

Back-End Programming for Front-End Developers

Day1

프론트엔드 개발 SCHOOL 2기, FASTCAMPUS

2016.11.21

Introduce Myself!

- github: <https://github.com/ulgoon>
- blog: <https://ulgoon.github.io>
- Solution Architect, Back-End Developer
- Web development
 - Python(django), Node.js(Express.js), golang, angularjs, ...
- Interested in Data science
 - Julia, Python, ...

학습목표

- 웹 개발 프로세스와 백엔드에 대해 이해한다.
- 백엔드 개발자와 상생할 수 있다.
- npm을 이용해 필요한 모듈을 설치하고 사용할 수 있다.
- Node.js를 이용한 라우팅과 렌더링에 대해 이해한다.
- Express를 이용해 웹 어플리케이션을 작성할 수 있다.
- Socket.io를 활용하여 실시간 채팅 서비스를 구현할 수 있다.
- Heroku를 통해 작성한 프로젝트를 배포할 수 있다.

Index

Understanding Back-End

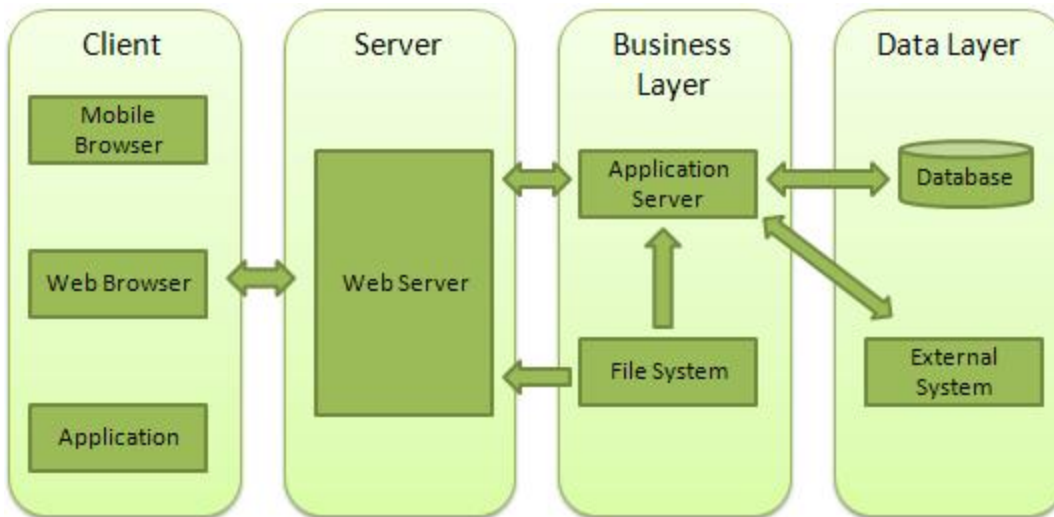
- 백엔드의 개념과 구조
- 백엔드 개발자는 어떻게 생각하는가
- 백엔드 개발자는 어떻게 일하는가

MEAN stack

- Node.js
- Express.js
- gulp Task runner
- Deploy with Heroku

Understanding Back-End

Web architecture



웹 개발 패턴의 변화

- 1991 ~ 1999: Sir Timothy John "Tim" Berners-Lee가 하이퍼텍스트 기반의 프로젝트를 제안한 이후 정적인 콘텐츠를 중심으로 한 웹 기술이 발달
- 1999 ~ 2009: Linux, Apache, Mysql, Php 중심의 동적인 서버, 정적인 클라이언트 모델이 지속됨
- 2010 ~ 현재: Javascript!! (Dynamic Web Client)

백엔드의 개념과 구조

Back-End

- Server, application, database
- **Interact** with Front-End(Middleware)

