TODO

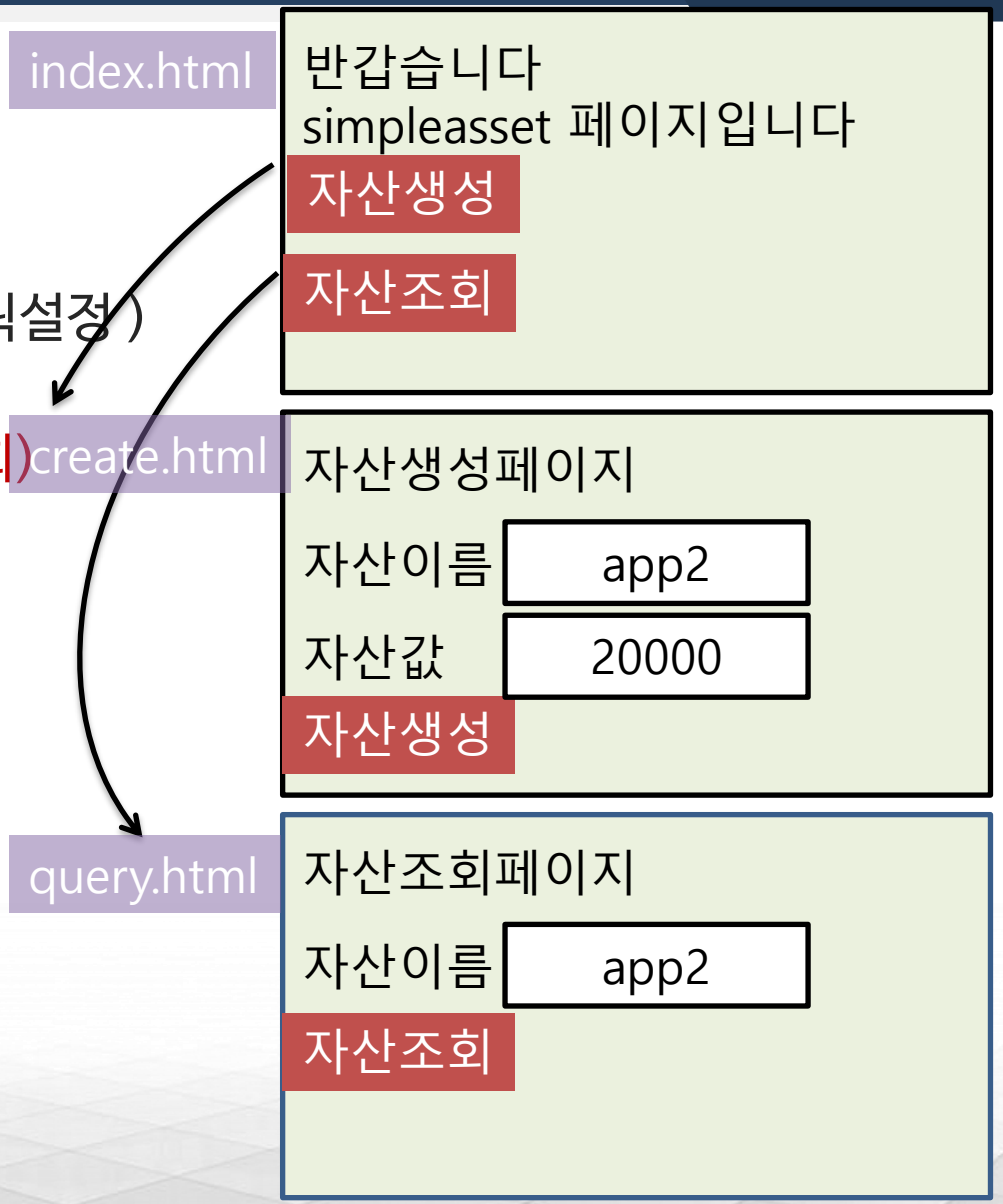
- **웹서비스 서버,**
  - nodejs (페이지, REST 라우팅, 체인코드 연동)
- **클라이언트 구현**
  - html 코딩 + css (bootstrap library), jquery+ajax
- **fabric-연동**
  - 지갑 (enrollAdmin.js, registerUser.js)
  - cc 호출 (submitTransaction, evaluateTransaction)

https://github.com/saarc/teamate/tree/master/application

simpleasset 웹서비스 구현 실습

nodejs application 구현

- 서버구현
  - 외부모듈포함
  - 서버속성설정 (웹서버설정, 패브릭설정)
  - 페이지 라우팅 (i,c,q)
  - REST 라우팅 (자산생성, 자산조회)
  - **체인코드 연결부를 작성**
  - **체인코드 호출결과를 -〉 client**
- 웹클라이언트 구현
  - index.html
  - create.html
  - query.html
- 네트워크에 simpleasset 설치배포
- 서버시작



index.html

반갑습니다
simpleasset 페이지입니다

관리자 지갑생성

사용자 지갑추가

자산생성

자산조회

wallet.html

지갑추가 페이지

지갑이름 user2

역할

지갑생성

history.html

자산이력 페이지

transaction **history**

[{txid:....., value:....., timestamp.....}...]

query.html

자산조회 페이지

자산이름 app2

자산조회

이력조회

잔액 10000

전송

transfer.html

자산전송 페이지

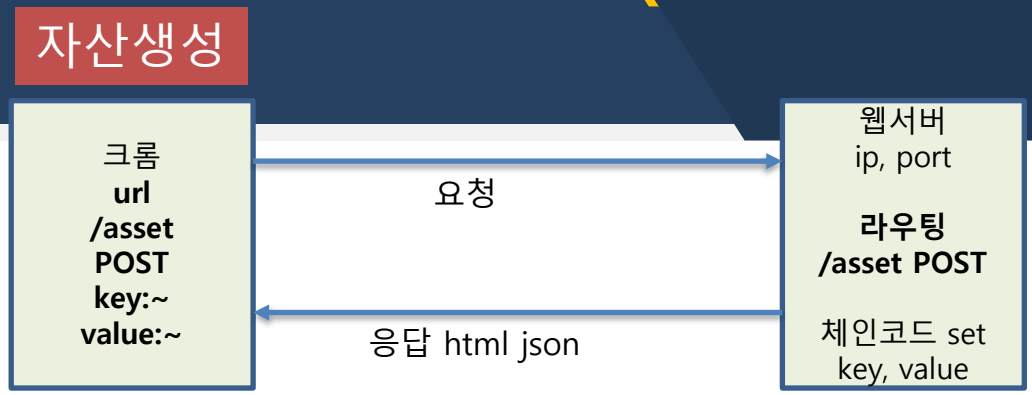
from app1

to app2

amount 200

전송

REST routing



- item
  - url: /asset
- 생성
  - POST （args - key, value）응답: 성공실패
    -〉peer0.org1 mychannel simpleasset submit transaction (“set”, key, value)
    -〉크롬에게 tx제출 결과보내주기
- 조회
  - GET (args - key) 응답: 성공(조회된 value JSON)실패
    -〉peer0.org1 mychannel simpleasset evaluate transaction (“get”, key)
    -〉크롬에게 결과를 클라이언트에게 JSON형태로 보내주기
- 수정
  - PUT (args - key, value） 응답: 성공실패
- 삭제
  - DELETE (args - key) 응답: 성공실패

directory구조 생성

- ~/dev/first-project/application
  - /views/
    - index.html
    - create.html
    - query.html
  - server.js
  - package.json
  - enrollAdmin.js
  - registerUser.js
  - connection.json

