

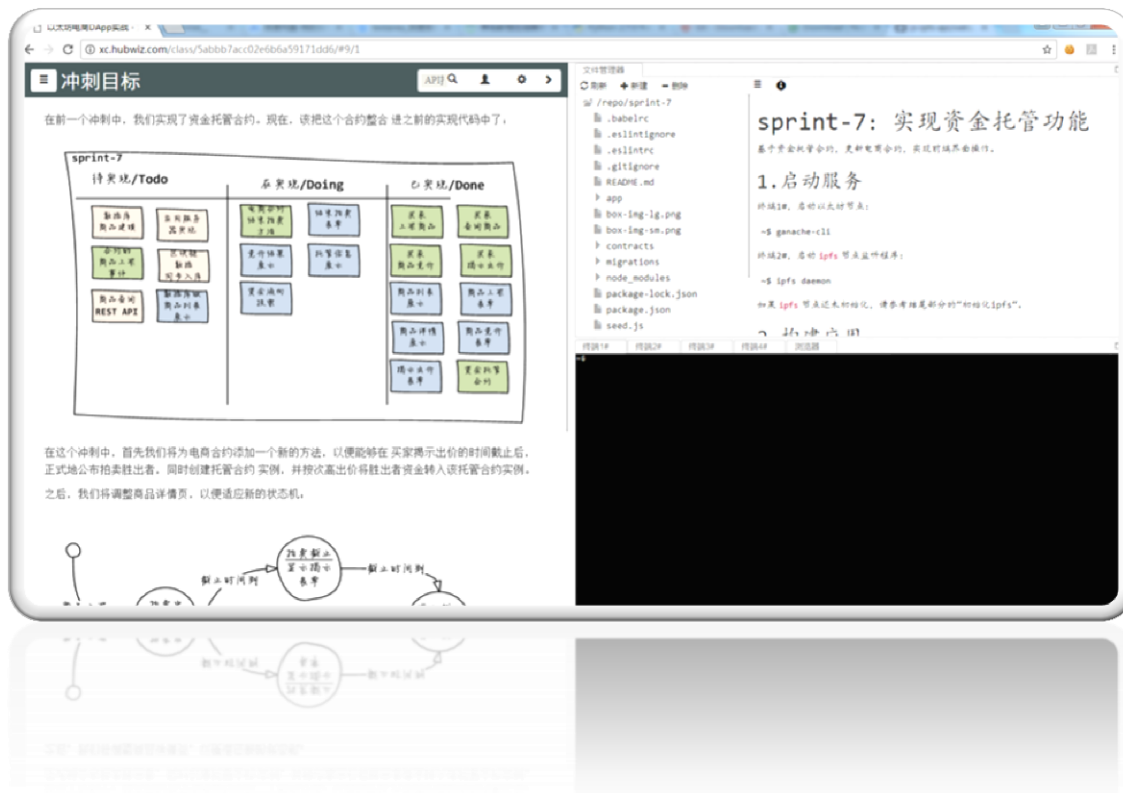
导读

本电子书由汇智网 (<http://www.hubwiz.com>) 创作, 介绍如何 Truffle 中使用 Infura 部署以太坊智能合约。

汇智网推出了在线交互式以太坊电商 DApp 实战开发课程, 以去中心化电商应用为课程项目, 内容涵盖以太坊合约开发、ipfs 去中心化文件系统、链下 mongodb 存储等诸多知识点, 同时采用敏捷开发模式, 通过 8 个冲刺周期的详细讲解与在线实践, 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《以太坊电商 DApp 实战》在线教程:

<http://xc.hubwiz.com/course/5abbb7acc02e6b6a59171dd6>

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:



使用 Infura 部署以太坊合约

Infura 是一个托管的以太坊节点集群，可以将你开发的以太坊智能合约部署到 infura 提供的节点上，而无需搭建自己的以太坊节点。

可能你还不了解 Infura，但如果你使用过 MetaMask，那么就已经接触过 Infura 了，因为它是 MetaMask 背后的以太坊供应商。

出于安全原因，Infura 不管理你的私钥，这意味着 Infura 不能代表你签署交易。

但是，Infura 可以通过使用 `HDWalletProvider` 来签署交易。该服务可以处理事务签名以及与以太坊网络的连接。点击[这里](#)了解更多关于 HDWalletProvider 的信息。

本教程将向你展示如何使用 Infura 将现有的 dapp 迁移到 Infura 支持的以太坊网络。在这个特定的例子中，我们将迁移到 Ropsten 测试网络。我们假设你已经有了一个 dapp，如果需要一个测试 dapp，可以使用 Truffle 提供的[宠物店](#) dapp。

1.1 安装 HDWalletProvider

Infura 的 HDWalletProvider 是一个独立的 npm 软件包，如下安装：

```
npm install truffle-hdwallet-provider
```

注意：如果你在 Windows 上安装并且遇到 MSBUILD 错误，则可能需要安装 Windows 构建工具。在具有管理员权限的控制台中，运行 `npm install -g windows-build-tools` 然后再次尝试安装。

