Welcome !

Fabric Application Developer Community Call

Thurs Apr 16th 2020

Agenda

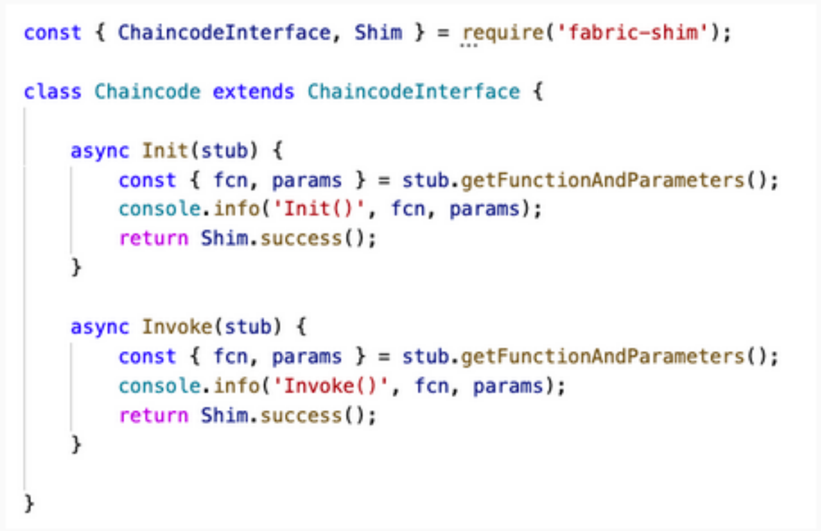
Today's Agenda :

- **Landscape of Application SDKs and Contract APIs**
  - Heather Pollard
- Using the new `test-network`
  - Rob Thatcher
- Fabric V2 uptake
- AOB

New Fabric Programming Model

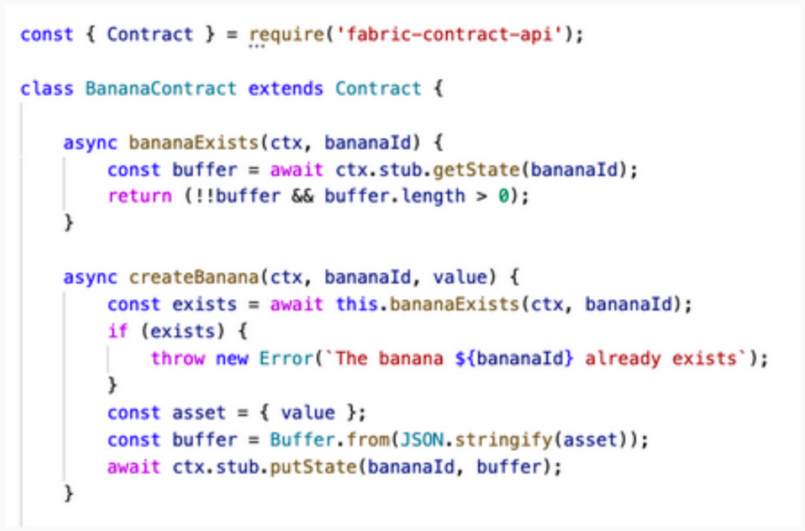
Chaincode

provides **a contract interface**, allowing developers to implement smart contracts



```
const { ChaincodeInterface, Shim } = require('fabric-shim');

class Chaincode extends ChaincodeInterface {

    async Init(stub) {
        const { fcn, params } = stub.getFunctionAndParameters();
        console.info('Init()', fcn, params);
        return Shim.success();
    }

    async Invoke(stub) {
        const { fcn, params } = stub.getFunctionAndParameters();
        console.info('Invoke()', fcn, params);
        return Shim.success();
    }

}
```