~/dev/fabcar/javascript
에서 복사

```
bstudent@bstudentvb:~$ cd dev/fabcar/javascript/
bstudent@bstudentvb:~/dev/fabcar/javascript$ ls
connection.json  enrollAdmin.js  invoke.js  package.json  query.js  registe
rUser.js
bstudent@bstudentvb:~/dev/fabcar/javascript$ tree
.
├── connection.json
├── enrollAdmin.js
├── invoke.js
├── package.json
├── query.js
└── registerUser.js

0 directories, 6 files
bstudent@bstudentvb:~/dev/fabcar/javascript$ cp enrollAdmin.js registerUser
.js connection.json ~/dev/first-project/application/
bstudent@bstudentvb:~/dev/fabcar/javascript$ cd ~/dev/first-project/applica
tion/
bstudent@bstudentvb:~/dev/first-project/application$ ls
connection.json  enrollAdmin.js  registerUser.js
```

nodejs 설정파일 생성

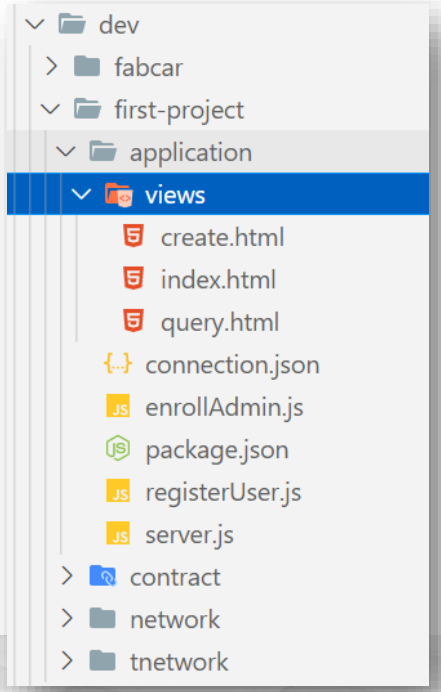
```
bstudent@bstudentvb:~/dev/first-project/application$ npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (application) simpleasset
version: (1.0.0)
description:
entry point: (enrollAdmin.js) server.js
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to /home/bstudent/dev/first-project/application/package.json:

{
  "name": "simpleasset",
  "version": "1.0.0",
  "description": "",
  "main": "server.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}


Is this OK? (yes) yes
```

package.json 자동 생성

```
bstudent@bstudentvb:~/dev/first-project/application$ ls
connection.json  enrollAdmin.js  package.json  registerUser.js
bstudent@bstudentvb:~/dev/first-project/application$ cat package.json
{
  "name": "simpleasset",
  "version": "1.0.0",
  "description": "",
  "main": "server.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}
```

디렉토리 구조 만들기

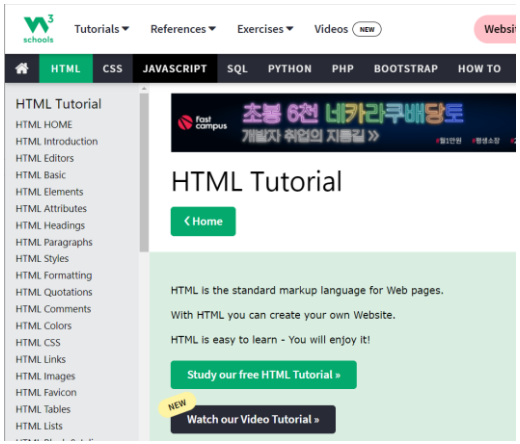
```
bstudent@bstudentvb:~/dev/first-project/application$ mkdir views
bstudent@bstudentvb:~/dev/first-project/application$ ls
connection.json  enrollAdmin.js  package.json  registerUser.js  views
bstudent@bstudentvb:~/dev/first-project/application$ touch server.js
bstudent@bstudentvb:~/dev/first-project/application$ ls
connection.json  enrollAdmin.js  package.json  registerUser.js  server.js  views
bstudent@bstudentvb:~/dev/first-project/application$ cd views/
bstudent@bstudentvb:~/dev/first-project/application/views$ ls
bstudent@bstudentvb:~/dev/first-project/application/views$ touch index.html create.html query.html
bstudent@bstudentvb:~/dev/first-project/application/views$ ls
create.html  index.html  query.html
bstudent@bstudentvb:~/dev/first-project/application/views$ cd ..
bstudent@bstudentvb:~/dev/first-project/application$ tree ./
./
├── connection.json
├── enrollAdmin.js
├── package.json
├── registerUser.js
├── server.js
└── views
    ├── create.html
    ├── index.html
    └── query.html

1 directory, 8 files
```



HTML작성

- index.html

https://www.w3schools.com/html/




```
⟨html⟩ <!-- 문서의 시작과 끝을 의미 -->
        ⟨head⟩ <!-- 해더영역 -->
                주로 외부 라이브러리 링크
        ⟨/head⟩

        ⟨body⟩ <!-- 몸체영역 -->
                페이지의 문서내용 작성 예) 문장, 그림, 사용자입력폼, 버튼등
                ⟨a⟩⟨/a⟩tag : 페이지 이동을 위한 링크를 표현
                ⟨p⟩⟨/p⟩tag: 문장을 표현
                ⟨h1⟩ ~⟨h9⟩ : 해더 문장을 표현
                ⟨br⟩ : 줄바꿈을 표현
        ⟨/body⟩

        ⟨script⟩⟨/script⟩ <!- 동작 프로그래밍 영역 -->
        ⟨style⟩⟨/style⟩   <!- CSS 스타일 영역 -->
⟨/html⟩
```

index 페이지

Run ›

Result Size: 471 x 708

Get your website

```
<!DOCTYPE html>

<html>
    <head>
        <title>SIMPLE ASSET PROTOTYPE</title>
    </head>

    <body>
        <H1> Wecome to SIMPLE ASSET PROTOTYE</H1>
        <p> 어릴적.... 순수한마음으로... </p>

        <br>
        <a href="./create.html">자산생성</a>
        <br>
        <a href="./query.html">자산조회</a>
        <br>

    </body>

    <script>

    </script>

    <style>

    </style>
</html>
```

Wecome to SIMPLE ASSET PROTOTYE

어릴적.... 순수한마음으로...

