

Express 도입부터 POST 정보 전달까지

Node.js

1990

WEB 의 탄생

팀버너스리가 만듦

1994

Javascript 의 탄생

web을 능동적으로 사용할 수 있도록 만듦

2004

Google⁰이 Gmail, Gmap을 만듦

사람들이 생각함('이런 미친, javascript 개 찔잖아!')
이때 부터 javascript 커뮤니티가 폭발적으로 늘어남

2008

Google이 Chrom을 개발함

성능을 향상시키기 위하여 Javascript 개발 엔진 'V8'만듬
오픈소스로 공개를 함 (javascript의 탈 웹화의 가속화)

2009

Node.js 프로젝트가 시작됨

라이언 달 이라는 사람이 만듬

V8 + Event-driven + Non-blocking IO = Node.js
(Web browser -> Server)

Javascript의 측면 2가지

1. Language

언어 그 자체(javascript의 문법을 사용한다.)

2. Runtime

실행환경(Web Browser, Nodejs)

Nodejs

```
alert('hello world')
```

-> 분명히 javascript 문법이지만 에러가 뜬다.

왜냐하면, web browser에서만 사용할 수 있다.

같은 언어를 사용하여

서버와 클라인트가 협력적인 관계 형성

경쟁자: java, ruby, python 등이 있지만 v8의 성능이 좋아서 개 빠름

Express란 무엇인가?

우리는 첫 시간에 서버를 만들어 보았다.

코드는 다음과 같다.

```
const http = require('http');

const hostname = '127.0.0.1';
const port = 3000;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello World\n');
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
```

아니.. 서버 만들기 힘들어.. 왜 저렇게 많이 쳐야 하는 거야..

그래서 탄생한 Express

인간 나태함의 결정체

Express

서버를 쉽게 만들게 해주는 Frame Work 인데..

```
var express = require('express');  
var app = express();  
app.listen(3000,function(){  
    console.log('Connected 3000 port!');  
});
```

실습해보세요.

```
var express = require('express');
var app = express();

app.get('/',function(req,res){
    res.send('hello');
});
app.get('/login',function(req,res){
    res.send('Login Please');
});
app.listen(3000,function(){
    console.log('Connected 3000 port!');
});
```

이번에는

HTML 파일 올린거 복붙합니다.

시간 절약을 위해서

정적파일 연결

```
app.use(express.static('public'));
```

받은 쿼리 값으로 처리

```
app.get('/form_receiver',function(req,res){  
    var title = req.query.title;  
    var description = req.query.description;  
    res.send(title + ',' + description);  
});
```

html에 post함수 넣습니다.

query를 사용하지 않고 post를 사용할 겁니다.

사용하기 위해서는 express의 미들웨어인 *body-parser*를 사용해야 합니다.

마지막 코드

```
var express = require('express');
var app = express();
var bodyParser = require('body-parser');
app.use(express.static('views'));
app.use(bodyParser.urlencoded({ extended: false }));
app.get('/',function(req, res){
    res.send('hello');
});

app.post('/form_receiver',function(req,res){
    var title = req.body.title;
    var description = req.body.description;
    res.send(title + ',' + description);
});
app.listen(3000, function(){
    console.log('Conneted 3000 port!');
});
```


왜 post를 이용하냐?

- query를 사용하는 것보다 보안이 된다.
(물론 그냥 데이터가 안보인다는 정도임.)
- query를 용량 제한이 있음. 그런데 저렇게 post에 담아서 보내면 그런게 없음. 객체를 전달한다는 느낌?