

Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Кафедра информатики

Отчет по лабораторной работе #2
«Write your first blockchain application»

Карп Александр Игоревич
магистрант кафедры информатики
группа №858641

Проверил: Прудник Александр Михайлович
доцент, кандидат технических наук

Минск 2020

Содержание

- 1. Цель работы**
- 2. Подготовка среды**
- 3. Сборка и установка тестового приложения**
- 4. Изучение смартконтракта приложения**
- 5. Выводы**

1. Цель работы

Получить опыт работы со смартконтрактами на примере тестового приложения fabcar

2. Подготовка среды

Установим окружения согласно данному гайду:

https://hyperledger-fabric.readthedocs.io/en/latest/write_first_app.html#writing-your-first-application

Установим Hyperledger Fabric Samples:

```
curl -sSL https://bit.ly/2ysb0FE | bash -s
```

Запускаем следующие команды:

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

3. Сборка и установка тестового приложения

Соберем тестовое приложение fabcar:

```
cd javascript  
npm install
```

Проверим ее работоспособность

Создадим аккаунт администратора:

```
node enrollAdmin.js
```

Создадим аккаунт пользователя:

```
node registerUser.js
```

Выполним скрипт для получения информации о всех автомобилях:

```
node query.js
```

Результат выполнения:

```
Wallet path: /home/dmitrysenkovich/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": "Adriana"}}, {"Key": "CAR5", "Record": {"color": "purple", "docType": "car", "make": "Peugeot", "model": "205", "owner": "Michel"}}, {"Key": "CAR6", "Record": {"color": "white", "docType": "car", "make": "Chery", "model": "S22L", "owner": "Aarav"}}, {"Key": "CAR7", "Record": {"color": "violet", "docType": "car", "make": "Fiat", "model": "Punto", "owner": "Pari"}}, {"Key": "CAR8", "Record": {"color": "indigo", "docType": "car", "make": "Tata", "model": "Nano", "owner": "Valeria"}}, {"Key": "CAR9", "Record": {"color": "brown", "docType": "car", "make": "Holden", "model": "Barina", "owner": "Shotaro"}}]
```

4. Изучение смарт контракта 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:

```
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:

```
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:

```
Transaction has been submitted
```

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

```
Transaction has been evaluated, result is:  
{ "color": "Black", "docType": "car", "make": "Lada", "model": "Sedan", "owner": "Ola" }
```

5. Выводы

В результате выполнения лабораторной работы были изучены основы фреймворка HyperLedger, изучен пример fabcar и работа смартконтракта в рамках данного приложения