자산생성
자산조회

HTML작성

- create.html

```
<html> <!-- 문서의 시작과 끝을 의미 -->
        <head> <!-- 해더영역 -->
                주로 외부 라이브러리 링크
        </head>

        <body> <!-- 몸체영역 -->
                페이지의 문서내용 작성 예) 문장, 그림, 사용자입력폼, 버튼등
                <form>
                        <label> 라벨표현
                        <input> 사용자 입력 형식 예) 텍스트입력, 버튼
                </form>
        </body>

        <script></script> <!- 동작 프로그래밍 영역 -->
        <style></style>   <!- CSS 스타일 영역 -->
</html>
```

create 페이지

Run ›

Result Size: 471 x 708

Get your website

```
<!DOCTYPE html>

<html>
    <head>
        <title>SIMPLE ASSET PROTOTYPE</title>
    </head>

    <body>
        <H1> 자산생성 페이지 </H1>
        <p> 자산생성에 필요한 정보를 입력하세요. </p>

        <br>
        <form action="/asset" method="post">
            자산이름:<br><input type="text" name="key">
            <br><br>
            자산값:<br><input type="txt" name="value">
            <br><br>
            <input type="submit" value="자산생성">
        </form>
        <br>

    </body>

    <script>

    </script>

    <style>

    </style>
</html>
```

자산생성 페이지

자산생성에 필요한 정보를 입력하세요.

자산이름:

자산값:

자산생성

query 페이지

Run ›

Result Size: 471 x 708

Get your website

```
<!DOCTYPE html>

<html>
    <head>
        <title>SIMPLE ASSET PROTOTYPE</title>
    </head>

    <body>
        <H1> 자산조회 페이지 </H1>
        <p> 자산조회에 필요한 정보를 입력하세요. </p>

        <br>
        <form action="/asset" method="post">
            자산이름:<br><input type="text" name="key">
            <br><br>
            <input type="submit" value="자산조회">
        </form>
        <br>

    </body>

    <script>

    </script>

    <style>

    </style>
</html>
```

자산조회 페이지

자산조회에 필요한 정보를 입력하세요.

자산이름:

자산조회

result 페이지 추가

EXPLORER

- 블록체인수업 (WORKSPACE)
  - SSH FS - fabric-node
    - .cache
    - .config
    - .hfc-key-store
    - .local
    - .npm
    - .nvm
    - .vim
    - dev
      - fabcar
      - first-project
        - application
          - views
            - create.html
            - index.html
            - query.html
            - result.html
          - connection.json
          - enrollAdmin.js
          - package.json
          - registerUser.js
          - server.js
        - contract
        - network
        - tnetwork
      - hyperledger

index.html | create.html | query.html | result.html ●

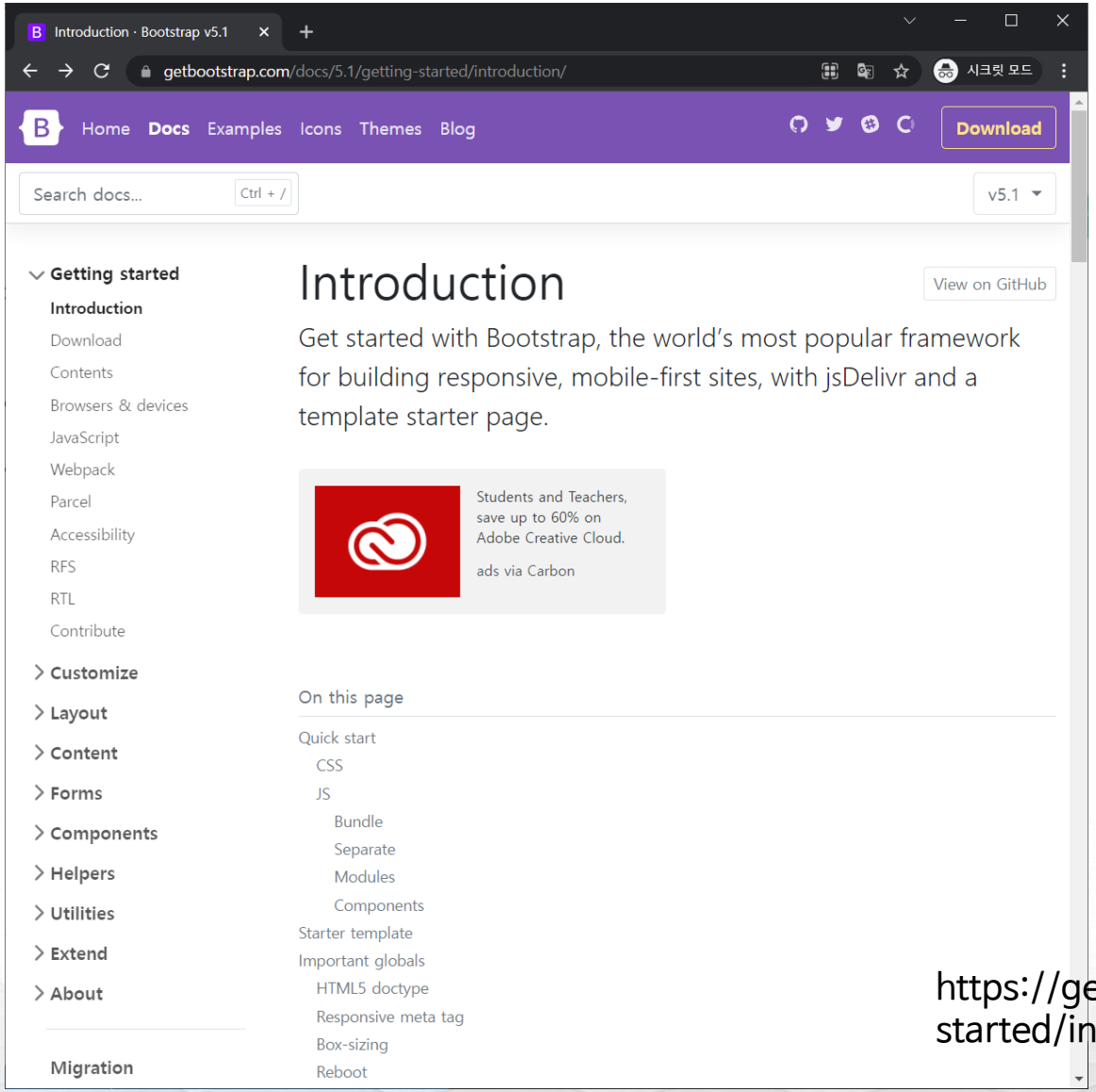
SSH FS - fabric-node > dev > first-project > application > views > result.html > html

```
1  <!DOCTYPE html>
2
3  <html>
4      <head>
5          <title>SIMPLE ASSET PROTOTYPE</title>
6      </head>
7
8      <body>
9          <H1> 자산생성/조회 결과 페이지 </H1>
10         <p> tx처리 결과입니다. </p>
11
12         <br>
13         <div></div>
14         <br>
15
16     </body>
17
18     <script>
19
20     </script>
21
22     <style>
23
24     </style>
25 </html>
```

자산생성/조회 결과 페이지

tx처리 결과입니다.