Language&Framework

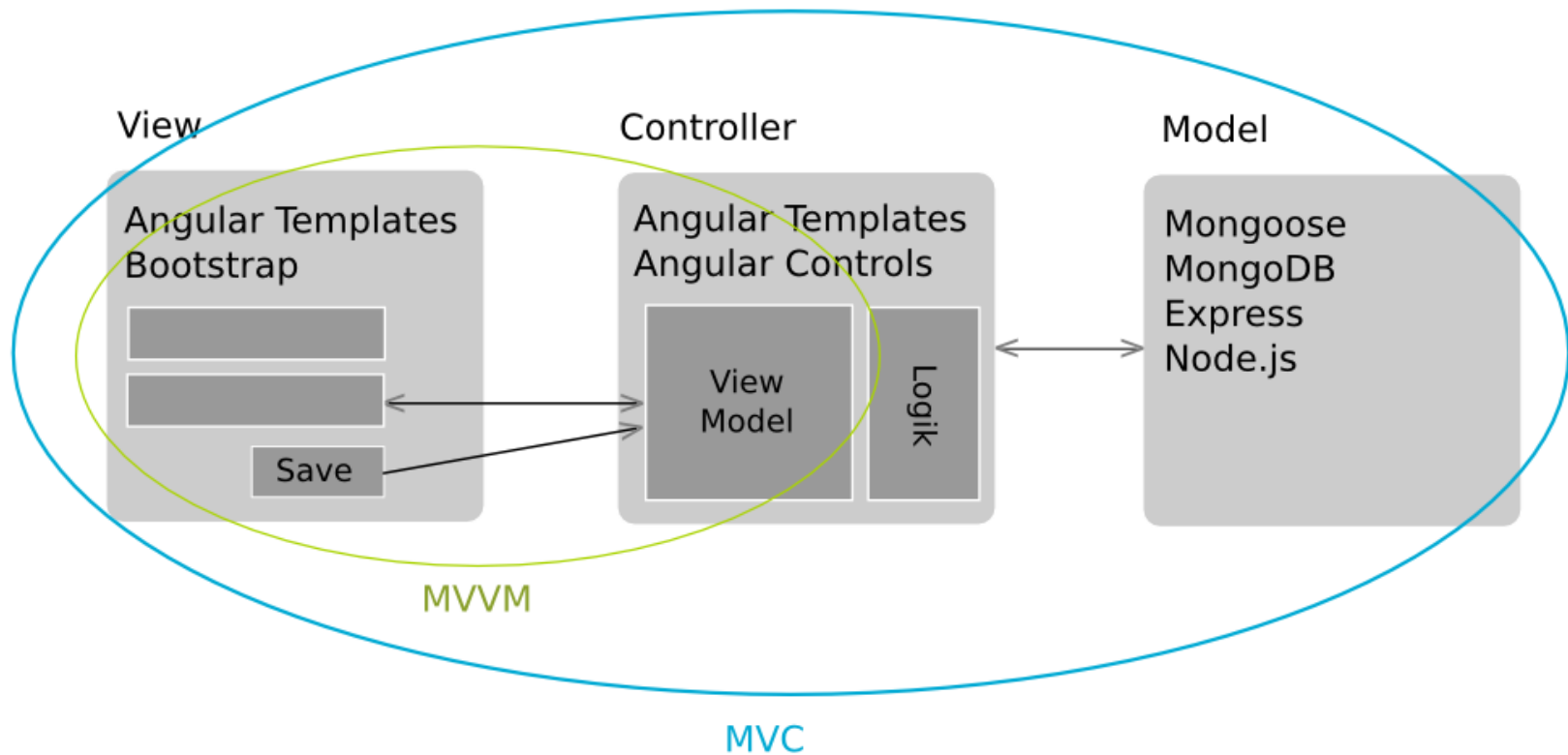
- PHP, Java, C#, Ruby, Python, Go, ...
(Laravel, Spring, [asp.net](#), RoR, Django, Gorilla, ...)

백엔드 개발자는 어떻게 생각하는가

MVC

MEAN Stack

(MongoDB, Express.js, Angular.js, Node.js)



CC-BY-SA Stephan Kristyn, [Netzathleten](#)
Media

URL, URN, URI

- URL: Uniform Resource Locator
(<http://www.example.com/>)
- URN: Uniform Resource Name
(urn: id=29764825)
- URI: Uniform Resource Identifier
(<http://www.example.com/people?id=29764825>)

License

- GPL(GNU Public License)
- FreeBSD
- MIT License

백엔드 개발자는 어떻게 일하는가

환경설정

매우매우매우매우매우매우 중요!

