接口自动化

开发环境配置及简单说明

|  |  |  |  |
| --- | --- | --- | --- |
| 编制 | 沈兵 | 日期 | 2017-03-10 |
| 评审 |  | 日期 |  |
| 密级 | 受控 |

修订记录

| 版本 | 作者 | 日期 | 修订记录 |
| --- | --- | --- | --- |
| V1.0 | 沈兵 | 2017/03/10 | 创建 |
|  |  |  | 修改涉及XXXX部分内容 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

[1 开发环境搭建 4](#_Toc8568)

[1.1 Jdk安装配置 4](#_Toc17242)

[1.2 Maven安装配置 4](#_Toc3876)

[1.3 Eclipse安装配置 5](#_Toc31722)

[1.4 Testng插件安装配置 5](#_Toc22487)

[2 工程创建及配置 5](#_Toc25917)

[2.1 Maven工程导入 5](#_Toc439)

[2.2 工程配置依赖包 6](#_Toc9655)

[2.3 工程配置文件 6](#_Toc14108)

[3 开发步骤说明 7](#_Toc3505)

[3.1 测试数据 7](#_Toc8405)

[3.2 测试代码 8](#_Toc15662)

[4 类常用方法说明 8](#_Toc22407)

[5 用例组织 8](#_Toc10308)

[6 用例开发原则 9](#_Toc19685)

## 1 开发环境搭建

### 1.1 Jdk安装配置

1. svn下载JDK，svn路径：<https://10.1.10.5:8443/svn/codyy/trunk/document/测试部/对内/00_部门管理/工作软件/开发工具/jdk-8u102-windows-x64.exe>
2. 双击exe，傻瓜式完成安装jdk和jre。
3. 环境变量配置：

变量名：JAVA\_HOME

变量值：jdk安装目录

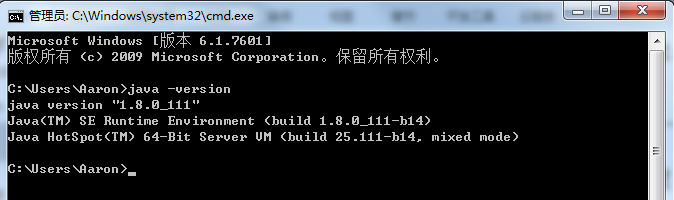
变量名：CLASSPATH

变量值：.;%JAVA\_HOME%\lib\dt.jar;%JAVA\_HOME%\lib\tools.jar;

变量名：Path

变量值：%JAVA\_HOME%\bin;%JAVA\_HOME%\jre\bin;

1. 安装验证：进入cmd环境，输入”java -version”,显示java版本即安装成功。



### 1.2 Maven安装配置

1. Svn下载maven，svn路径：<https://10.1.10.5:8443/svn/codyy/trunk/document/测试部/对内/00_部门管理/工作软件/开发工具/apache-maven-3.0.4.rar>
2. 解压缩到指定目录XXX。
3. 环境变量配置：

变量名：MAVEN\_HOME

变量值：XXX/apache-maven-3.0.4(绝对路径)

变量名：Path

变量值：%MAVEN\_HOME%\bin;

1. 安装验证：进入cmd环境，输入”mvn -version”,显示Maven版本即安装成功。



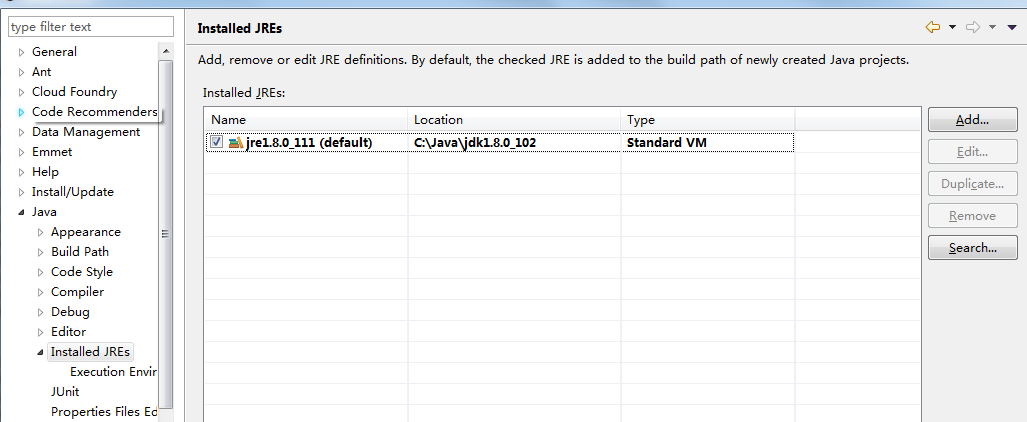
1. maven的setting.xml配置项中增加私有仓库信息，保存后放入maven仓库目录下：