부트스트랩 라이브러리



https://getbootstrap.com/docs/5.1/getting-started/introduction/

CDN via jsDelivr

Skip the download with jsDelivr to deliver cached version of Bootstrap's compiled CSS and JS to your project.

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.cs
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle
```

Copy

If you're using our compiled JavaScript and prefer to include Popper separately, add Popper before our JS, via a CDN preferably.

`.container-xxl`	100%	100%	100%	100%	100%	1320px
`.container-fluid`	100%	100%	100%	100%	100%	100%

Default container

Our default `.container` class is a responsive, fixed-width container, meaning its `max-width` changes at each breakpoint.

Copy

```
<div class="container">
  <!-- Content here -->
</div>
```

- Getting started
- Customize
- Layout
- Content
- Forms
  - Overview
  - Form control
  - Select
  - Checks & radios
  - Range
  - Input group
  - Floating labels
  - Layout
  - Validation
- Components
- Helpers
- Utilities
- Extend
- About

Example

Email address

name@example.com

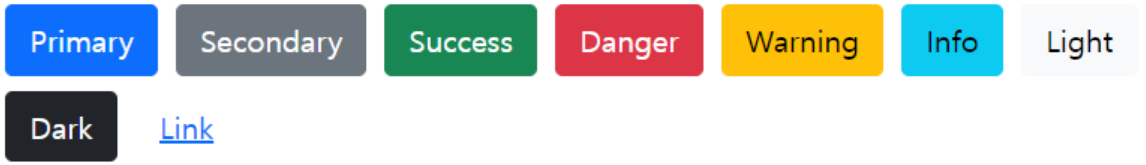
Example textarea

Copy

```
<div class="mb-3">
  <label for="exampleFormControlInput1" class="form-label">Email address</label>
  <input type="email" class="form-control" id="exampleFormControlInput1" placehold
</div>
<div class="mb-3">
  <label for="exampleFormControlTextarea1" class="form-label">Example textarea</la
  <textarea class="form-control" id="exampleFormControlTextarea1" rows="3"></texta
</div>
```

Examples

Bootstrap includes several predefined button styles, each serving its own semantic purpose, with a few extras thrown in for more control.



Copy

```
<button type="button" class="btn btn-primary">Primary</button>
<button type="button" class="btn btn-secondary">Secondary</button>
<button type="button" class="btn btn-success">Success</button>
<button type="button" class="btn btn-danger">Danger</button>
<button type="button" class="btn btn-warning">Warning</button>
<button type="button" class="btn btn-info">Info</button>
<button type="button" class="btn btn-light">Light</button>
<button type="button" class="btn btn-dark">Dark</button>

<button type="button" class="btn btn-link">Link</button>
```

index.html

```
1  <!DOCTYPE html>
2
3  <html>
4      <head>
5          <title>SIMPLE ASSET PROTOTYPE</title>
6
7          <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet"
           integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">
8          <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ka7S
           +IlRH9sENBO0LRn5q+8nbTov4+1p" crossorigin="anonymous"></script>
9
10     </head>
11
12     <body>
13         <div class="container">
14             <br>
15             <H1> Wecome to SIMPLE ASSET PROTOTYE</H1>
16             <p> 어릴적.... 순수한마음으로... </p>
17
18             <br>
19             <a href="./create.html" class="btn btn-danger">자산생성</a>
20             <br><br>
21             <a href="./query.html" class="btn btn-danger">자산조회</a>
22             <br>
23         </div>
24     </body>
25
26     <script>
27
28     </script>
29
30     <style>
31
32     </style>
33 </html>
```

create.html

```
<!DOCTYPE html>

<html>
    <head>
        <title>SIMPLE ASSET PROTOTYPE</title>

        <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet"
        integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">
        <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ka7Sk0Gl
        +IlRH9sENBO0LRn5q+8nbTov4+1p" crossorigin="anonymous"></script>
    </head>

    <body>
        <div class="container">
            <br>
            <H1> 자산생성 페이지 </H1>
            <p> 자산생성에 필요한 정보를 입력하세요. </p>
            <br>
            <form action="/asset" method="post">
                <label class="form-label">자산이름:</label>
                <br><input type="text" name="key" class="form-control">
                <br><br>
                <label class="form-label">자산값:</label>
                <br><input type="txt" name="value" class="form-control">
                <br><br>
                <input type="submit" value="자산생성" class="btn btn-primary">
            </form>
            <br>
        </div>
    </body>

    <script>
    </script>

    <style>
    </style>
</html>
```

query.html

```
<!DOCTYPE html>

<html>
    <head>
        <title>SIMPLE ASSET PROTOTYPE</title>

        <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet"
        integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">
        <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ka7Sk
        +IlRH9sENBO0LRn5q+8nbTov4+1p" crossorigin="anonymous"></script>
    </head>

    <body>
        <div class="container">
            <br>
            <H1> 자산조회 페이지 </H1>
            <p> 자산조회에 필요한 정보를 입력하세요. </p>
            <br>
            <form action="/asset" method="get">
                <label class="form-label">자산이름:</label>
                <br><input type="text" name="key" class="form-control">
                <br><br>
                <input type="submit" value="자산조회" class="btn btn-primary">
            </form>
            <br>
        </div>
    </body>

    <script>

    </script>

    <style>

    </style>
</html>
```

result.html

```
<!DOCTYPE html>

<html>
    <head>
        <title>SIMPLE ASSET PROTOTYPE</title>

        <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet"
        integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">
        <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ka7Sk
        +IlRH9sENBO0LRn5q+8nbTov4+1p" crossorigin="anonymous"></script>

    </head>

    <body>
        <div class="container">
            <br>
            <H1> 자산생성/조회 결과 페이지 </H1>
            <p> tx처리 결과입니다. </p>

            <br>
            <div></div>
            <br>
        </div>
    </body>

    <script>

    </script>

    <style>

    </style>
</html>
```

package.json

index.html ● | server.js | package.json ● | create.html | query.html

SSH FS - fabric-node > dev > first-project > application > package.json > license

```
1  {
2    "name": "simpleasset",
3    "version": "1.0.0",
4    "description": "",
5    "main": "server.js",
6    "scripts": {
7      "test": "echo \"Error: no test specified\" && exit 1"
8    },
9    "author": "",
10   "license": "ISC",
11   "dependencies" : {
12     "express": "~4.17.0",
13     "body-parser": "~1.19.0",
14     "fabric-ca-client": "~1.4.0",
15     "fabric-network": "~1.4.0"
16   }
17 }
18
```

npm install

