# JAVA接口架构

## 基本说明

以太坊官方工具是nodejs，通用web3.js封装了接口，java版本的web3j包是参考web3.js的集成包，方法的调用风格比较偏向js和lambda。

web3j包是java8写的，所以spring集成版本一般用java8兼容的。

实测版本：Spring-boot-starter 2.3.5-Release可用

实测版本：2.4.5集成就有问题，springmvc版本不统一。

## Pom结构

用Spring initializer生成一个spring boot 2.X的版本（当前默认最新稳定是2.7.10，不能用），然后替换pom即可。

```

*?*xml version**="1.0"** encoding**="UTF-8"***?>*<project xmlns**="http://maven.apache.org/POM/4.0.0"** xmlns:xsi**="http://www.w3.org/2001/XMLSchema-instance"** xsi:schemaLocation**="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd"**>  
 <modelVersion>4.0.0</modelVersion>  
 **<!-- https://mvnrepository.com/artifact/org.springframework.boot/spring-boot-starter -->** <parent>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-parent</artifactId>  
 <version>2.3.5.RELEASE</version>  
**<!-- <groupId>org.springframework.boot</groupId>-->  
<!-- <artifactId>spring-boot-starter-parent</artifactId>-->  
<!-- <version>2.4.5</version>-->  
 <!--至少从245这个版本开始就不行了-->** <relativePath/> **<!-- lookup parent from repository -->** </parent>  
 <groupId>com.example</groupId>  
 <artifactId>demo</artifactId>  
 <version>0.0.1-SNAPSHOT</version>  
 <name>demo</name>  
 <description>demo</description>  
 <properties>  
 <java.version>1.8</java.version>  
 </properties>  
 <dependencies>  
  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter</artifactId>  
 </dependency>  
  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-test</artifactId>  
 <scope>test</scope>  
 </dependency>  
 **<!-- https://mvnrepository.com/artifact/org.web3j/core -->** <dependency>  
 <groupId>org.web3j</groupId>  
 <artifactId>web3j-spring-boot-starter</artifactId>  
 <version>4.0.3</version>  
 </dependency>  
 <dependency>  
 <groupId>org.web3j</groupId>  
 <artifactId>core</artifactId>  
 <version>4.9.8</version>  
 </dependency>  
 **<!-- https://mvnrepository.com/artifact/com.squareup.okhttp3/okhttp -->** <dependency>  
 <groupId>com.squareup.okhttp3</groupId>  
 <artifactId>okhttp</artifactId>  
 <version>4.10.0</version>  
 </dependency>  
  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
 </dependency>  
 **<!-- https://mvnrepository.com/artifact/com.google.code.gson/gson -->** <dependency>  
 <groupId>com.google.code.gson</groupId>  
 <artifactId>gson</artifactId>  
 <version>2.10.1</version>  
 </dependency>  
  
 **<!-- https://mvnrepository.com/artifact/org.web3j/web3j-maven-plugin -->** <dependency>  
 <groupId>org.web3j</groupId>  
 <artifactId>web3j-maven-plugin</artifactId>  
 <version>4.9.4</version>  
 </dependency>  
  
 <dependency>  
 <groupId>org.web3j</groupId>  
 <artifactId>web3j-sokt</artifactId>  
 <version>0.2.2</version>  
 </dependency>  
  
 </dependencies>  
  
  
  
 <build>  
 <plugins>  
 <plugin>  
 <groupId>org.web3j</groupId>  
 <artifactId>web3j-maven-plugin</artifactId>  
 <version>4.9.4</version>  
 <dependencies>  
 <dependency>  
 <groupId>org.web3j</groupId>  
 <artifactId>web3j-sokt</artifactId>  
 <version>0.2.2</version>  
 </dependency>  
 </dependencies>  
 <configuration>  
 **<!-- 指定Java版智能合约生成的位置 -->** <packageName>com.example.demo.smartContract</packageName>  
 <soliditySourceFiles>  
 **<!-- solidity源文件放置位置 -->** <directory>src/main/resources/solc</directory>  
 <includes>  
 **<!-- 只将后缀为.sol的文件包括进去 -->** <include>\*\*/\*.sol</include>  
 </includes>  
 </soliditySourceFiles>  
 <outputDirectory>  
 <java>src/main/java</java>  
 </outputDirectory>  
 </configuration>  
 </plugin>  
 </plugins>  
 </build>  
  
</project>

