EOS DAPP实战教程

如何开发一款基于EOS的区块链游戏

郭金宏 2018.5

课程主要内容介绍

- 1、简介EOS环境搭建，及DAPP完整开发流程简要说明
- 2、创建前端源码框架，编写前端源码
- 3、创建智能合约目录，编写智能合约源码
- 4、前端调用eosjs，与智能合约交互
- 5、详解eosjs查询智能合约的table信息
- 6、详解eosjs调用智能合约接口
- 7、课程总结

课程说明

- 一共7节课
- 每节课尽量不超过10分钟
- 第一课时主要讲理论部分
- 第二课时开始进入实战环节

课程产出

一、简介EOS环境搭建，简介DAPP完整开发流程

本节课内容说明：

- 开发环境说明
- 搭建开发环境
- 开发环境目录说明
- 简要说明一个包含前后端的DAPP的主要开发流程
- 课程内容说明
- 总结

开发环境说明

- 推荐Mac OS、Ubuntu、Centos
- Windows系统，需要使用虚拟机
- 内存需要至少8GB

搭建开发环境

- 获取源码： git clone https://github.com/EOSIO/eos --recursive
- 开始编译：进入eos目录，执行./eosio_build.sh
- 构建可执行环境：进入build目录，执行sudo make install
- 注意：以上3个步骤，可能会遇到N个坑，可能会耗费不止一天时间。

开发环境搭建成功提示

恭喜！

可以开启DAPP开发之旅了！

```
[ 99%] Building CXX object tests/CMakeFiles/chain_test.dir/tests/database_tests.cpp.o
[ 99%] Building CXX object tests/CMakeFiles/chain_test.dir/tests/misc_tests.cpp.o
[ 99%] Linking CXX executable cleos
[ 99%] Built target cleos
[ 99%] Building CXX object tests/CMakeFiles/chain_test.dir/wasm_tests/currency_tests.cpp.o
[ 99%] Building CXX object tests/CMakeFiles/chain_test.dir/wasm_tests/dice_tests.cpp.o
[ 99%] Building CXX object tests/CMakeFiles/chain_test.dir/wasm_tests/eosio.system_tests.cpp.o
[ 99%] Building CXX object tests/CMakeFiles/chain_test.dir/wasm_tests/eosio.token_tests.cpp.o
[ 99%] Building CXX object tests/CMakeFiles/chain_test.dir/wasm_tests/exchange_tests.cpp.o
[ 99%] Building CXX object tests/CMakeFiles/chain_test.dir/wasm_tests/identity_tests.cpp.o
[ 99%] Building CXX object tests/CMakeFiles/chain_test.dir/wasm_tests/multi_index_tests.cpp.o
[ 99%] Building CXX object tests/CMakeFiles/chain_test.dir/wasm_tests/multisig_tests.cpp.o
[100%] Building CXX object tests/CMakeFiles/chain_test.dir/wasm_tests/payloadless_tests.cpp.o
[100%] Building CXX object tests/CMakeFiles/chain_test.dir/wasm_tests/tic_tac_toe_tests.cpp.o
[100%] Building CXX object tests/CMakeFiles/chain_test.dir/wasm_tests/wasm_tests.cpp.o
[100%] Building CXX object tests/CMakeFiles/chain_test.dir/tests/message_buffer_tests.cpp.o
[100%] Building CXX object tests/CMakeFiles/chain_test.dir/tests/special_accounts_tests.cpp.o
[100%] Building CXX object tests/CMakeFiles/chain_test.dir/tests/wallet_tests.cpp.o
[100%] Building CXX object tests/CMakeFiles/chain_test.dir/library_tests/chain/resource_limits_test.cpp.o
[100%] Building CXX object tests/CMakeFiles/chain_test.dir/common/main.cpp.o
[100%] Linking CXX executable chain_test
[100%] Built target chain_test

        EOSIO

        EOS.IO has been successfully built. 0:7:28

        To verify your installation run the following commands:

        /usr/local/bin/mongod -f /usr/local/etc/mongod.conf &
        export PATH=${HOME}/opt/mongodb/bin:$PATH
        cd /Users/guojh/eos/build; make test

        For more information:
        EOS.IO website: https://eos.io
        EOS.IO Telegram channel @ https://t.me/EOSProject
        EOS.IO resources: https://eos.io/resources/
        EOS.IO wiki: https://github.com/EOSIO/eos/wiki
```