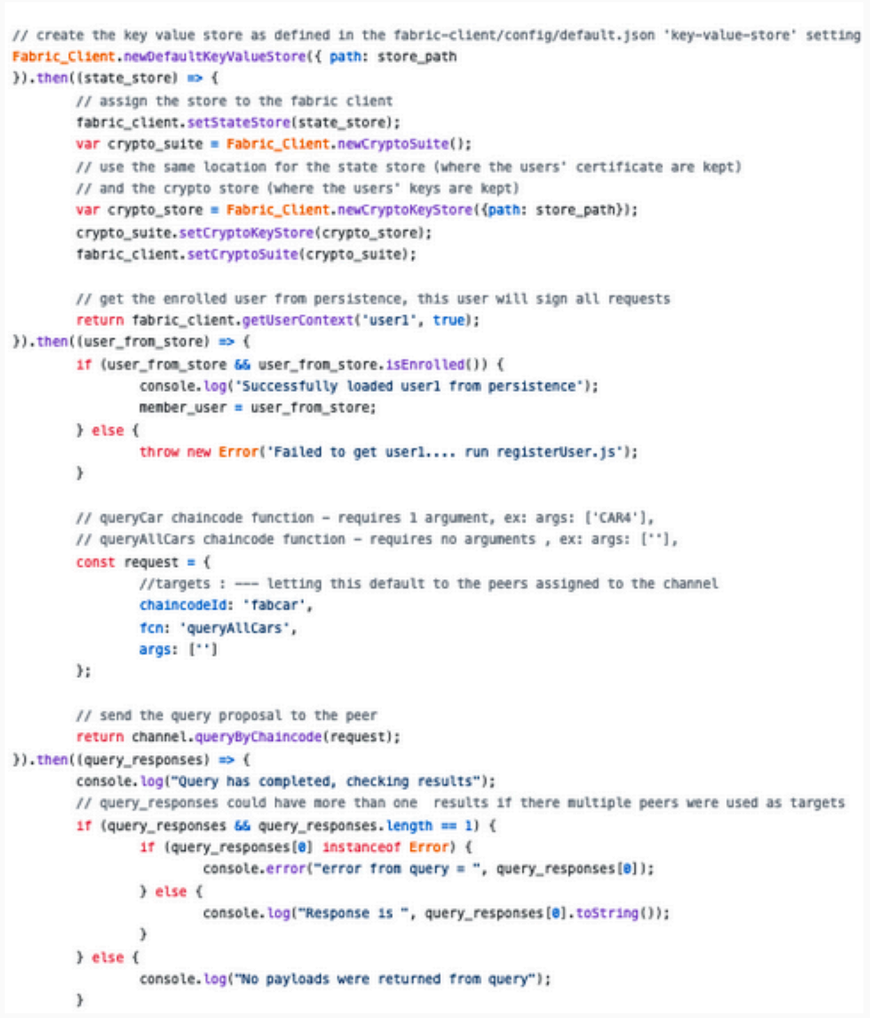
replaced by



```
const { Contract } = require('fabric-contract-api');

class BananaContract extends Contract {

    async bananaExists(ctx, bananaId) {
        const buffer = await ctx.stub.getState(bananaId);
        return (!!buffer && buffer.length > 0);
    }

    async createBanana(ctx, bananaId, value) {
        const exists = await this.bananaExists(ctx, bananaId);
        if (exists) {
            throw new Error(`The banana ${bananaId} already exists`);
        }
        const asset = { value };
        const buffer = Buffer.from(JSON.stringify(asset));
        await ctx.stub.putState(bananaId, buffer);
    }
```

