AELF BOILERPLATE INTRODUCTION

aelf Boilerplate is a development template based on aelf blockchain system with built-in development scaffoldings and simple demo. You can use aelf Boilerplate to quickly build your own blockchain system and develop smart contracts and DAPPs based on aelf blockchain system.

This tutorial will guide you through a demo, and you will able to build aelf blockchain environment in just 5 minutes.

Demo Introduction

This tutorial includes 4 demos:
1.HelloWorld Smart Contract Demo: Guides you to develop smart contracts based on aelf blockchain system using aelf Boilerplate

2.JS SDK Demo: Guide you to build a DAPP that interacts with the above *HelloWorld Smart Contract* by using *JS SDK*

3.aelf Browser Plugin Demo: Guide you to develop DAPP based on aelf blockchain system by using aelf browser plugin

4.DAPP Demo: Guide you to run DAPP based on aelf blockchain system - Bingo Game

Hint:
This tutorial only shows you how to run demos on MacOS system. Procedure for other operating systems are slightly different.
Check aelf official Gitbook for details: https://docs.aelf.io/main

Operating Tutorial

1. Preparation

(1) Development Environment

Visual Studio Code (vscode), need to install *c#* extension

(2) Software Tool

Google Chrome, git, Dotnet core SDK, nodejs

MAC users also need to install *Homebrew*

Related links can be found in the official Gitbook: https://docs.aelf.io/main

2. Run the Demo

We default you have successfully completed the above preparations.

(1) Download and Open *AElf.Boilerplate*

Open terminal and download AElf.Boilerplate by entering the following command

```
git clone https://github.com/aelfProject/aelf-boilerplate
```

When the download is complete, open *AElf.Boilerplate* file in *Visual Studio Code*.

(2) Download *protobuf* script

Open *Visual Studio Code* terminal and enter command *sh Chain/scripts/install.sh* to download *protobuf* script.

(3) Run *HelloWorld* Smart Contract Demo

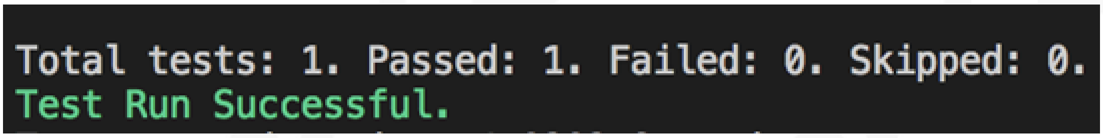
Open *Visual Studio Code* terminal and start the node with the following command:

```
Cd chain/src/AElf.Boilerplate.Launcher/
Dotnet build
Dotnet run bin/Debug/netcoreapp2.2/AElf.Boilerplate.Launcher
```

Test the contract by using the following command:

```
Cd ../../test/HelloWorldContract.Test/
Dotnet test
```

Result:



```
Total tests: 1. Passed: 1. Failed: 0. Skipped: 0.
Test Run Successful.
```

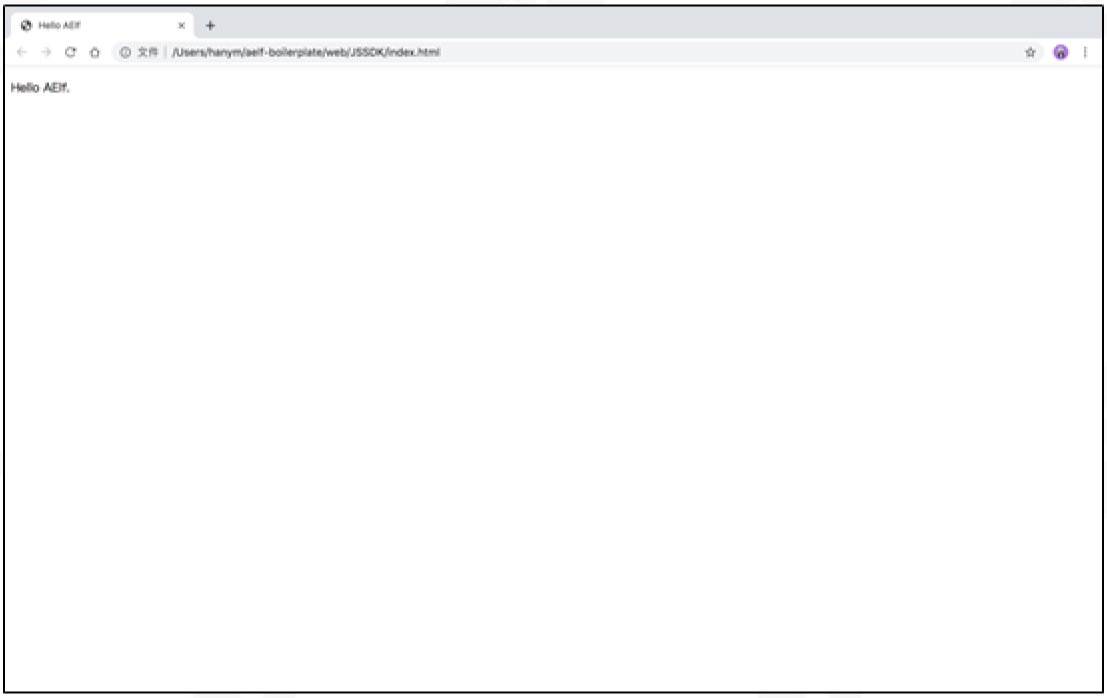
(4) Run *JS SDK* Demo

Open Visual Studio Code terminal to start the node

Run JS SDK Demo with the following command:

```
Cd ../../../web/JSSDK/
Npm install
Npm start
```

Result:



Hello AElf.

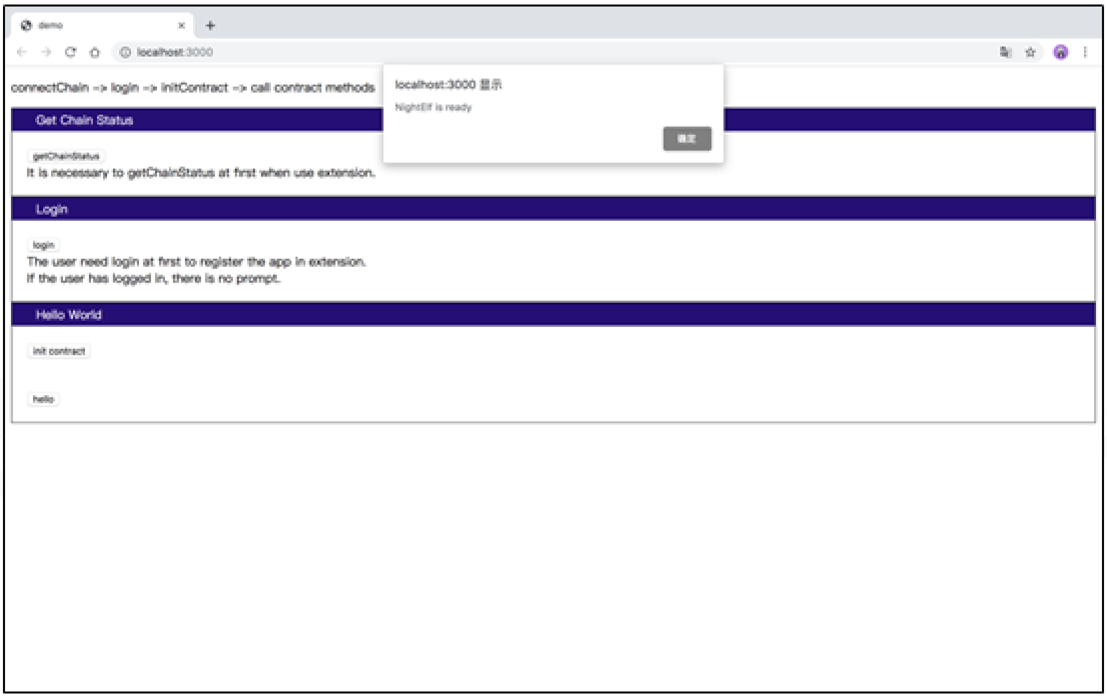
(5) Run aelf browser plugin Demo

Open *Visual Studio Code* terminal to start the node

Run aelf browser plugin with the following command:

```
Cd ../browserExtension/
Npm install
Npm start
```

Result:



connectChain -> login -> initContract -> call contract methods

localhost:3000 显示

NightElf is ready

It is necessary to getChainStatus at first when use extension.

The user need login at first to register the app in extension.
If the user has logged in, there is no prompt.

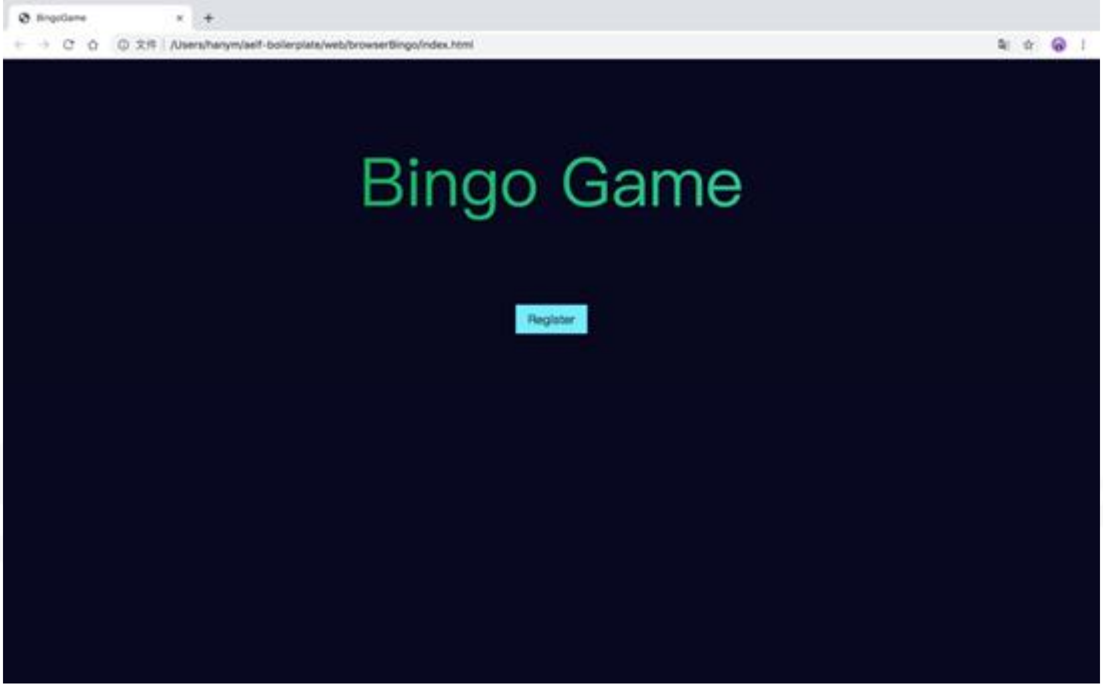
(6) Run DAPP Demo - BingoGame

Open Visual Studio Code terminal to start the node

Run BingoGame with the following command:

```
Cd web/browserBingo/
Open index.html
```

Result:



BingoGame
文件
/Users/hanym/aelf-boilerplate/web/browserBingo/index.html
Bingo Game
Register