开发环境主要目录说明

- ./contracts：智能合约目录
- ./contracts/eosio.bios：基本输入输出合约
- ./contracts/eosio.token：EOS的发币合约
- ./contracts/dice：一个eos官方提供的骰子游戏
- ./contracts/ping：一个示例合约
- ./contracts/tic_tac_toe：eos官网提供的一个三连棋游戏的示例合约
- ./programs：这里面主要是提供一些编译好的可执行程序

包含前后端的DAPP的主要开发流程

- 1、编写前端源码
- 2、编写后端的智能合约源码，编译并部署智能合约
- 3、使用命令行对智能合约的功能进行测试
- 4、前端通过eosjs来与合约进行交互

说明

- 团队开发，前端界面和智能合约会分别开发
- 我们希望你能成为一个DAPP全站工程师
- 自己开发的话，先开发界面，再写智能合约，可以避免走弯路

本课总结

- 主要介绍了开发环境搭建，以及DAPP的开发流程
- 下节课开始编写前端代码

二、创建前端源码框架，编写前端源码

本节课内容说明：

- 前端简介
- 创建源码目录及相关文件
- 详细介绍一下前端界面的源代码

前端简介

- EOS使用react.js来编写前端的用户界面
- 使用eosjs来与智能合约交互
- eosjs实现了大多数的rpc功能，常用的是合约接口调用和表数据查询功能

创建源码目录及相关文件

- ./www_luckpoint/dist　用于存放入口的index.html文件
- ./www_luckpoint/dist/index.html　前端的入口文件
- ./www_luckpoint/dist/images 存放前端需要的图片等资源文件
- ./www_luckpoint/src 存放react源文件的目录
- ./www_luckpoint/src/index.jsx 核心文件，用来编写真正的用户界面
- ./www_luckpoint/webpack.config.js 打包配置文件
- ./www_luckpoint/package.json 安装依赖包的配置文件
- ./www_luckpoint/.babelrc 配置react引入的组件

详细介绍前端界面源代码

- ./www_luckpoint/src/index.jsx

三、创建智能合约目录，编写智能合约源码

本节课内容说明：

- 智能合约简介
- 使用eosiocpp来创建一个mytest的临时测试合约
- 讲解智能合约源码
- 编译智能合约，生成abi文件

智能合约简介

- 使用C++进行开发，要求必须是依赖boost 1.66.0版本的开源库

使用eosiocpp 来创建一个mytest合约

- cd contracts
- eosiocpp -n mytest
- mytest.hpp 智能合约的头文件
- mytest.cpp 智能合约的源文件
- mytest.abi 智能合约的ABI文件

讲解智能合约源码

- ./contracts/luckpoint/luckpoint.*

编译智能合约，生成abi文件

- eosiocpp -o luckpoint.wast luckpoint.cpp
- eosiocpp -g luckpoint.abi luckpoint.cpp

四、前端调用eosjs，与智能合约交互

本节课内容说明：

- eosjs简介
- 讲解一下在index.jsx中，调用eosjs的流程
- 为前端界面实现eosjs的调用，与合约进行交互

eosjs简介

- eosjs实现了大多数的rpc功能
- 主要使用eosjs来进行智能合约的接口调用，以及查询智能合约的表数据

在index.jsx中，调用eosjs的流程

- 导入eosjs组件
- 实现eosjs配置信息
- 使用指定网络来加载配置文件，实例化eosjs对象
- 载入指定的智能合约
- 调用智能合约接口，或查询智能合约表数据

为前端界面实现eosjs的调用，与合约进行交互

- 为前端文件index.jsx补充进eosjs的代码调用，并跑通整个流程

五、详解使用eosjs，来查询智能合约的table信息

本节课内容说明：

- 通过与cleos命令配合，来详解eosjs查询table信息的操作

六、详解调用eosjs，来调用智能合约接口

本节课内容说明：

- 通过与cleos命令配合，来详解eosjs调用智能合约接口的操作

课程总结

- 源码地址：https://github.com/sailorgege/eos_luckpoint
- E-Mail: strongsailor2005@163.com

谢谢大家！