（该文件保存到本地仓库目录）

### 1.3 Eclipse安装配置

1. Svn下载eclipse，svn路径：<https://10.1.10.5:8443/svn/codyy/trunk/document/测试部/对内/00_部门管理/工作软件/开发工具/eclipse-neon.rar>
2. 解压缩到指定目录，双击eclipse-neon.exe，启动eclipse。
3. 配置jre,点击Window--Preferences--java--Installed JREs，增加1.1安装的jre。



### 1.4 Testng插件安装配置

下载的Eclipse已经安装。

## 2 工程创建及配置

### 2.1 Maven工程创建

略

### 2.2 工程配置依赖包

* 私有maven仓库下载，仓库地址：

<http://10.5.233.31:8081/nexus/content/repositories/releases/>

* 引入接口开发包：

<dependency>

<groupId>com.codyy.test</groupId>

<artifactId>autointerface</artifactId>

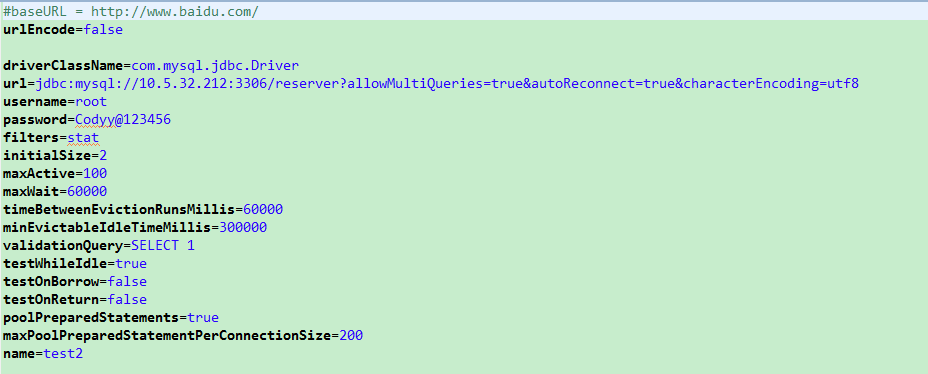
<version>1.1.0</version>

</dependency>

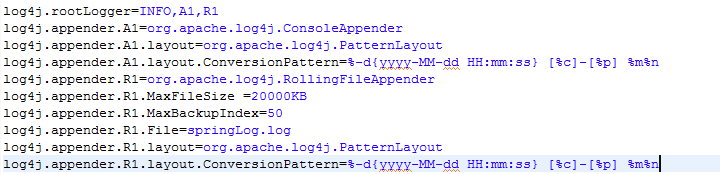
* 其他包按需引入，如数据库驱动包、日志等

### 2.3 工程配置文件

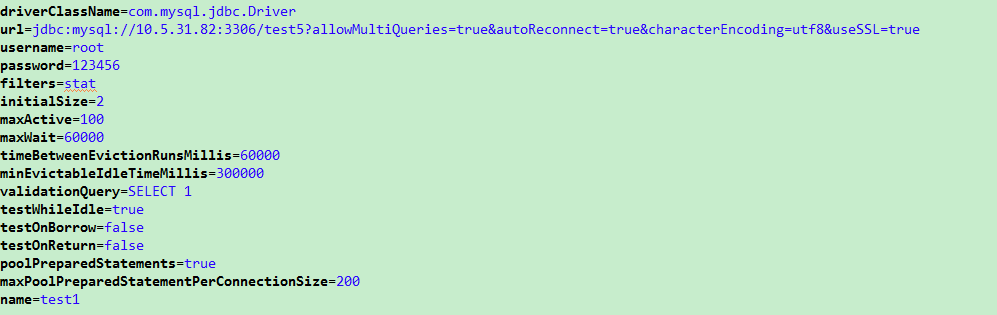
1. Config.properties文件



1. log4j.properties文件



1. 数据库文件内容格式：