- node: package.json, nvm
- django: requirements.txt, (pyenv, virtualenv, autoenv)

git

협업부터 버전관리까지 모든 것을 관리하기 위한 도구

git-scm.com



REST API

- REpresentational Safe Transfer Application Programming Interface
- by Roy Fielding, 2000

구분				
CRUD	Create	Read	Update	Delete
HTTP/REST	Post	Get	Put	Delete
SQL	INSERT	SELECT	UPDATE	DELETE

REST API 설계시 주의할 점

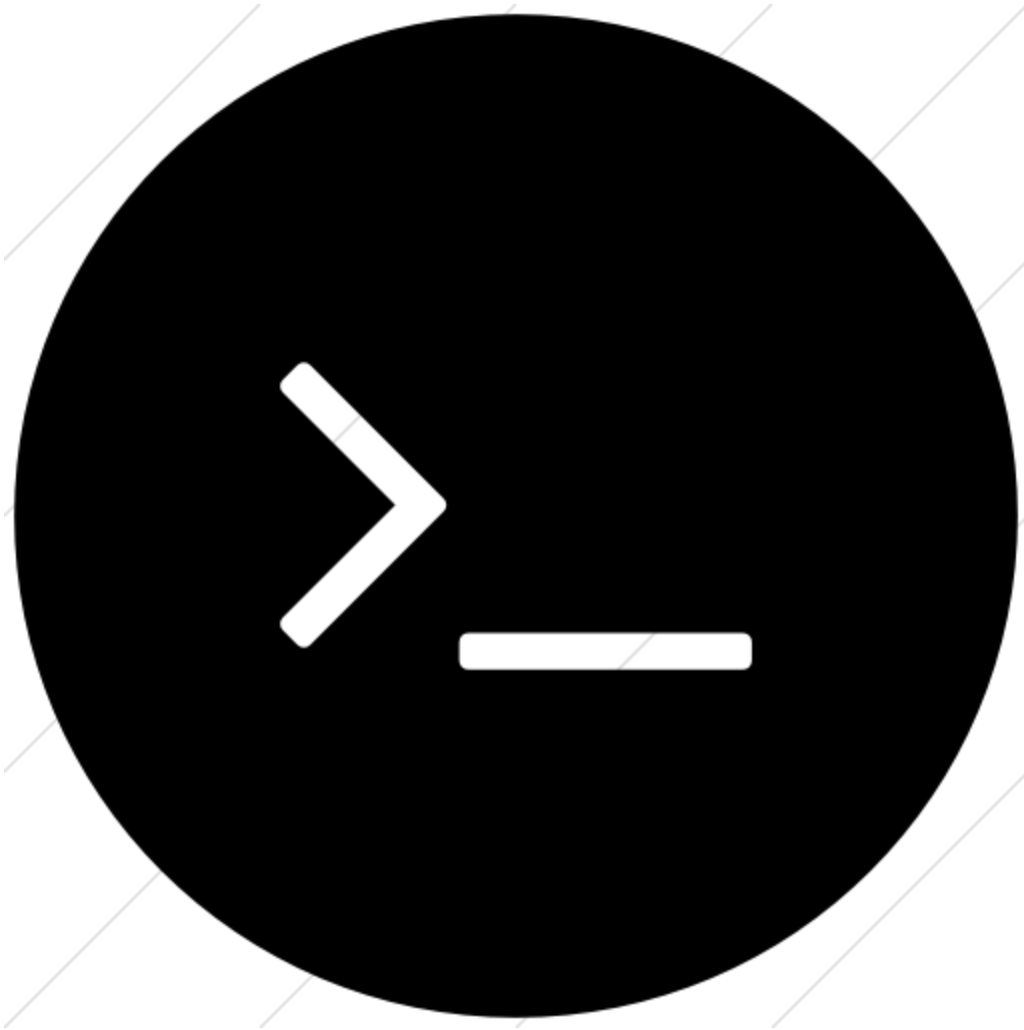
- 버전관리 <https://api.foo.com/v1/bar>
- 명사형 사용 <https://foo.com/showid/> --> <https://foo.com/user/>
- 반응형 <https://foo.com/m/user/>, <https://m.foo.com/user/> (x)
- 언어코드 <https://foo.com/kr/>, <https://kr.foo.com/> (x)
- 응답상태 코드 (200, 400, 500)

**BUT IT'S NOT
"RESTful" IF YOU...**



ENOUGH!

