

Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет Компьютерных сетей и систем

Кафедра Информатики

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2

по курсу «Технология блокчейн»

Writing Your First Application

Студент:
гр. 758641
Ярош Г.И.

Проверил:
Самаль Д. И.

Минск, 2019

ХОД РАБОТЫ

В данной лабораторной работе изучались основы работы с Hyperledger Fabric на примере приложения FabCar.

В начале работы мною был запущен сам Hyperledger Fabric с помощью скрипта из репозитория fabric-dev-servers:

```
> ./startFabric.sh
```

Затем был клонирован репозиторий с примерами приложений fabric-samples. В нем использовалось демонстрационное приложение FabCar. Перед началом работы были установлены зависимости приложения с помощью следующей команды:

```
> npm install
```

После установки зависимостей приложения был создан администратор сети с помощью команды:

```
> node enrollAdmin.js
Wallet path: .../fabric-samples/fabcar/javascript/wallet
Successfully enrolled admin user "admin" and imported it into the wallet
```

Далее был зарегистрирован пользователь user1:

```
> node registerUser.js
Wallet path: .../fabric-samples/fabcar/javascript/wallet
Successfully registered and enrolled admin user "user1" and imported it into the wallet
```

Следующим шагом стало выполнение запроса к Hyperledger Fabric с помощью запуска скрипта query.js. Данный скрипт подключается к сети и запрашивает все созданные машины:

```
> node query.js
Wallet path: .../fabric-samples/fabcar/javascript/wallet
Transaction has been evaluated, result is:
[
  {
    "Key": "CAR0",
    "Record": {
      "colour": "blue",
      "make": "Toyota",
      "model": "Prius",
      "owner": "Tomoko"
    }
  },
  ...
]
```

Далее скрипт query.js был изменен таким образом, чтобы получать запись только об одной машине. Для этого строка с выполнением транзакции queryAllCars была заменена на следующую:

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

Выполнение скрипта query.js теперь дает следующий вывод:

```
> node query.js
Wallet path: ../fabric-samples/fabcar/javascript/wallet
Transaction has been evaluated, result is:
{"colour":"black","make":"Tesla","model":"S","owner":"Adriana"}
```

Скрипт был несколько раз запущен с различными идентификаторами записей машин:

```
> node query.js // CAR1
Wallet path: ../fabric-samples/fabcar/javascript/wallet
Transaction has been evaluated, result is:
{"colour":"red","make":"Ford","model":"Mustang","owner":"Brad"}

> node query.js // CAR5
Wallet path: ../fabric-samples/fabcar/javascript/wallet
Transaction has been evaluated, result is:
{"colour":"purple","make":"Peugeot","model":"205","owner":"Michel"}
```

Далее с помощью скрипта invoke.js было выполнено создание новых записей о машинах:

```
> node invoke.js // default
Wallet path: ../fabric-samples/fabcar/javascript/wallet
2019-04-18T18:37:50.380Z - info: [TransactionEventHandler]:
_strategySuccess: strategy success for transaction
"4403478f1a79efeb44d54eaab5ca279902a567f3352e6587df3ffeb3830b93da"
Transaction has been submitted

> node invoke.js // CAR13, Ford Escort, Blue, Jack
Wallet path: ../fabric-samples/fabcar/javascript/wallet
2019-04-18T18:39:33.089Z - info: [TransactionEventHandler]:
_strategySuccess: strategy success for transaction
"bc1a9cad5ffcf7001bab41da86dafd3b57c0df87e2852dfea2bd539ff5b25a2e"
Transaction has been submitted
```

Созданные машины были запрошены с помощью скрипта query.js:

```
> node query.js // CAR13
Wallet path: ../fabric-samples/fabcar/javascript/wallet
Transaction has been evaluated, result is:
{"colour":"Blue","make":"Ford","model":"Escord","owner":"Jack"}

> node query.js // CAR12
Wallet path: ../fabric-samples/fabcar/javascript/wallet
Transaction has been evaluated, result is:
{"colour":"Black","make":"Honda","model":"Accord","owner":"Tom"}
```

Далее была выполнена транзакция смены владельца машины. Для этого был модифицирован скрипт invoke.js путем добавления следующей строки:

```
await contract.submitTransaction('changeCarOwner', 'CAR13', 'John');
```

Соответственно, для обновления владельца был запущен скрипт `invoke.js`:

```
> node query.js // Update CAR13 owner
Wallet path: ../fabric-samples/fabcar/javascript/wallet
2019-04-18T18:47:45.137Z - info: [TransactionEventHandler]:
_strategySuccess: strategy success for transaction
"5571ca551504e172b4dc60566337c131984d46d19cce7681286ec5d4effd6848"
Transaction has been submitted
```

Для проверки, что действительно поменялся владелец машины, CAR13 была перезапрошена:

```
> node query.js // CAR13
Wallet path: ../fabric-samples/fabcar/javascript/wallet
Transaction has been evaluated, result is:
{"colour":"Blue","make":"Ford","model":"Escord","owner":"John"}
```

Как мы можем убедиться, запись о машине изменилась, владельцем стал John.

ЗАКЛЮЧЕНИЕ

В данной лабораторной работе были изучены основы работы с Hyperledger Fabric, а также с Hyperledger Javascript API .