Учреждение образования
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Кафедра информатики

Отчет по предмету:
«Технология блокчейн»
По лабораторной работе №2

«Write your first blockchain application»

Выполнил: Зюсько Кирилл Дмитриевич
магистрант кафедры информатики
группы №858642

Проверил: Прудник Александр Михайлович
доцент, кандидат технических наук

Минск 2020

Оглавление

1 Цель работы 2
2 Ход работы 3
2.1 Выбор и подготовка среды 3
2.2 Сборка и установка тестового приложения 3
2.3 Проверка и доведение приложения до работоспособности 3
2.4 Изучение смарт контракта приложения и небольшие изменения приложения 4
Вывод 6

1 Цель работы

Получить навыки работы с Hyperledger Fabric, воспроизведя и разобравшись с примером тестового приложения fabcar.

2 Ход работы

2.1 Выбор и подготовка среды

Установим требуемые компоненты Hyperledger Fabric:

```
curl -sSL https://bit.ly/2ysbOFE | bash -s
```

Данный скрипт устанавливает компоненты фреймворка последней версии (на момент написания лабораторной работы это 2.0.0).

Запускаем следующие команды:

```
cd fabric-samples/fabcar
./startFabric.sh javascript
```

2.2 Сборка и установка тестового приложения

Соберем тестовое приложение fabcar:

```
cd javascript
npm install
```

2.3 Проверка и доведение приложения до работоспособности

Создадим аккаунт администратора:

```
node enrollAdmin.js
```

Создадим аккаунт пользователя:

```
node registerUser.js
```

Я получил ошибку:

```
Wallet path: /Users/kiryl.ziusko/fabric-samples/fabcar/javascript/wallet
Failed to register user "user1": TypeError: gateway.getClient is not a function
```

Я нашёл в issue на гите. После патчинга всё заработало (похожие проблемы также наблюдались для query.js и invoke.js):

```
Wallet path: /Users/kiryl.ziusko/fabric-samples/fabcar/javascript/wallet
Successfully registered and enrolled admin user "user1" and imported it into the wallet
```

Выполним скрипт для получения информации о всех автомобилях:

```
node query.js
```

Результат выполнения:

```
Wallet path: /Users/kiryl.ziusko/fabric-samples/fabcar/javascript/wallet
Transaction has been evaluated, result is:
[{"Key":"CAR0","Record":{"color":"blue","docType":"car","make":"Toyota","model":"Prius","owner
":"Tomoko"}},{"Key":"CAR1","Record":{"color":"red","
docType":"car","make":"Ford","model":"Mustang","owner":"Brad"}},{"Key":"CAR2","Record":{"color
":"green","docType":"car","make":"Hyundai","model":"Tucson","owner":"Jin
Soo"}},{"Key":"CAR3","
Record":{"color":"yellow","docType":"car","make":"Volkswagen","model":"Passat","owner":"Max"}}
,{"Key":"CAR4","Record":{"color":"black","docType":"car","make":"Tesla","model":"S","owner":"A
d
riana"}},{"Key":"CAR5","Record":{"color":"purple","docType":"car","make":"Peugeot","model":"20
5","owner":"Michel"}},{"Key":"CAR6","Record":{"color":"white","docType":"car","make":"Chery","
m
odel":"S22L","owner":"Aarav"}},{"Key":"CAR7","Record":{"color":"violet","docType":"car","make"
:"Fiat","model":"Punto","owner":"Pari"}},{"Key":"CAR8","Record":{"color":"indigo","docType":"c
a
r","make":"Tata","model":"Nano","owner":"Valeria"}},{"Key":"CAR9","Record":{"color":"brown","d
ocType":"car","make":"Holden","model":"Barina","owner":"Shotaro"}}]
```

2.4 Изучение смарт контракта приложения и небольшие изменения приложения

Blockchain оперирует chaincode’ом из смарт контрактов, которые установлены в сети. Приложение использует Hyperledger Fabric, вызывая эти функции.

Посмотрим на информацию о конкретном автомобиле, для этого изменим в query.js вызов функции queryAllCars на queryCar:

```
const result = await contract.evaluateTransaction('queryCar', 'CAR4');
```

Запустив скрипт получим:

```
Wallet path: /Users/kiryl.ziusko/fabric-samples/fabcar/javascript/wallet
Transaction has been evaluated, result is:
{"color":"black","docType":"car","make":"Tesla","model":"S","owner":"Adriana"}
```

Изменение состояния blockchain сети отдельная тяжеловесная create/update операция submitTransaction, которая включает в себя “соглашение” компонентов-участников сети, что транзакция валидна по некоторому выбранному алгоритму консенсуса.

С помощью invoke.js создадим новый автомобиль:

```
await contract.submitTransaction('createCar', 'CAR10', 'Lada', 'Sedan', 'Black', 'Tom');
```

Запустим скрипт:

```
node invoke.js
```

Результат выполнения:

```
Wallet path: /Users/kiryl.ziusko/fabric-samples/fabcar/javascript/wallet
Transaction has been evaluated, result is:
{"color":"Black","docType":"car","make":"Lada","model":"Sedan","owner":"Tom"}
```

При попытке получить данные о новой машине по ключу CAR10 с помощью query.js получим эту же информацию.

Проверим update операцию, а именно changeCarOwner с помощью вызова:

```
await contract.submitTransaction('changeCarOwner', 'CAR10', 'Ola');
```

Результат выполнения:

```
Wallet path: /Users/kiryl.ziusko/fabric-samples/fabcar/javascript/wallet
Transaction has been submitted
```

При получении информации по ключу CAR10 видим, что owner автомобиля изменился:

```
Wallet path: /Users/kiryl.ziusko/fabric-samples/fabcar/javascript/wallet
Transaction has been evaluated, result is:
{"color":"Black","docType":"car","make":"Lada","model":"Sedan","owner":"Ola"}
```

Вывод

Запустил тестовое приложение fabcar, использующее Hyperledger Fabric блокчейн. Ознакомился с основами использования сети приложениями.