OS



Editor



Dashboard

- **Admin:** Built-in or 3rd party
- **Google Analytics, Google Tag Manager:** general
- **Firebase Analytics:** additional

웹 개발 프로세스

서비스 제안 기획 및 개발

- 기획자

스토리보드(유저 시나리오), DB정책, 페이지별 Data Flow 기획 - 프로젝트 관리 - Test

- Back-End

유저 시나리오 검토 - API 설계 - Back-End 개발 - Unit Test - Load Test

- Front-End

유저 시나리오 검토 - API 설계 - Front-End 개발 - Mock/Unit Test - API 연동

배포 및 유지보수

Do not Deploy with **FTP** --> sftp or ftps

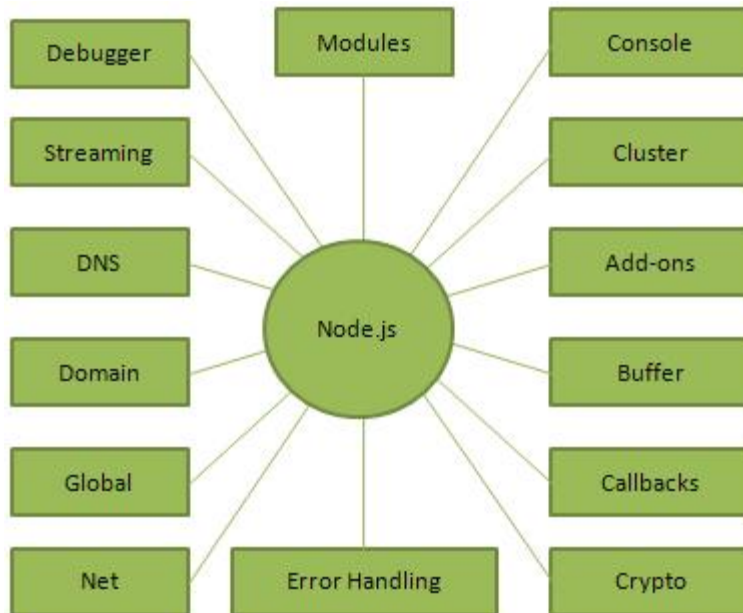
- IaaS or PaaS
 - 트래픽에 따라 자유롭게 용량할당 가능
 - 로컬로 구성하는 것보다는 조금은 비싼 가격(트래픽에따라 다름)

MEAN stack

"Write programs that do one thing and do it well."

Node.js

- by Ryan Dahl, 2009
- Google V8 JS Engine 기반
- Single Thread Event loop 기반 비동기 IO 처리
- 프론트와 백엔드 기술 통합 -> 생산성 향상



Tutorial with MEAN stack

Node.js 설치

- <https://nodejs.org/>
- or `brew(apt-get) install node`
- `node --version`

MongoDB 설치

- <https://mongodb.com/downloads>
- or `brew(apt-get) install mongo`
- follow <https://docs.mongodb.com/v3.0/tutorial/install-mongodb-on-windows/>

npm

npm != Node Package Manager

<https://npmjs.com>

```
$ npm version
```

```
$ npm install <something>
```

```
$ npm install express -g
```

```
$ npm install gulp -g
```

```
//install devdependency
```

```
$ npm install gulp-<something> --save-dev
```

```
$ npm install body-parser --save
```

First Node application

```
$ touch app.js  
$ npm init  
$ vi app.js
```

```
var http = require("http");  
//http Status(200): OK, send response "hello world"  
http.createServer(function(request, response){  
    response.writeHead(200,"Content-Type":"text/plain");  
    response.write("hello world \n");  
    response.end();  
}).listen(process.env.PORT || 3030);  
  
//server listen  
console.log("Server is running at localhost:3030");
```

routing rule

- 유의미한 정보를 둘것
- 너무 긴 URL은 SEO에 악영향
- `_` 대신 `-` 로
- 공백이나 영어, 숫자외의 문자는 되도록이면 지양
 - '/2 만원만 주세요' --> '/show-me-the-money'
- 소문자!

route

```
$ mkdir layout  
$ touch home.html
```

```
$ touch router.js  
$ vi app.js  
  
var router = require("./router.js");  
router.home(request, response);  
router.slug(request, response);  
  
$ vi router.js  
function home(request, response){  
    ...  
};  
  
module.exports.home = home;
```