```

Plugin中的web3j-maven-plugin，在此主要用于将solidity源文件.sol生成JAVA实体类的工具包，也有其它功能。在pom.xml位置（也就是src目录位置），使用以下命令，可读取sol文件生成到<packageName>标签指定的实体存放目录。

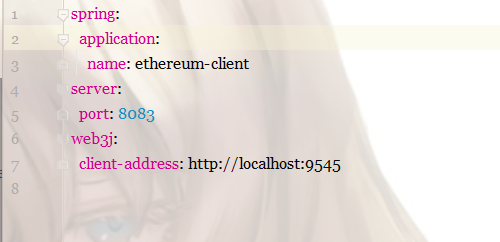
web3j-sokt0.2.2是对0.2.1的升级版本，旧版不支持solidity0.7以后的windows版本的sol编译工具自动调用。

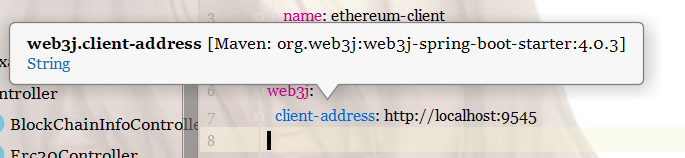
## 项目结构



application.yml配置

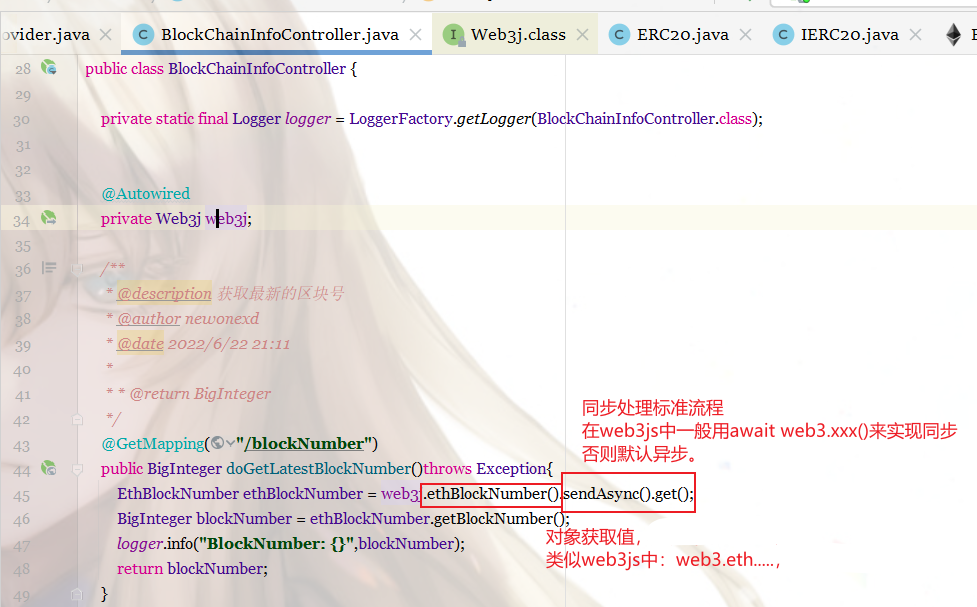
web3j.client-address就是区块链（以太坊节点）的地址，9545为truffle测试链的默认地址

