Python SDK的前世今生

原创 陈宇杰 FISCO BCOS开源社区 2019-12-26

陈宇杰

FISCO BCOS 核心开发者

人生重要的不是所站的位置，而是所朝的方向

AUTHOR 作者

今年6月，在已有Java SDK的基础上，FISCO BCOS相继推出了Python SDK和Go SDK，不少开发者表示，这两个组件极大降低了开发区块链应用的难度，让更多开发者可以参与到FISCO BCOS的开源共建中。

今天，本文就来科普一下Python SDK的前世今生，看看这个简单易上手又发挥着极大作用的组件是如何实现功能的。

FISCO BCOS

Why: 为什么要做Python SDK

众所周知，Python语言简单明了，几乎接近自然语言，可以快速上手；Java语言虽然功能强大，但语法、组件相对复杂，一般会在企业级应用中使用。为了融合两者优势，FISCO BCOS首席架构师张开翔经过两周集中开发，成就了Python SDK的第一个版本。

这个版本在今年7月份社区的黑客松比赛中发挥了极大作用，帮助参赛者更好聚焦在开发项目本身。

在高强度的黑客松比赛中，参赛者只有36小时去完成区块链相关项目。在和选手交流中，我们发现大多数极客开发者均是从比特币、以太坊等公链开始入门，对Java语言不是很熟悉，反而精通Python、Go、Node.JS。

当时团队主推Java SDK、控制台和WeBASE等基于Java的区块链开发工具，可是短短36小时，要求不熟悉Java的极客开发者基于这些工具完成开发，确实有些强人所难。在这种情况下，已经初见雏形Python SDK就成了开发者的救命良药。最终，很多参赛项目就是基于Python SDK完成的。

What: Python SDK实现了哪些功能

Python SDK经过不同版本的演进优化，逐步实现丰富的功能。

基础版本Python SDK

第一版的Python SDK主要基于Windows平台开发，实现了访问FISCO BCOS链RPC服务的接口，通过Python SDK接口，用户可以获取区块链的基本信息(如区块高度、区块、交易、交易回执等)，还可以部署和调用合约。

但该版本的SDK不支持Channel协议，因此无法保证SDK与节点间通信安全；且Python SDK无法接收FISCO BCOS区块链节点推送的回执、区块高度信息，因此发送交易后，必须轮询节点获取最新信息。

支持Channel协议的Python SDK

为了提升SDK与节点通信的安全性，Python SDK在基础版本上，实现了Channel协议。支持Channel协议后的SDK在发送交易后，不需轮询节点获取交易执行结果，直接从节点接收执行结果的推送即可，这样SDK发送交易后，不会阻塞在状态轮询，实现了异步发送交易的功能。

支持Precompile合约调用

为了突破区块链虚拟机性能瓶颈，大幅度提升交易处理吞吐量，FISCO BCOS实现了预编译合约框架，现有的预编译合约包括命名服务合约、共识合约、系统配置合约、权限控制合约、CRUD等，为方便用户使用这些预编译合约，Python SDK提供了访问Precompile合约的接口。

Python SDK控制台

以上功能ready后，Python SDK已经可以实现与FISCO BCOS链的所有交互，但用户仍无法通过Python SDK直观感受FISCO BCOS链，于是在实现以上所有接口的基础上，Python SDK集成了控制台console，console通过命令去调用以上接口，用户可通过命令行获取区块链节点信息、部署和调用合约。

Python SDK多平台支持

众所周知，Python是跨平台语言，支持Windows、MacOS、 CentOS、Ubuntu等各种开发平台。而Python SDK是基于Windows系统开发的基础版本，因此部署在MacOS、CentOS系统还有一些问题，比如说：各平台依赖包要求不一致、符合Python SDK版本的Python安装方式有差异、合约编译器安装方式不相同等。

以上这些问题，会使部分开发者的精力在安装基本环境上消耗殆尽，为了解决这个问题，Python SDK提供了部署脚本init_env.sh，用户可使用该脚本，一键初始化Python SDK环境。

脚本init_env.sh的主要功能包括：

- 若运行Python SDK的系统Python版本小于3.6.0，安装3.7.3版本的Python虚拟环境python-sdk；
- 安装v0.4.25版本的solidity编译器。

How: Python SDK如何实现

上节介绍了Python SDK的主要功能，这节就来说说Python SDK的实现。

罗马不是一日建成的，Python SDK在开发过程中参考了以太坊客户端部分模块，包括：

- ABI编解码模块
- RLP编解码模块
- 账户和密钥生成模块

以太坊客户端地址：
https://github.com/ethereum/web3.py

实现Channel协议

Python SDK在以上模块的基础上，实现了支持FISCO BCOS Channel协议的RPC接口和发送交易接口。

Channel消息包类型定义实现于模块client.channelpack.ChannelPack，主要消息包类型包括：

- 0x12: RPC类型消息
- 0x13: 节点心跳消息包
- 0x30: AMOP请求包
- 0x31: 失败的AMOP消息的包体
- 0x32: SDK topics
- 0x35: AMOP消息包包体
- 0x1000: 交易上链通知(包含交易回执)
- 0x1001: 区块高度推送包