```
bstudent@bstudentvb:~/dev/first-project/application$ ls
connection.json  enrollAdmin.js  package.json  registerUser.js  server.js  views
bstudent@bstudentvb:~/dev/first-project/application$ ll
total 28
drwxrwxr-x 3 bstudent bstudent 4096 Mar 18 02:24 ./
drwxrwxr-x 6 bstudent bstudent 4096 Mar 14 06:26 ../
-rw-rw-r-- 1 bstudent bstudent 1157 Mar 18 02:16 connection.json
-rw-rw-r-- 1 bstudent bstudent 1706 Mar 18 02:16 enrollAdmin.js
-rw-rw-r-- 1 bstudent bstudent  355 Mar 18 05:21 package.json
-rw-rw-r-- 1 bstudent bstudent 2395 Mar 18 02:16 registerUser.js
-rw-rw-r-- 1 bstudent bstudent    0 Mar 18 02:24 server.js
drwxrwxr-x 2 bstudent bstudent 4096 Mar 18 03:03 views/
bstudent@bstudentvb:~/dev/first-project/application$ npm install
npm WARN deprecated grpc@1.24.11: This library will not receive further updates other than security fixes. We recommend
npm WARN deprecated querystring@0.2.0: The querystring API is considered Legacy. new code should use the URLSearchParams
npm WARN deprecated request@2.88.2: request has been deprecated, see https://github.com/request/request/issues/3142
npm WARN deprecated cloudant-follow@0.17.0: This package is no longer maintained.
npm WARN deprecated har-validator@5.1.5: this library is no longer supported
npm WARN deprecated uuid@3.4.0: Please upgrade  to version 7 or higher.  Older versions may use Math.random() in certain
details.

> pkcs
> node

make:
```

```
bstudent@bstudentvb:~/dev/first-project/application$ ls
connection.json  enrollAdmin.js  node_modules  package.json  package-lock.json  registerUser.js  server.js  views
bstudent@bstudentvb:~/dev/first-project/application$ ls node_modules/ | grep fabric
fabric-ca-client
fabric-client
fabric-network
bstudent@bstudentvb:~/dev/first-project/application$ ls node_modules/ | grep express
express
bstudent@bstudentvb:~/dev/first-project/application$ ls node_modules/ | grep body-
body-parser
bstudent@bstudentvb:~/dev/first-project/application$
```

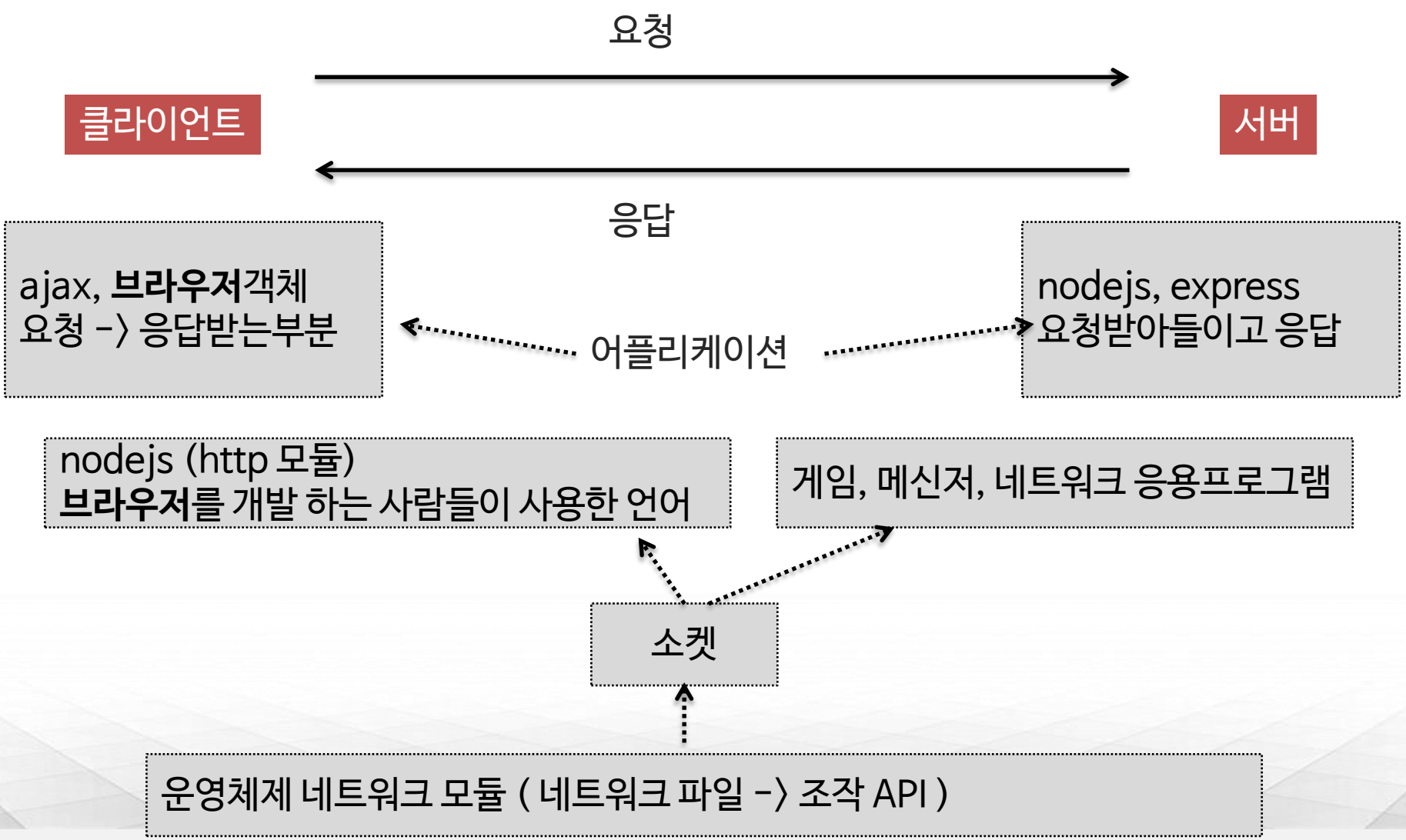
웹서버 작성

server.js

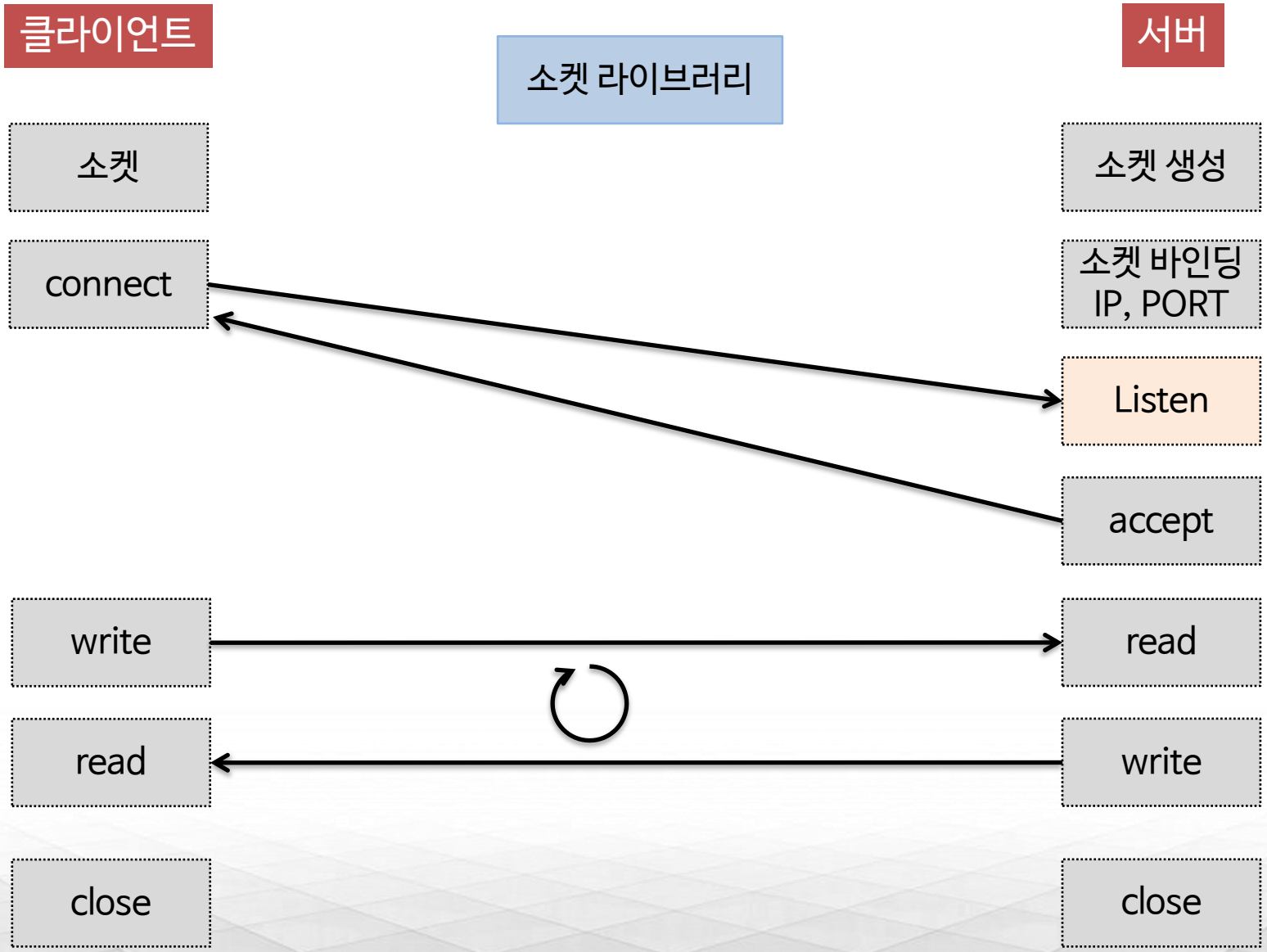
SSH FS - fabric-node > dev > first-project > application > server.js

