下



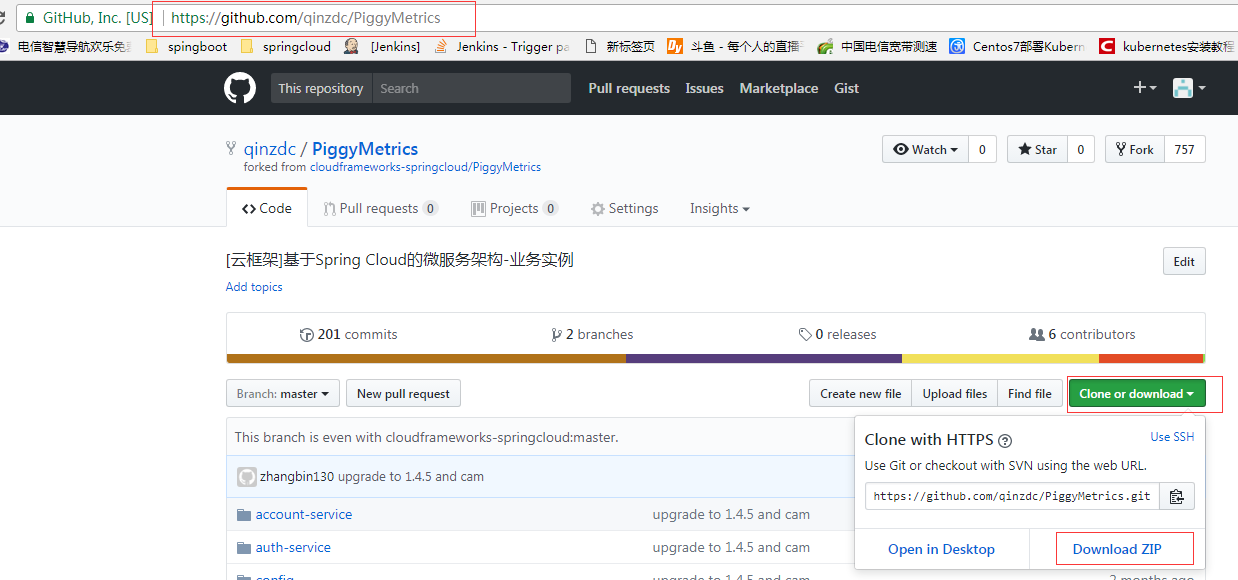
1. eclipse安装

略

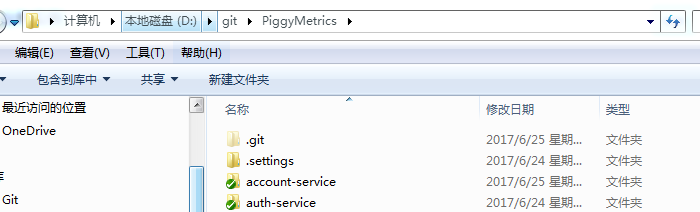
1. 代码下载

<https://github.com/qinzdc/PiggyMetrics>

也可以自己使用git下载。



解压缩后如下图

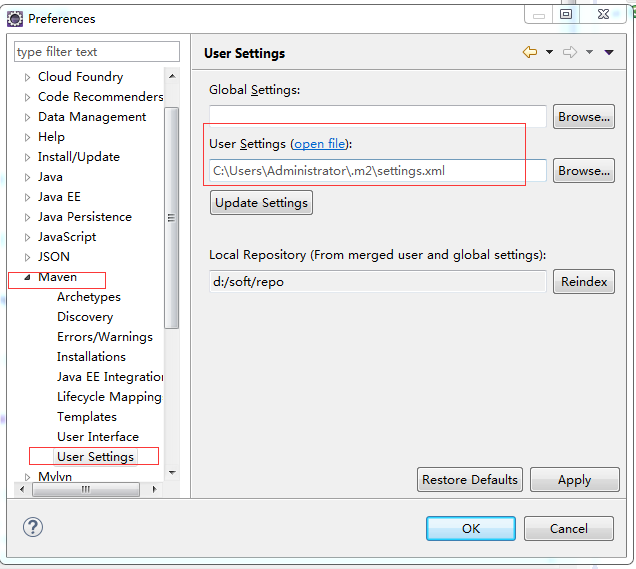


1. 环境检查

在D:\git\PiggyMetrics\account-service目录下执行mvn clean spring-boot:run看控制台是否报错。报错处理掉。