Channel协议的实现位于client.channel.handler.ChannelHandler模块，为了支持异步发送交易，Python SDK引入了pymitter和Promise组件，用于管理异步事件，该模块发送交易并获取交易回执的主要工作流程如下：

- 1. Python SDK调用sendRawTransactionAndGetReceipt接口，对交易进行编码和签名，并将编码好的数据作为ChannelHandler模块发送socket请求包的参数；
- 2. ChannelHandler接收到1的数据后，将数据包的uuid注册到异步事件队列后，将消息包放入发送缓冲区发送并发送；
- 3. 节点处理完交易后，向SDK发送类型为0x1001的交易上链通知；
- 4. SDK端ChannelHandler收到消息类型为0x1001的回包后，取出消息包的uuid，并触发该uuid对应的异步事件，将交易执行结果返回给上层应用。

实现控制台

Python SDK控制台实现位于console模块，主要调用了一下模块接口实现与区块链的交互：

- client.bcosclient：Python SDK基本接口类，提供了访问FISCO BCOS RPC服务的接口、部署和调用合约的接口；
- console_utils.precompile：提供了访问Precompile预编译合约的功能；
- console_utils.rpc_console: 封装了client.bcosclient，实现命令行调用RPC服务功能。

Python SDK使用展示

了解了Python SDK的由来、功能和实现，下面从控制台角度直观感受下Python SDK:

获取节点版本

命令行终端输入./console.py getNodeVersion可获取节点版本：

部署HelloWorld合约

Python SDK内置HelloWorld合约，命令行终端输入./console.py deploy HelloWorld可部署HelloWorld合约，若用户需要部署自编写合约，则需将该合约放置于contracts子目录，并使用命令./console.py deploy 【合约名】来部署合约。

下图是HelloWorld合约部署输出，从中可看出，合约地址为:
0xbbe16a7054c0f1d3b71f4efdb51b9e40974ad651

调用HelloWorld合约

Python SDK控制台通过sendtx子命令发送交易，使用call子命令调用合约常量接口。

HelloWorld合约代码如下，主要包括set和get两个接口，前者用于设置合约局部变量name，后者是常量接口，获取局部变量name的当前值。

```
pragma solidity^0.4.24;
contract HelloWorld{
  string name;
  constructor() public{
      name ="Hello, World!";
  }
# get interface
   function get() constant public returns(string){
      return name;
  }
# set interface
   function set(string n) public{
      name = n;
  }
}
```

调用HelloWorld set接口

由于set接口改变了合约状态，因此使用sendtx子命令调用，sendtx用法如下：

```
./console.py sendtx [contract_name][contract_address] [function][args]
```

参数包括：

- contract_name：合约名
- contract_address：合约地址
- function：函数接口
- args：参数列表

使用./console.py sendtx HelloWorld 0xbbe16a7054c0f1d3b71f4efdb51b9e40974ad651 set "Hello,Fisco"命令可将HelloWorld合约的name成员变量设置为"Hello, Fisco"，输入如下：

调用HelloWorld get接口

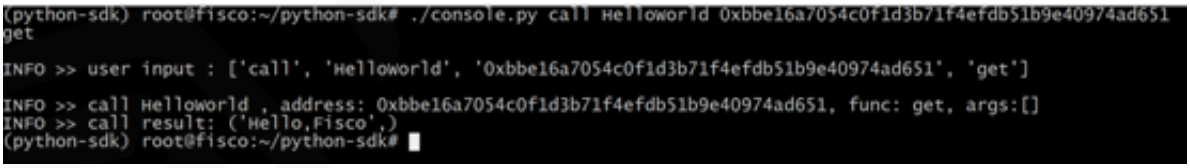
HelloWorld的get是常量接口，因此使用call子命令调用，call用法如下：

```
./console.py call [contract_name] [contract_address] [function] [args]
```

参数包括：

- contract_name：合约名
- contract_address：调用的合约地址
- function：调用的合约接口
- args：调用参数

使用./console.py call HelloWorld 0xbbe16a7054c0f1d3b71f4efdb51b9e40974ad651 get获取HelloWorld合约name成员变量的最新值：



小结

本文介绍了Python SDK的前世今生，包括Python SDK的由来、功能和实现，Python SDK详细使用方法可参考：
https://fisco-bcos-documentation.readthedocs.io/zh_CN/latest/docs/sdk/python_sdk/index.html。

当然，目前Python SDK还处于不断完善过程，欢迎社区爱好者关注相关issue，为Python SDK的优化贡献宝贵PR。相关参考：

https://github.com/FISCO-BCOS/python-sdk/issues