1.2 注册 Infura

在使用 Infura 之前，需要[注册 Infura 访问令牌](#)。

填写并提交表格后你就可以收到访问令牌。相关信息将显示在屏幕上并发送到你提供的电子邮件。需要记录下来这个访问令牌并确保它不被别人看到！

1.3 配置 Truffle 项目

下一步是编辑你的 `truffle.js` 文件来启用 `HDWalletProvider` 并为部署到 Ropsten 进行必要的配置。

STEP 1: 首先，在配置文件中定义 `HDWalletProvider` 对象。在 `truffle.js` 文件的顶部添加以下代码：

```
var HDWalletProvider = require("truffle-hdwallet-provider");
```

STEP 2: 接下来，提供助记词（`mnemonic`）来生成你的账户。

```
var mnemonic = "orange apple banana ... ";
```

警告： 在此过程中，我们强烈建议将助记符存储在另一个（秘密）文件中，以降低助记符泄漏风险。如果有人知道你的助记符，他们将拥有你所有的地址和私钥！

STEP 3: 添加 Ropsten 网络定义：

```
module.exports = {
  networks: {
    ropsten: {
      provider: function() {
        return new HDWalletProvider(mnemonic,
          "https://ropsten.infura.io/<INFURA_Access_Token>")
      },
      network_id: 3
    }
  }
};
```

注意事项：

- 虽然该示例仅定义了单个网络，但你可以像往常一样定义多个网络。
- ropsten 网络定义中的 `provider` 将使用实例化的 `HDWalletProvider`。
- `HDWalletProvider` 以助记符和期望的网络为参数。[Infura 主页](#)上提供 Infura 支持的 网络列表。
- 确保使用前面拿到的 Infura 访问令牌替换。

- `provider` 值被封装在一个函数中，这可以确保它在需要之前不会被初始化。如果连接到多个网络，这一点尤为重要。关于该主题的更多信息，请参阅 Truffle 文档的[网络配置](#)部分。
- 默认情况下，由助记符产生的第一个账户将负责执行合约迁移任务。但如果需要的话，你可以传入参数以指定要使用的帐户。例如，要使用第三个帐户：

```
new HDWalletProvider(mnemonic, "https://ropsten.infura.io/<Infura_Access_Token>", 2);
```

账户索引是从零开始的，所以 2 表示第三个地址。

1.4 使用 Faucet 获取 ether

确保你的帐户有足够的账户余额来进行部署。可以通过称 **Faucet** 的服务在 Ropsten 网络上获取 **Ether**。虽然在那里有多个 **Faucet** 网站，我们推荐的一个服务是在 [EthTools](#) 上托管的。

- 导航至 EthTools 的 [Ether Faucet](#)。
- 输入你的助记符，并选择你想要多少 **ether**（最多 5 个）。
- **Faucet** 将链接到你的第一个帐户。点击“Request Ether”提交请求。
- 很快，你的账户将获得请求的 **ether**。

注意：也可以通过 MetaMask 申请 ether。在 Ropsten 上连接你的帐户，然后点击“Buy”按钮，该按钮将提供 MetaMask 的 Ropsten 测试 Faucet 的链接，它的工作方式与上述类似。

我们现在可以开始将合约部署到 Ropsten 上了！

1.5 部署合约

STEP 1: 编译项目：

```
truffle compile
```

STEP 2: 部署到 Ropsten 网络：

```
truffle migrate --network ropsten
```

如果一切顺利，应该可以看到类似于以下内容的输出：

```
Using network 'ropsten'.
Running migration: 1_initial_migration.js
  Deploying Migrations...
  ... 0xd79bc3c5a7d338a7f85db9f86febbbee738ebdec9494f49bda8f9f4c90b649db7
  Migrations: 0x0c6c4fc8831755595eda4b5724a61ff989e2f8b9
Saving successful migration to network...
  ... 0xc37320561d0004dc149ea42d839375c3fc53752bae5776e4e7543ad16c1b06f0
Saving artifacts...
Running migration: 2_deploy_contracts.js
  Deploying MyContract...
  ... 0x7efbb3e4f028aa8834d0078293e0db7ff8aff88e72f33960fc806a618a6ce4d3
  MyContract: 0xda05d7bfa5b6af7feab7bd156e812b4e564ef2b1
Saving successful migration to network...
  ... 0x6257dd237eb8b120c8038b066e257baee03b9c447c3ba43f843d1856de1fe132
Saving artifacts...
```

需要提醒的是，你的交易 ID 将与上面的不同。

注意：如果收到错误 **Error: Exceeds block gas limit**，你可能需要为合约手动设置油量上限（**gas limit**）。有关详细信息，请参阅 [Truffle 配置文档](#)。

STEP 3: 如果想验证合约是否已成功部署，可以在 Etherscan 的 Ropsten 部分进行检查。在搜索字段中，输入部署交易 ID。在上面的例子中，交易 ID 是：

```
0x7efbb3e4f028aa8834d0078293e0db7ff8aff88e72f33960fc806a618a6ce4d3
```

你应该可以看到有关交易的详细信息，包括交易受到保护的区块号。

恭喜！你已经使用 Infura 和 truffle 的组合将合约部署到 Ropsten 上了！