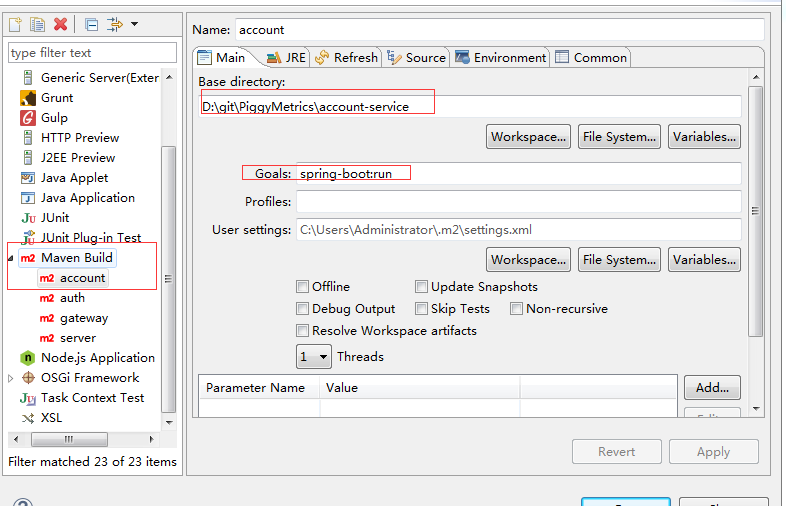
1. Eclipse导入工程编译

注意将eclipse里的maven的setting.xml也要使用和自己maven的一样的。



1. Eclipse里能编译通过后就可以编码了。
2. 代码调试

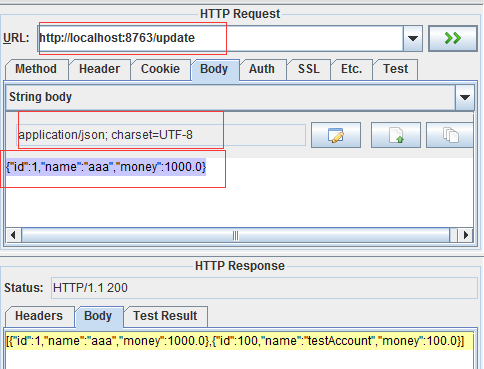
快速在eclipse启动应用。



模拟前端发请求restclient

<https://github.com/wiztools/rest-client/releases/download/3.6.1/restclient-cli-fat-3.6.1.jar>

注意是POST



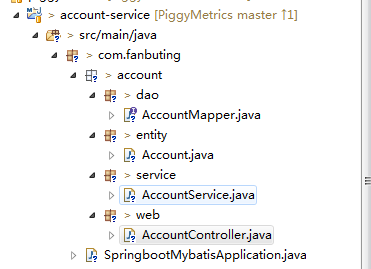
1. 打包
2. 在对应的微服务目录下执行mvn clean install
3. 在对应target目录下会有jar生成，jar就是最终的应用
4. 执行java -jar xxx.jar即可启动应用。

# 代码框架---单服务

1. 综述
2. 使用 spring boot,
3. 持久化使用mybaties，
4. 日志集成logback,

只看account-service工程即可， 其他忽略

1. 代码结构



1. 服务发布

AccountController.java里

1. 调用mybaties
2. AccountMapper.java
3. 使用注解和使用xml 2种方式

# 代码框架---微服务（忽略）

1. 背景

使用spring boot,springcloud，均使用比较简单，建议花2~4小时阅读下面网页

Spring boot: <https://github.com/qinzdc/SpringBootLearning> 看入门和配置、mybaties即可

Spring cloud: <https://github.com/qinzdc/SpringCloudLearning> 1~5篇看看即可。

1. 启动后应用端口和地址等

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| 注册中心 | registry | <http://localhost:8761/> |  |
| 服务网关 | Gateway | <http://localhost:8769> |  |
| 账号服务 | Accout-service | <http://localhost:8763> | 通过这个端口可以调用内部微服务  JDBC配置等也在\*.yml里。 |
|  |  |  |  |
|  |  |  |  |

每个应用的端口等配置在对应的src\main\resources\\*.yml里，可以自行修改。

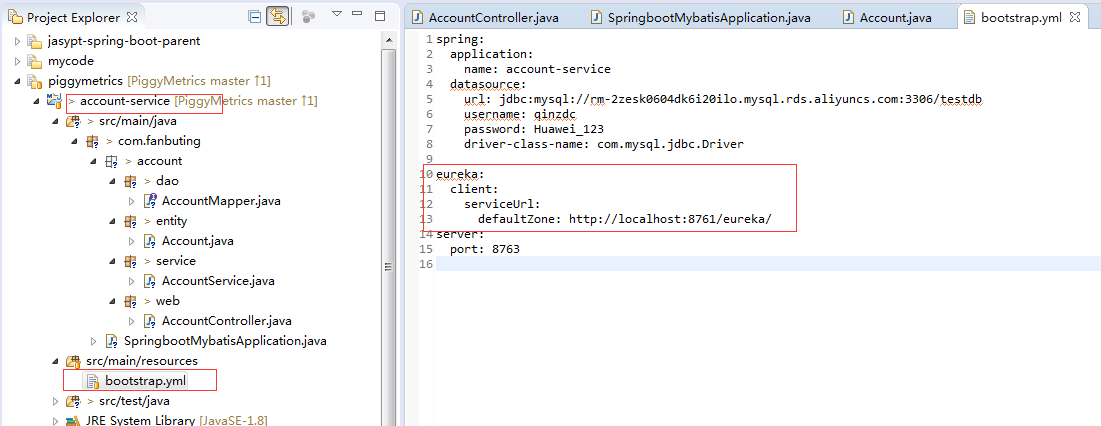
最终对前端暴漏<http://IP:8769/auth/xxxx>等接口。 其他端口建议公网不对外暴漏。

**启动顺序**：先启动注册中心，然后启动其他微服务。

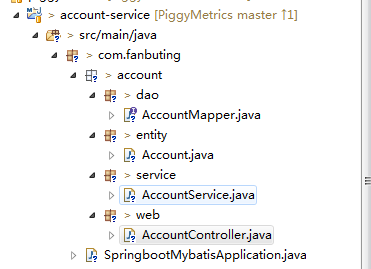
1. 单模块，不使用微服务

如果只搞一个模块，不搞微服务， 那只使用account-server.jar即可

使用前将\*.yml里下面红色框里删除启动即可。



1. 微服务代码结构



1. 微服务内部发布接口

AccountController.java里

1. 系统对外部暴漏最终接口

Gateway工程的yml里. 如果是单服务，就直接用单服务的端口和地址

1. 微服务互相调用
2. 使用mysql数据库交互

AccountMapper.java

使用注解和使用xml 2种方式

1. 集群支持

系统如果要支持集群,对外启动多个gateway,然后nginx负载。

内部微服务之间集群，已经可以自己管理。

后续有诉求再详细看。