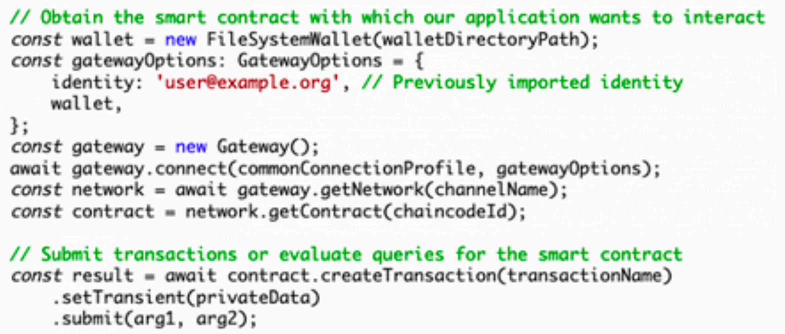
Applications

provides **a gateway class**; a connection to a peer within a blockchain network



```
// create the key value store as defined in the fabric-client/config/default.json 'key-value-store' setting
Fabric_Client.newDefaultKeyValueStore({ path: store_path
}).then((state_store) => {
        // assign the store to the fabric client
        fabric_client.setStateStore(state_store);
        var crypto_suite = Fabric_Client.newCryptoSuite();
        // use the same location for the state store (where the users' certificate are kept)
        // and the crypto store (where the users' keys are kept)
        var crypto_store = Fabric_Client.newCryptoKeyStore({path: store_path});
        crypto_suite.setCryptoKeyStore(crypto_store);
        fabric_client.setCryptoSuite(crypto_suite);

        // get the enrolled user from persistence, this user will sign all requests
        return fabric_client.getUserContext('user1', true);
}).then((user_from_store) => {
        if (user_from_store && user_from_store.isEnrolled()) {
                console.log('Successfully loaded user1 from persistence');
                member_user = user_from_store;
        } else {
                throw new Error('Failed to get user1.... run registerUser.js');
        }

        // queryCar chaincode function - requires 1 argument, ex: args: ['CAR4'],
        // queryAllCars chaincode function - requires no arguments , ex: args: [''],
        const request = {
                //targets : --- letting this default to the peers assigned to the channel
                chaincodeId: 'fabcar',
                fcn: 'queryAllCars',
                args: ['']
        };

        // send the query proposal to the peer
        return channel.queryByChaincode(request);
}).then((query_responses) => {
        console.log("Query has completed, checking results");
        // query_responses could have more than one  results if there multiple peers were used as targets
        if (query_responses && query_responses.length == 1) {
                if (query_responses[0] instanceof Error) {
                        console.error("error from query = ", query_responses[0]);
                } else {
                        console.log("Response is ", query_responses[0].toString());
                }
        } else {
                console.log("No payloads were returned from query");
        }
```

replaced by



```
// Obtain the smart contract with which our application wants to interact
const wallet = new FileSystemWallet(walletDirectoryPath);
const gatewayOptions: GatewayOptions = {
    identity: 'user@example.org', // Previously imported identity
    wallet,
};
const gateway = new Gateway();
await gateway.connect(commonConnectionProfile, gatewayOptions);
const network = await gateway.getNetwork(channelName);
const contract = network.getContract(chaincodeId);

// Submit transactions or evaluate queries for the smart contract
const result = await contract.createTransaction(transactionName)
    .setTransient(privateData)
    .submit(arg1, arg2);
```

Node (JavaScript/TypeScript)

fabric-chaincode-node

- **fabric-contract-api**
- fabric-shim
- fabric-shim-crypto

- *node chaincode docker image (major/minor releases)*

```
npm install --save fabric-contract-api
```

1.4.5

2.1.0

fabric-sdk-node

- **fabric-network**
- fabric-common
- fabric-protos

1.4.8

2.0.0-beta.4

- fabric-client
- fabric-ca-client

1.4.8

Java

fabric-chaincode-java

- **fabric-chaincode-shim**

- *java chaincode docker image (major/minor releases)*

```
dependencies {
    compile group: 'org.hyperledger.fabric-chaincode-java', name: 'fabric-chaincode-shim', version: '1.4.+'
}
```

1.4.6

2.1.0

fabric-gateway-java

1.4.3

2.0.0

```
implementation 'org.hyperledger.fabric:fabric-gateway-java:2.0.0'
```

fabric-sdk-java

1.4.8

2.0.0

Go

fabric-contract-api-go

- **contractapi**

```
go get -u github.com/hyperledger/fabric-contract-api-go
```

1.0.0

fabric-sdk-go

- fabsdk

```
go get github.com/hyperledger/fabric-sdk-go
```

1.0.0-beta1

Agenda

Today's Agenda :

- Landscape of Application SDKs and Contract APIs
  - Heather Pollard
- **Using the new `test-network`**
  - Rob Thatcher
- Fabric V2 uptake
- AOB

test-network

- BYFN (first-network)
- Deprecated
- Too difficult to maintain with new features
- Cryptogen dependency
- End to end scenario

test-network

- Clean slate
- Underpins tutorials
    - Fabcar
    - Commercial paper etc.
- Deploy a Smart Contract (new chaincode lifecycle)

test-network

- Composable with checkpoints
- Artefacts easier to find and understand
- Smaller and Faster
    - Single peer per organisation
    - Single node Raft (not 5 node)
- Cryptogen used by default – but **not** mandated (CA x3 is option)
- Fabcar is the sample contract used

Community Chat

HYPERLEDGER FABRIC

- Developer Chat (RocketChat) – channel names for posting:

  https://chat.hyperledger.org/channel/

  #fabric-chaincode-dev

  #fabric-sdk-node

  #fabric-sdk-java

  #fabric-sdk-go

HYPERLEDGER
FABRIC

Community Support

- **Questions & Open Community Support (Stack, Mailing List, Twitter)**

Contracts: http://stackoverflow.com/questions/tagged/hyperledger-fabric

http://stackoverflow.com/questions/tagged/hyperledger-chaincode

SDKs: http://stackoverflow.com/questions/tagged/hyperledger-fabric-sdk-js

(etc ie *'language suffix'*)