```
1   // 1. 모듈포함
2
3   // 1.1 객체 생성
4
5   // 2. 서버설정
6
7   // 3. HTML 라우팅
8   // 3.1 index.html
9   // 3.2 create.html
10  // 3.3 query.html
11
12  // 4. REST api 라우팅
13  // 4.1 /asset POST
14  // 4.2 /asset GET
15
16  // 5. 서버시작
17
```

소켓


요청
클라이언트
서버
응답
ajax, 브라우저객체
요청 -> 응답받는부분
어플리케이션
nodejs, express
요청받아들이고 응답
nodejs (http 모듈)
브라우저를 개발 하는 사람들이 사용한 언어
게임, 메신저, 네트워크 응용프로그램
소켓
운영체제 네트워크 모듈 (네트워크 파일 -> 조작 API)

소켓 동작 원리


클라이언트
소켓 라이브러리
서버
소켓
connect
write
read
close
소켓 생성
소켓 바인딩
IP, PORT
Listen
accept
read
write
close

express

Fast, unopinionated, minimalist web framework for node.


```
const express = require('express')
const app = express()

app.get('/', function (req, res) {
  res.send('Hello World')
})

app.listen(3000)
```

https://www.npmjs.com/package/express

Installation

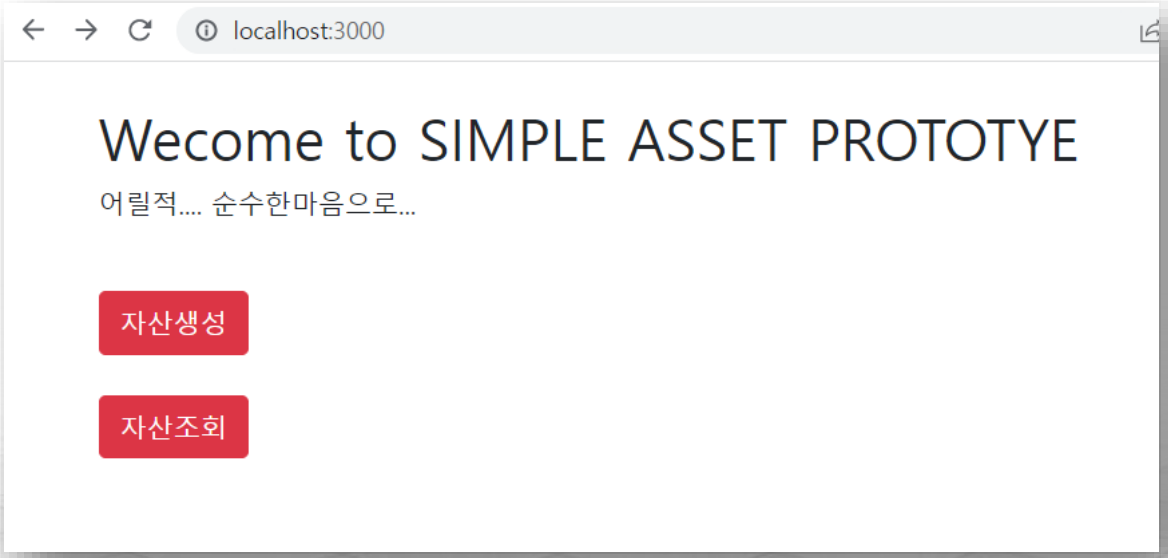
This is a Node.js module available through the npm registry.

Before installing, download and install Node.js. Node.js 0.10 or higher is required.