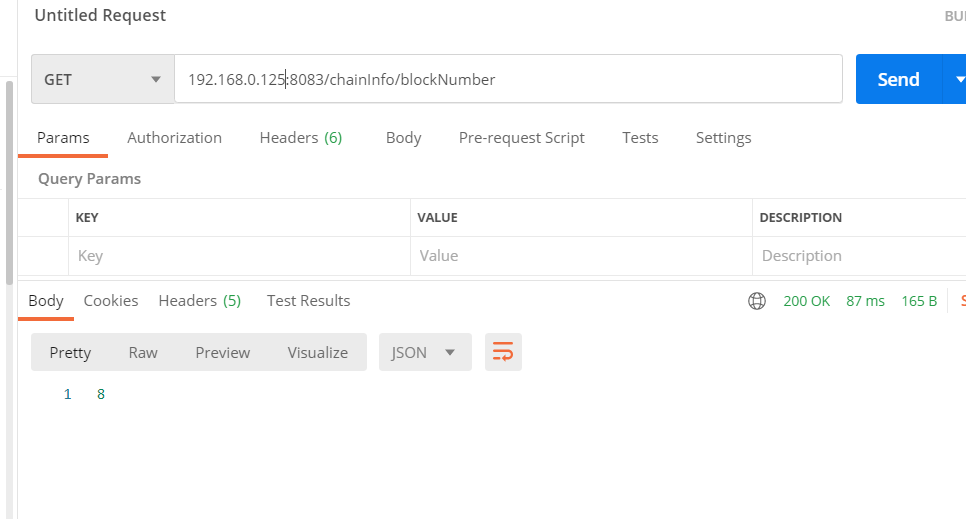


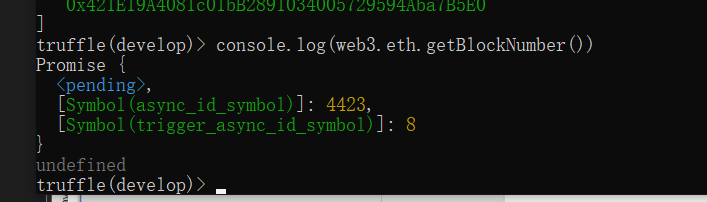


这个包好就好在，启动的时候自动根据client-address生成web3j，用的时候直接Autowired注入就可以用了，不用自己声明，但也应该是这一点导致和别的mvc版本冲突。

## 接口示例：链







确实是8个。

## 接口示例：合约





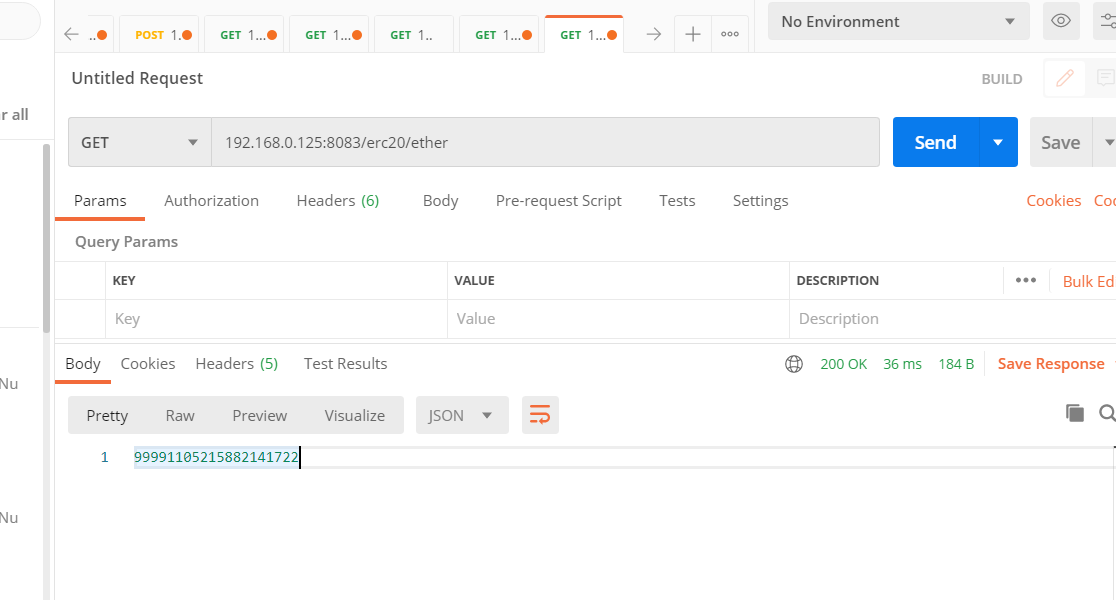
Chatgpt解释：

在以太坊中，随着新交易被添加到网络中，区块链的状态会不断变化。每个区块包含已验证并添加到账本中的一组交易。

当您想要检查以太坊帐户的余额时，需要指定块编号或块参数，以指示要用作余额计算参考点的块。这是因为帐户的余额随着新交易被添加到区块链上而发生变化。

总结：

默认的区块参数是“Latest”，也就是当前最新状态。



单位：wei，也就是99.9911…个eth