

Back-End Programming for Front-End Developers

Day2

프론트엔드 개발 SCHOOL 2기, FASTCAMPUS

2016.11.22

YWL(Yesterday, We Learned..)

- 웹 개발이란
- 백엔드가 뭐야?
- node.js
- routing

rendering

```
<body>  
  <header></header>  
  <p></p>  
  <footer></footer>  
</body>
```

```
{% header %}  
{% content %}  
{% footer %}
```

in router.js

```
var baseResource = fs.readFileSync("./views/layout.html",  
var headerResource = fs.readFileSync("./views/header.html",  
var footerResource = fs.readFileSync("./views/footer.html",  
var homeResource = fs.readFileSync("./views/home.html", "  
  
baseResource = baseResource.replace("{% header %}", headerResource;  
baseResource = baseResource.replace("{% footer %}", footerResource;  
baseResource = baseResource.replace("{% content %}", homeResource;  
  
response.write(baseResource);  
response.end();
```

readFile, readFileSync

```
fs.readFile("./views/layout.html", function(request, response) {
    fs.readFile("./views/header.html", function(request, response) {
        fs.readFile("./views/home.html", function(request, response) {
            fs.readFile("./views/footer.html", function(request, response) {
                ...
            });
        });
    });
});
```

renderer.js

- router.js --> only routing
- renderer.js --> only rendering

renderer.js(router.js)

```
var renderer = require('./renderer');  
  
if(request.url === '/') {  
    return renderer(request, response, "home");  
}
```

renderer.js

```
var fs = require('fs');  
  
module.exports = function(request, response, layoutName){  
}
```


package.json

- 프로젝트를 설명하고, 의존성 목록을 생성
- `$ npm install --save` 로 정의
- `$ npm install` 로 명시된 패키지 다운로드
- `.gitignore` 에 `node_module/` 추가

```
"dependencies": {  
    "express": "^4.13.1",  
    ...  
}
```

npm 패키지는 어떻게 설치하는가

```
$ npm install express --save
```

```
$ npm install -g nodemon
```

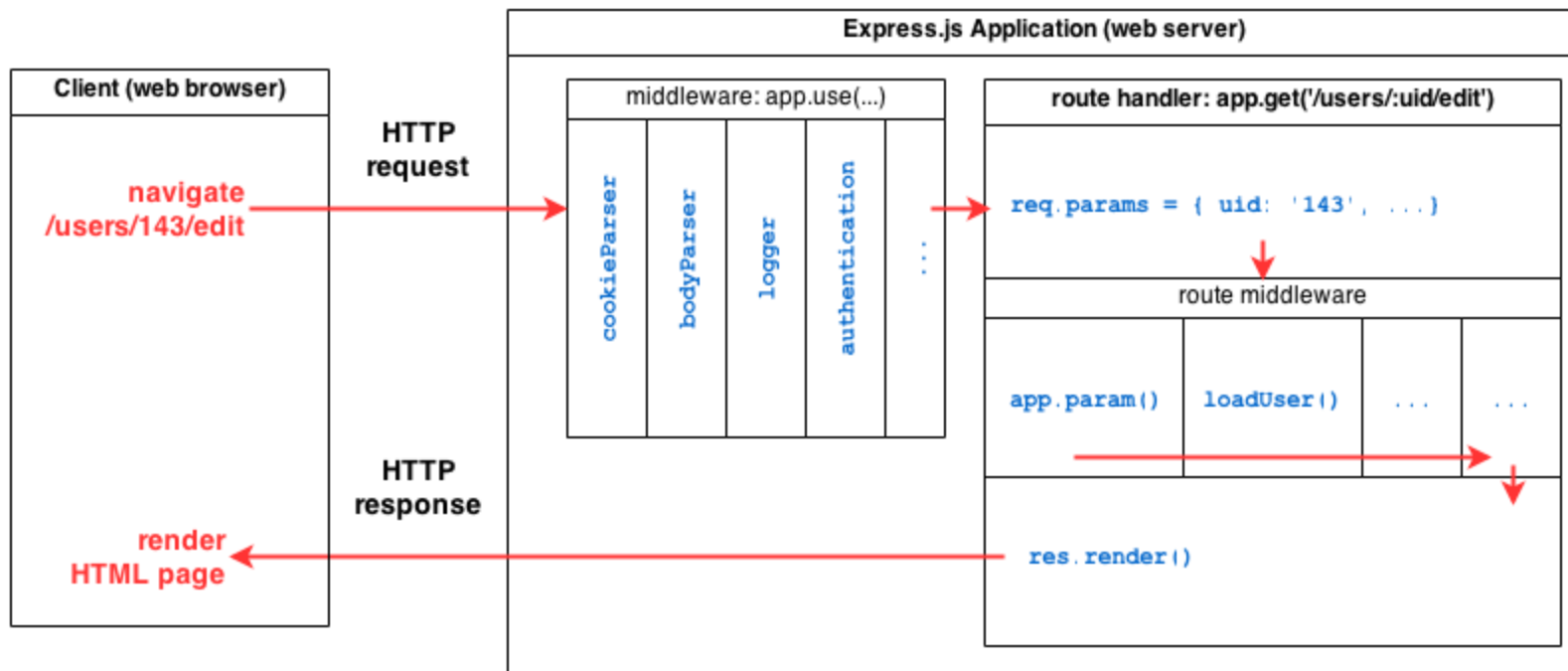
```
$ npm install gulp --save-dev
```

express

- 웹개발 및 REST API 서버에 특화된 프레임워크
- Hapi.js, Koa.js를 제치고 node.js에서 가장 유명
- 가볍고 유연한 미들웨어구조

```
$ npm install express --save
```

Express.js Application Flow



First express server

```
$ touch server.js
```

```
$ npm init
```

```
$ npm install express --save
```

```
var express = require('express');  
var app = express();  
  
app.get('/', function(req, res){  
    res.send('Hello Express!');  
});  
  
app.listen(process.env.PORT || 3030, function(){  
    console.log('Express Server is running at localhost:3030');  
});
```