Twitter: https://twitter.com/hyperledger

Mailing List: **details next page**

Application Developer Community Support (Mailing List)

- **Hyperledger Mailing List**

To subscribe or unsubscribe, visit **https://lists.hyperledger.org/g/fabric**

or, via email, send a message with subject or body 'help' to **fabric@lists.hyperledger.org**

Lists (Subgroups): https://lists.hyperledger.org/g/main

Help**:** fabric+help@lists.hyperledger.org

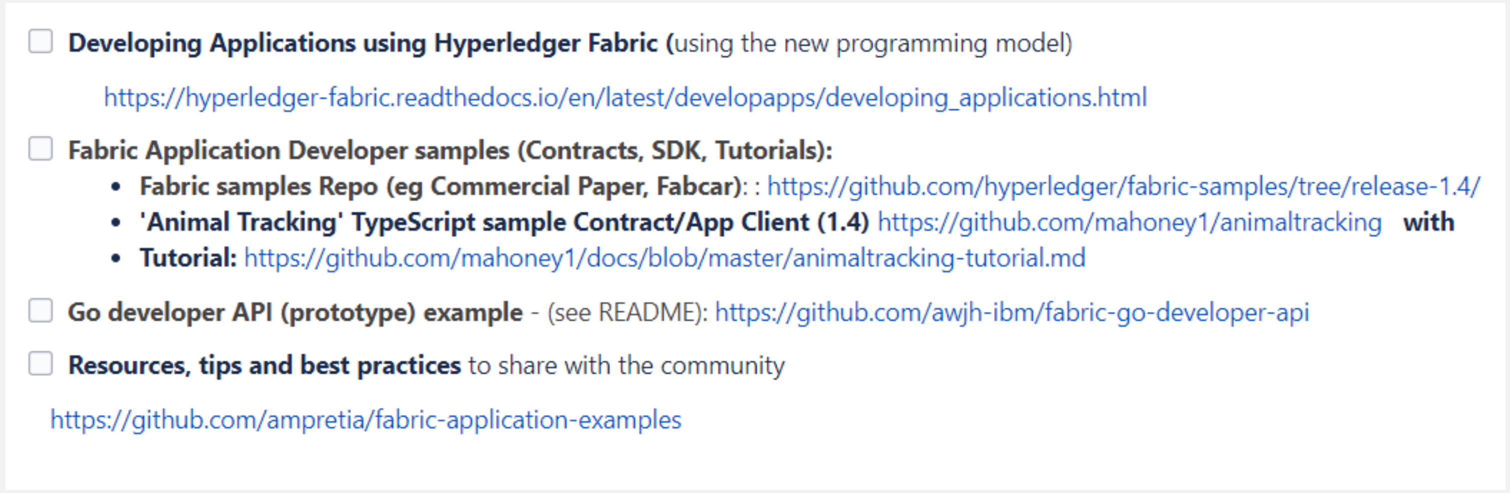
You can change your settings once you log in at

https://lists.hyperledger.org/g/main

Further Links

HYPERLEDGER FABRIC

- **Fabric Developer - Community Home Page**
  https://wiki.hyperledger.org/display/fabric/Fabric+Application+Developer+Community+Calls
- **Fabric Developer Community – resources**
  https://wiki.hyperledger.org/display/fabric/Resources%3A+Fabric+App+Developer+Community



☐ **Developing Applications using Hyperledger Fabric** (using the new programming model)

https://hyperledger-fabric.readthedocs.io/en/latest/developapps/developing_applications.html

☐ **Fabric Application Developer samples (Contracts, SDK, Tutorials):**

- **Fabric samples Repo (eg Commercial Paper, Fabcar)**: : https://github.com/hyperledger/fabric-samples/tree/release-1.4/
- **'Animal Tracking' TypeScript sample Contract/App Client (1.4)** https://github.com/mahoney1/animaltracking **with**
- **Tutorial:** https://github.com/mahoney1/docs/blob/master/animaltracking-tutorial.md

☐ **Go developer API (prototype) example** - (see README): https://github.com/awjh-ibm/fabric-go-developer-api