```
1  // 1. 모듈포함
2  // 1.1 객체 생성
3  const express = require('express');
4  const app = express();
5  var bodyParser = require('body-parser');
6
7  const { FileSystemWallet, Gateway } = require('fabric-network');
8
9  const fs = require('fs');
10 const path = require('path');
11
12 // 2. 서버설정
13 // 2.1 패브릭 연결설정
14 const ccpPath = path.resolve(__dirname, 'connection.json'); // /home/bstudent/dev/
   first-project/application/connection.json
15 const ccpJSON = fs.readFileSync(ccpPath, 'utf8');
16 const ccp = JSON.parse(ccpJSON); // unmarshal? []byte -> 구조체 객체화~
17
18 // 2.2 서버 속성 설정
19 const PORT = 3000;
20 const HOST = '0.0.0.0';
21
22 app.use(express.static(path.join(__dirname, 'views')));
23 app.use(bodyParser.json());
24 app.use(bodyParser.urlencoded({extended: false}));
```

```
26  // 3. HTML 라우팅
27  app.get('/', (request, response)=>{  // callback 함수
28      response.sendFile(__dirname + '/index.html');
29  })
30  // 3.1 index.html
31  // 3.2 create.html
32  // 3.3 query.html
33
34  // 4. REST api 라우팅
35  // 4.1 /asset POST
36  // 4.2 /asset GET
37
38  // 5. 서버시작
39  app.listen(PORT, HOST);
40  console.log(`Running on http://${HOST}:${PORT}`);
```

```
bstudent@bstudentvb:~/dev/first-project/application$ node server.js
Running on http://0.0.0.0:3000
```



/asset, POST 라우팅

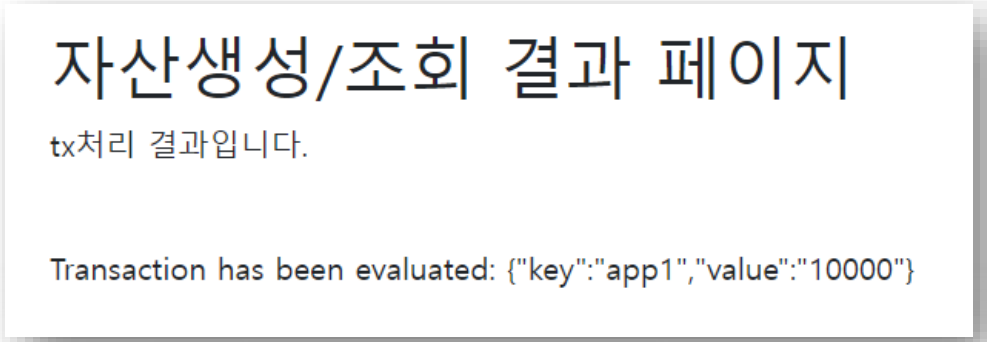
```
41  app.post('/asset', async(request, response)=>{
42      // 어플리케이션 요청문서에서 파라미터 꺼내기 ( POST method에서는 body에서 꺼냄 )
43      const key   = request.body.key;
44      const value = request.body.value;
45      console.log('/asset-post-'+key+'-'+value);
46      // 인증서작업 -> user1
47      const walletPath = path.join(process.cwd(), 'wallet') // ~/dev/first-project/application/wallet
48      const wallet = new FileSystemWallet(walletPath);
49      console.log(`Wallet path: ${walletPath}`);
50      const userExists = await wallet.exists('user1');
51      if(!userExists) {
52          console.log('An identity for the user "user1" does nto exist in the wallet');
53          console.log('Run the registerUser.js application before retrying');
54          // 클라이언트에서 인증서에 관한 안내 HTML을 보내줘야 함
55          response.status(401).sendFile(__dirname + '/unauth.html');
56          return;
57      }
58      // 게이트웨이연결
59      const gateway = new Gateway();
60      await gateway.connect(ccp, { wallet, identity: 'user1', discovery: { enabled: false } });
61      // 채널 연결
62      const network = await gateway.getNetwork('mychannel');
63      // 체인코드 연결
64      const contract = network.getContract('simpleasset');
65      // 트랜젝션처리
66      await contract.submitTransaction('set', key, value);
67      console.log('Transaction has been submitted');
68      // 게이트웨이연결 해제
69      await gateway.disconnect();
70      // 결과 클라이언트에 전송
71      // result.html수정
72      const resultPath = path.join(process.cwd(), '/views/result.html')
73      var resultHTML = fs.readFileSync(resultPath, 'utf8');
74      resultHTML = resultHTML.replace("<div></div>", "<div><p>Transaction has been submitted</p></div>");
75      response.status(200).send(resultHTML);
76  })
```

/asset, GET 라우팅