## 3 开发步骤说明

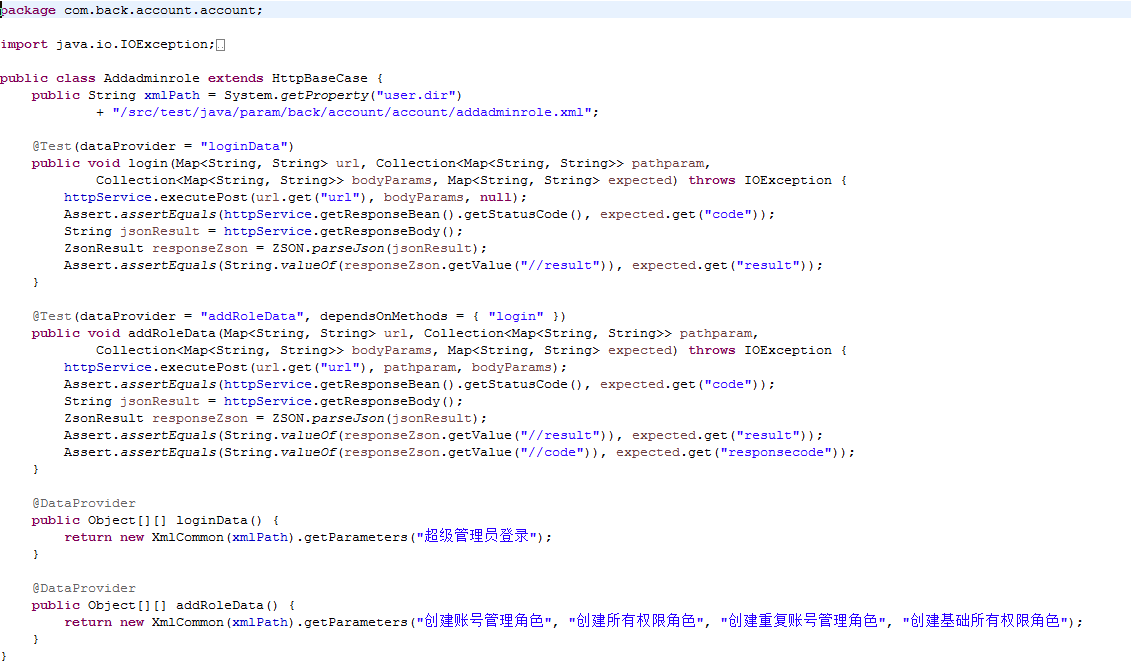
### 3.1 测试数据



说明：

1. 测试数据使用xml存储，xml根节点为map。
2. 每一个bean提供给一个或多个测试方式使用。
3. 每个bean中分参数paramters和断言数据asserts两部分。
4. paramters中有3个不同名tag，每个bean有且一个url标签，标记接口访问路径； 多个pathparam标签，标记请求的路径参数；多个bodyparam标签，标记请求的body 数据（基本都在Post请求时使用）。
5. Asserts中有1个tag,可以包含多个assert标签，标记多个断言点。

### 3.2 测试代码



说明：

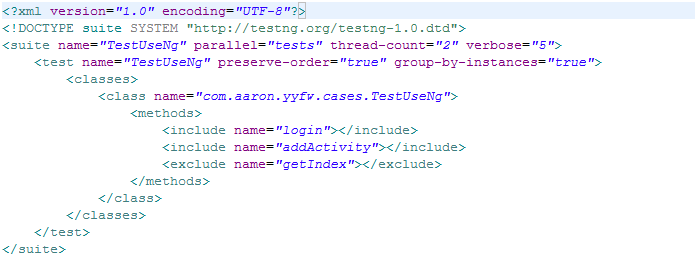
1. 测试代码结构简单，继承基本测试类HttpBaseCase。
2. 测试代码基于testng开发，关于testng的使用如有不清楚，可以咨询帅气的沈兵。
3. 一个接口需要执行多次时，只需要提供多个测试数据即可。

## 4 类常用方法说明

基本使用的类包括数据解析类XmlCommon、接口服务类HttpService、json解析类ZSON、json解析结果类ZsonResult、数据库访问类DbProxoolUnit、断言类Assert。对于类方法的使用可以咨询帅气的沈兵。

## 5 用例组织

用例组织是通过testng的xml配置文件实现的，下图展示了一个测试用例的基本配置。更多信息咨询帅气的沈兵。



## 6 用例开发原则

1. 类名、参数文件名使用接口名定义。
2. 每个测试用例中依赖的外部数据原则上在本测试类中生成，如果依赖其他测试类生 成的数据，在该类中注解说明依赖于哪个类的哪个测试方法。
3. 一个测试类生成的数据原则上本测试类中清除。