☐ **Resources, tips and best practices** to share with the community

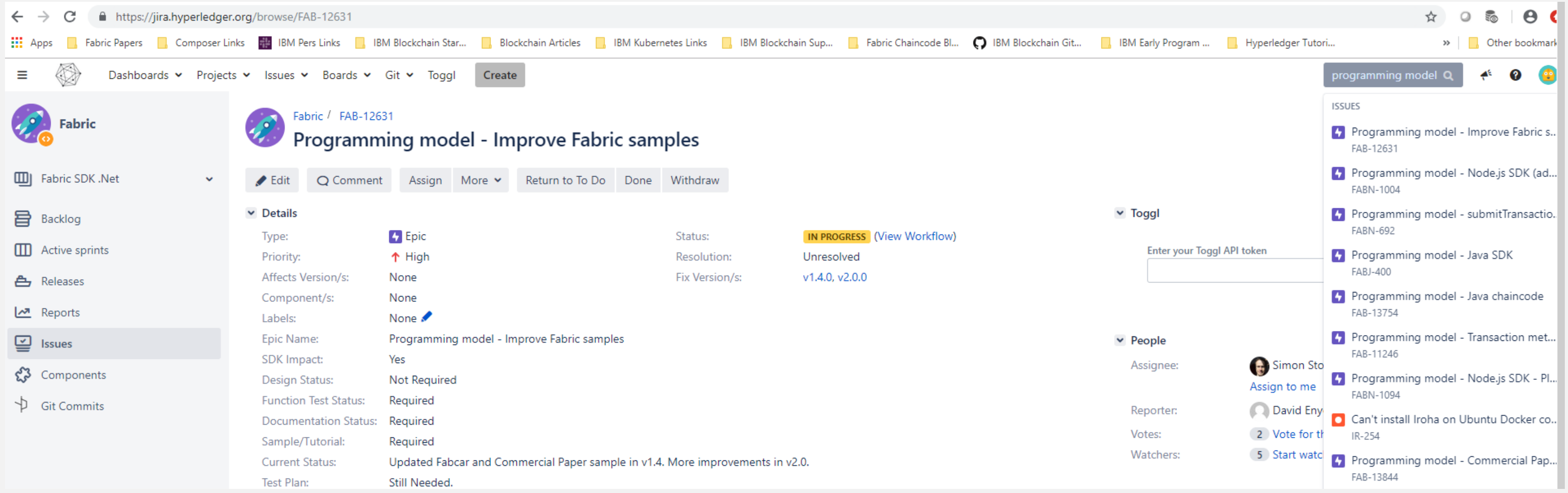
https://github.com/ampretia/fabric-application-examples

Fabric Team encourages you to review/comment on current Fabric JIRAs

(ie stories, epics, requests relating to work-in-progress)

Eg. JIRA search - ‘programming model’ – (results below)

JIRAs – feel free to comment, give your input



https://jira.hyperledger.org/browse/FAB-12631

Fabric / FAB-12631

Programming model - Improve Fabric samples

Type:	Epic
Priority:	High
Affects Version/s:	None
Component/s:	None
Labels:	None
Epic Name:	Programming model - Improve Fabric samples
SDK Impact:	Yes
Design Status:	Not Required
Function Test Status:	Required
Documentation Status:	Required
Sample/Tutorial:	Required
Current Status:	Updated Fabcar and Commercial Paper sample in v1.4. More improvements in v2.0.
Test Plan:	Still Needed.
Status:	IN PROGRESS (View Workflow)
Resolution:	Unresolved
Fix Version/s:	v1.4.0, v2.0.0

People

Assignee: Simon Sto

Reporter: David Eny

Votes: 2

Watchers: 5

Search: programming model

ISSUES

- Programming model - Improve Fabric s... FAB-12631
- Programming model - Node.js SDK (ad... FABN-1004
- Programming model - submitTransactio... FABN-692
- Programming model - Java SDK FABJ-400
- Programming model - Java chaincode FAB-13754
- Programming model - Transaction met... FAB-11246
- Programming model - Node.js SDK - Pl... FABN-1094
- Can't install Iroha on Ubuntu Docker co... IR-254
- Programming model - Commercial Pap... FAB-13844

Reference

HYPERLEDGER FABRIC

New Programming Model (1.4.x) :

- JIRA references - Fabric Programming Model Info:
  - https://jira.hyperledger.org/projects/FABN/issues/FABN-692
  - https://jira.hyperledger.org/browse/FAB-11246

THANKS !

HYPERLEDGER FABRIC

CIAO FOR NOW !