```
78  app.get('/asset', async(request, response)=>{
79      // 어플리케이션 요청문서에서 파라미터 꺼내기 ( POST method에서는 query에서 꺼냄 )
80      const key   = request.query.key;
81
82      console.log('/asset-get-'+key);
83      // 인증서작업 -> user1
84      const walletPath = path.join(process.cwd(), 'wallet') // ~/dev/first-project/application/wallet
85      const wallet = new FileSystemWallet(walletPath);
86      console.log(`Wallet path: ${walletPath}`);
87      const userExists = await wallet.exists('user1');
88      if(!userExists) {
89          console.log('An identity for the user "user1" does nto exist in the wallet');
90          console.log('Run the registerUser.js application before retrying');
91          // 클라이언트에서 인증서에 관한 안내 HTML을 보내줘야 함
92          response.status(401).sendFile(__dirname + '/unauth.html');
93          return;
94      }
95      // 게이트웨이연결
96      const gateway = new Gateway();
97      await gateway.connect(ccp, { wallet, identity: 'user1', discovery: { enabled: false } });
98      // 채널 연결
99      const network = await gateway.getNetwork('mychannel');
100     // 체인코드 연결
101     const contract = network.getContract('simpleasset');
102     // 트랜젝션처리
103     const txresult = await contract.evaluateTransaction('get', key);
104     console.log('Transaction has been evaluated: '+txresult);
105     // 게이트웨이연결 해제
106     await gateway.disconnect();
107     // 결과 클라이언트에 전송
108     // result.html수정
109     const resultPath = path.join(process.cwd(), '/views/result.html')
110     var resultHTML = fs.readFileSync(resultPath, 'utf8');
111     resultHTML = resultHTML.replace("<div></div>", `<div><p>Transaction has been evaluated: ${txresult}</p></div>`);
112     response.status(200).send(resultHTML);
113 })
```

연동테스트 및 데모

- **네트워크 수행 (기관 3개짜리, first-project/network)**
  - 네트워크 초기화
    - docker rm -f $(docker ps -aq)
    - docker rmi -f $(docker images dev-* -q)
    - docker network prune
  - **./start.sh**
- **체인코드 설치배포/테스트 (simpleasset.go)**
  - cd ~/dev/first-project/contract/simpleasset/1.1
  - ./install.sh ex) **./install.sh simpleasset 1.1 instantiate mychannel**
- **웹서버 수행 (nodejs)**
  - 인증키생성복사 (fabric-ca-client) admin, user1
    - cd ~/dev/first-project/application
    - rm -rf wallet
    - node enrollAdmin
    - node registerUser
  - 서버수행 (localhost, 3000)
    - **node server**
- **웹 UI접속**
  - 브라우저 사용 (localhost:3000) -> 자산생성 확인
  - 데이터 생성/조회
- **DB접속하여 블록체인 데이터 확인 (localhost:5894/_utils)**



first-project

- network
  - 3 org 확장된 네트워크
    - generate.sh, start.sh
    - ./start.sh -〉 peer x 3, cli, orderer, ca컨테이너 net_basic 네트워크
    - mychannel 사용가능한 상태
- contract
  - simpleasset
    - 1.1
      - » install.sh
      - » ./install.sh simpleasset 1.1 instantiate mychannel
      - » -〉 체인코드가 설치 배포 invoke, query 수행확인

이 과정으로 습득해야 할 능력

- **블록체인에 대한 이해**
  - 공유경제를 위한 퍼블릭 블록체인 네트워크 (개방형)
  - 산업의 비용과 시간을 절약시켜주는 기업용 블록체인 네트워크 (허가형)
  - 요소기술에 대한이해(보안, 합의, 네트워크, 스마트컨트랙트, SDK)
  - 최신 적용 기술의 동향과 종류들.. (코인이코노미, NFT, CBCD, 유통, DID, DEFI…)
- **프로그래밍 스킬**
  - 고언어를 통한 체인코드 개발, **자바스크립트나 nodejs… 웹언어를 기반한 웹서비스 프로토타입** 개발
  - 시스템 운용을 위한 쉘프로그램…
- **블록체인 플랫폼의 프로토타입작성**
  - 체인코드 설계구현운용
  - 패브릭 네트워크 설정 및 운용
  - 웹서비스 프로타토타입 개발과 블록체인 연동
- **블록체인 서비스 기획력, 블록체인 프로젝트 진행능력**

수료증 -〉 출석

패브릭 프로젝트 포토폴리오
1. 교육노트..-〉 블러그(노션..)
2. B/C프로토타입 소스코드(깃허브)
3. 데모 (리드미, 유투브영상)

상장(입상 ~ 우수상) 공모전, 해커톤.. 도전과정, 결과

복습

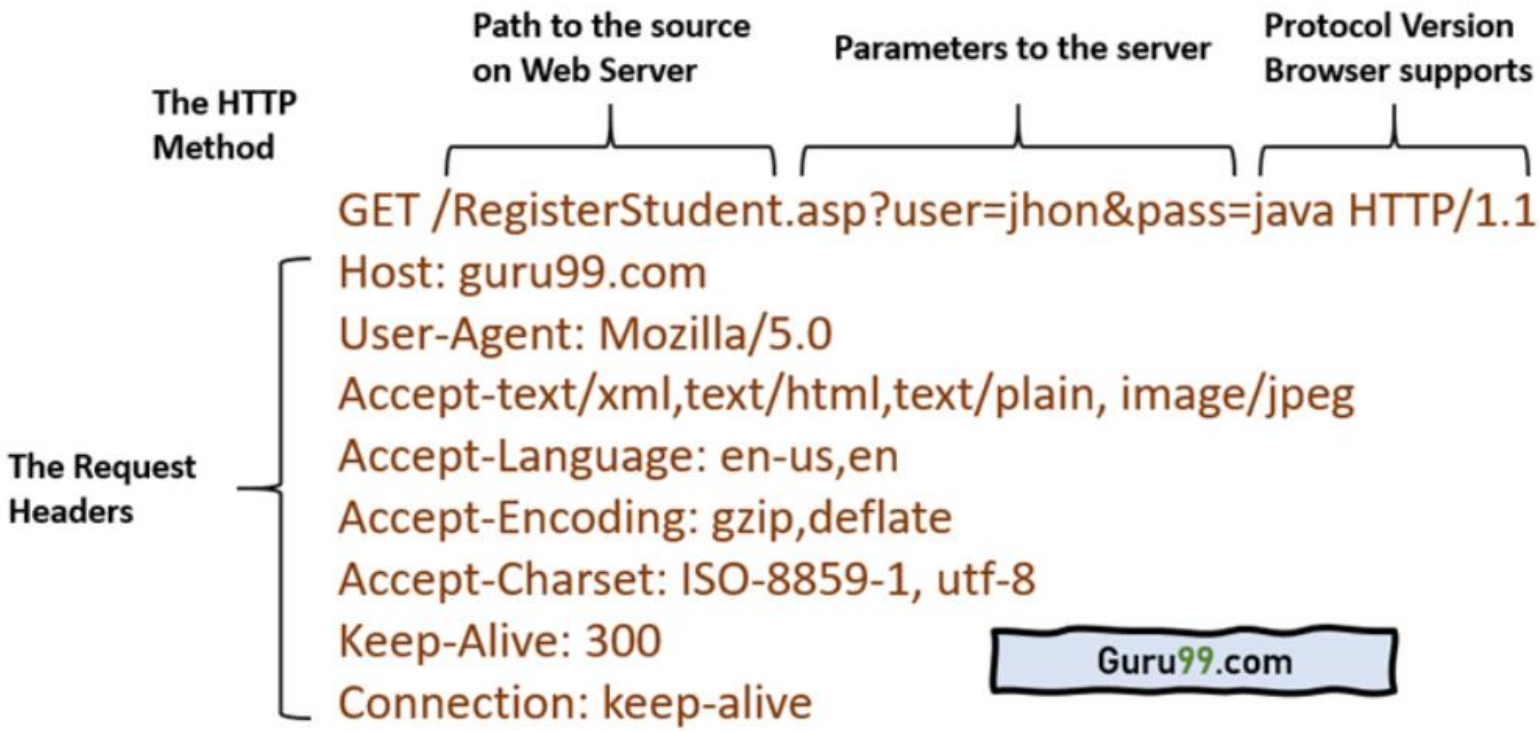
- **네트워크**
  - basic-network->확장 네트워크, test-network,
  - first-network-> second-network
  - 쉘스크립트, 설정파일들
- **체인코드**
  - 프라이빗 데이터 (marbles02_private) Put Get Del PrivateData , GetTransient
  - 채널내 기관별 권한관리, SideDB의 분산화정도, 자동삭제
- **웹서비스**
  - simpleasset 웹연동(fabric-network, fabric-ca-client), 웹서버(nodejs-express, 웹클라이언트(html+bootstrap-css)

http 요청문서

GET Method Example

Here is an example of GET method:

```
GET/RegisterStudent.asp?user=value1&pass=value2
```



POST Method Example

Here is an example of POST method:

```
POST/RegisterStudent.asp HTTP/1.1
Host: www.guru99.com
user=value1&pass=value2
```

A form using the default application/x-www-form-urlencoded content type: