

# 导读

本电子书由汇智网 ( <http://www.hubwiz.com> ) 创作, 适用于乌班图 ( Ubuntu ) 平台下以太坊 DApp 开发环境的搭建。

汇智网推出了在线交互式以太坊 DApp 实战开发课程, 以去中心化投票应用 ( Voting DApp ) 为课程项目, 通过三次迭代开发过程的详细讲解与在线实践, 并且将区块链的理念与去中心化思想贯穿于课程实践过程中, 为希望快速入门区块链开发的开发者提供了一个高效的学习与价值提升途径。读者可以通过以下链接访问《以太坊 DApp 开发实战入门》在线教程:

<http://xc.hubwiz.com/course/5a952991adb3847553d205d1?affid=ubuntu7878>

教程预置了开发环境。进入教程后, 可以在每一个知识点立刻进行同步实践, 而不必在开发环境的搭建上浪费时间:



# 一、安装前的准备

## 1.1 查看当前 CPU 架构

在终端中执行以下命令，确定是 32 位架构还是 64 位架构：

```
~$ uname -p  
x86_64
```

如果你看到输出 x86\_64，那么就是 64 位系统，否则是 32 位。

## 1.2 下载工具

确保你安装了下载工具 wget：

```
~$ wget -V  
GNU Wget 1.17.1 built on linux-gnu
```

如果还没有安装 wget，使用 apt-get 来安装

```
~$ sudo apt-get install wget
```

## 二、安装 DApp 开发环境

### 2.1 安装 Node.js

首先根据你的 ubuntu 是 32 位还是 64 位，分别下载不同的预编译版本，我们使用官方长期支持的 8.10.0 LTS 版本：

64 位：

```
~$ wget https://nodejs.org/dist/v8.10.0/node-v8.10.0-linux-x64.tar.gz
```

32 位：

```
~$ wget https://nodejs.org/dist/v8.10.0/node-v8.10.0-linux-x86.tar.gz
```

然后解压到当前目录，以 64 位为例：

```
~$ tar zxvf node-v8.10.0-linux-x64.tar.gz
```

然后接下来修改 .bashrc 来设置相关的环境变量：

```
~$ echo "export NODE_HOME=$HOME/node-v8.10.0-linux-x64" >> .bashrc
~$ echo "export NODE_PATH=$NODE_HOME/lib/node_modules" >> .bashrc
~$ echo "export PATH=$NODE_HOME/bin:$PATH" >> .bashrc
```

最后重新载入 .bashrc（或者重新登陆）来使 node 生效：

```
~$ source .bashrc
```

现在，你可以使用 node 了：

```
~$ node -v
v8.10.0
```

### 2.2 安装节点仿真器

在终端执行以下命令：

```
~$ npm install -g ganache-cli
```

安装完毕后，执行命令验证安装成功：

```
~$ ganache-cli
Ganache CLI v6.0.3 (ganache-core: 2.0.2)
```

## 2.3 安装 solidity 编译器

```
~$ npm install -g solc
```

安装完毕后，执行命令验证安装成功

```
~$ solcjs -version
0.40.2+commit.3155dd80.Emscripten.clang
```

## 2.4 安装 web3

```
~$ npm install -g web3@0.20.2
```

安装验证：

```
~$ node -p 'require("web3")'
{[Function: Web3]
  providers:{...}}
```

## 2.5 安装 truffle 框架

执行以下命令安装 truffle 开发框架：

```
~$ npm install -g truffle
```

验证安装：

```
~$ truffle version
Truffle v4.1.3 (core 4.1.3)
```

## 2.6 安装 webpack

执行以下命令安装 webpack：

```
~$ npm install -g webpack@3.11.0
```

验证安装

```
~$ webpack -v
3.11.0
```

## 三、构建示例项目

### 3.1 新建 DApp 项目

执行以下命令创建项目目录并进入该目录：

```
~$ mkdir demo
~$ cd demo
```

然后用 webpack 模版初始化项目骨架结构：

```
~/demo$ truffle unbox webpack
Downloading...
Unpacking...
Setting up...
Unbox successful. Sweet!
```

### 3.2 安装项目依赖的 NPM 包

执行以下命令安装 npm 包：

```
~/demo$ npm install
```

### 3.3 修改 truffle 配置

truffle.js 中，修改 port 为 8545，因为 ganache-cli 在 8545 端口监听：

```
module.exports = {
  networks: {
    development: {
      ...
      port: 8545
      ...
    }
  }
}
```

## 3.4 启动节点

执行以下命令启动节点仿真器，以便部署合约并执行交易：

```
~/demo$ ganache-cli
```

## 3.5 编译合约

执行以下命令编译项目合约：

```
~/demo$ truffle compile
```

## 3.6 部署合约：

执行以下命令来部署合约：

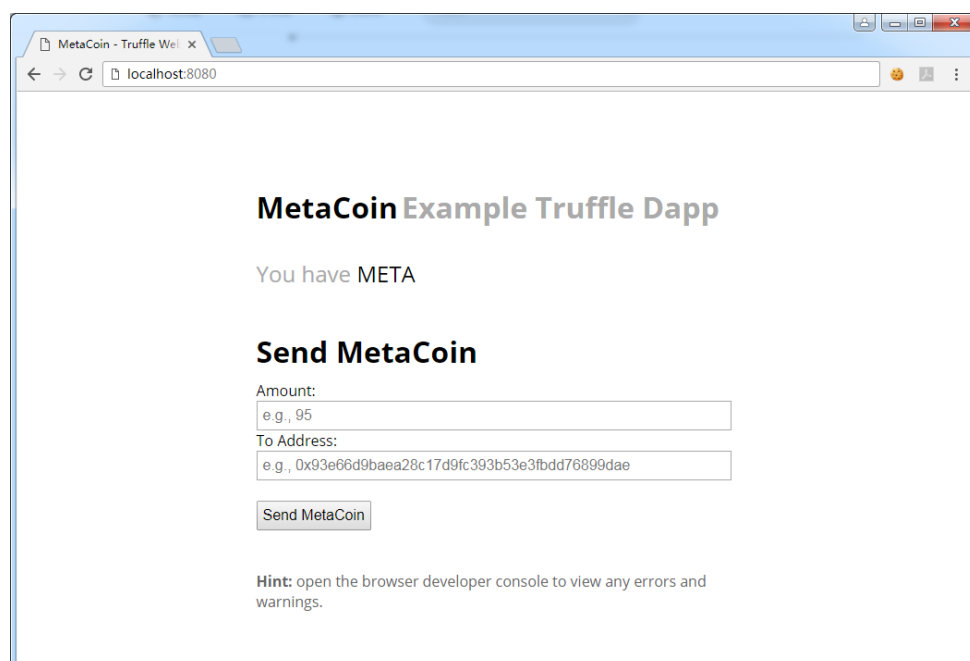
```
~/demo$ truffle migrate
```

## 3.7 启动 DApp

执行以下命令来启动 DApp：

```
~/demo$ npm run dev
```

在浏览器里访问 <http://localhost:8080> 即可



如果你希望从别的机器也可以访问你的 DApp 应用，修改一下 package.json：

```
{
  scripts:{
    "dev": "webpack-dev-server --host 0.0.0.0"
  }
}
```