

# Back-End Programming for Front-End Developers

## Day4

프론트엔드 개발 SCHOOL 2기, FASTCAMPUS

2016.11.24



## socket.io

```
npm install --save socket.io
```

- 실시간 통신기술의 웹 브라우저 호환성 문제 해결을 위한 프로젝트
- IE6 부터 최신 브라우저까지 지원
- WebSocket, Flash Socket, AJAX Long Polling, AJAX Multipart Streaming, Forever iframe, JSONP Polling 기술 모두 포함
- 브라우저에 따라 최적화된 기술 사용
- 일관성있는 문법과 API로 개발 가능

## chat.html

```
<style>
  * { margin: 0; padding: 0; box-sizing: border-box; }
  body { font: 13px Helvetica, Arial; }
  form { background: #000; padding: 3px; position: relative; }
  form input { border: 0; padding: 10px; width: 90%; }
  form button { width: 9%; background: rgb(130, 224, 255); }
  #messages { list-style-type: none; margin: 0; padding: 0; }
  #messages li { padding: 5px 10px; }
  #messages li:nth-child(odd) { background: #eee; }
</style>

<ul id="messages"></ul>
<form action="">
  <input id="m" autocomplete="off" /><button>Send</button>
</form>
```

## connection

```
var http = require("http").Server(app);
var io = require("socket.io")(http);

io.on("connection", function(socket){
    console.log("a user connected");
});

http.listen(8080, function(){
    console.log("Chat Server is running at localhost:8080");
});
```

```
<script src="/socket.io/socket.io.js"></script>
<script>
    var socket = io();
</script>
```

## disconnect

```
socket.on("disconnect", function(){  
    console.log("user disconnected");  
});
```

## emit message

```
socket.on("chat message", function(msg){  
    console.log("message: " + msg);  
});
```

```
<script src="http://code.jquery.com/jquery-1.11.1.js"></script>  
  
var socket = io();  
$("form").submit(function(){  
    socket.emit("chat message", $("#m").val());  
    $("#m").val("");  
    return false;  
});
```

## broadcast

```
io.emit("chat message", msg);
```

```
socket.on("chat message", function(msg){  
    $("#messages").append($("<li>").text(msg));  
});
```



## **Web scraping**

## request

```
$ npm install request --save
```

```
var request = require('request');

request({
  method: 'GET',
  uri: 'https://www.rottentomatoes.com/browse/in-theate
}, function(err, res, body){
  if (err) throw err;
  console.log(body);
});
```

## Analyze HTML

with CSS selector, jquery selector

## cheerio

```
$ npm install cheerio --save
```

```
var $ = cheerio.load(body);  
var movieInfo = $("table#Top-Box-Office.movie_list");  
console.log(movieInfo); movieInfo.each(function(){  
    var movieName = $(this).find("td.left_col").text();  
    var meterScore = $(this).find("td.middle_col").text();  
    console.log(movieName);  
    console.log(meterScore);  
});
```

## cheerio

```
var movieInfo = $("table#Top-Box-Office.movie_list");
movieInfo.each(function(){
    var meterScore = $(this).find("td.left_col").text
    var movieName = $(this).find("td.middle_col").text
    boxOffice.push({meterScore:meterScore, moverName:moverName})
})
```

**gulp**

## task flow

코드작성 - JS test - JS Minify - JS Merge - CSS Minify - CSS

Merge - 결과물

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

```
$ touch gulpfile.js  
  
var gulp = require("gulp");  
  
//hello라는 gulp task를 등록  
gulp.task("hello", function () {  
    return console.log("hello gulpworld");  
});  
  
$ gulp hello
```

## 기본값 설정하기

```
$ gulpfile.js

var gulp = require("gulp");

//hello라는 gulp task를 등록
gulp.task("hello", function () {
    return console.log("hello gulpworld");
});

gulp.task("default", ["hello"]);

$ gulp
```



## 우선순위 설정하기

```
$touch gulpfile.js

var gulp = require("gulp");

//hello라는 gulp task를 등록
gulp.task("hello", function () {
    return console.log("hello");
});

gulp.task("gulpworld", ["hello"], function () {
    return console.log("gulpworld");
});

gulp.task("default", ["gulpworld"]);

$ gulp
```

## Minify(uglify)

```
$ npm install gulp-uglify --save-dev  
  
var uglify = require("gulp-uglify");  
  
gulp.task("uglify", function(){  
    return gulp.src("public/src/*.js").pipe(uglify())  
});  
  
gulp.task("default", ["uglify"]);
```

## watch

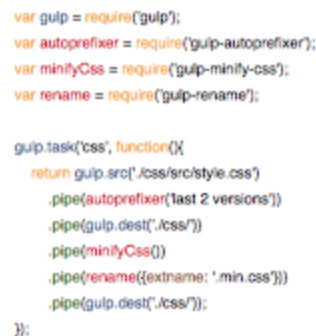
```
gulp.task("watch", function(){  
    gulp.watch("public/src/*.js", ["uglify"]);  
});  
  
gulp.task("default", ["uglify", "watch"]);
```

## 이외에도..

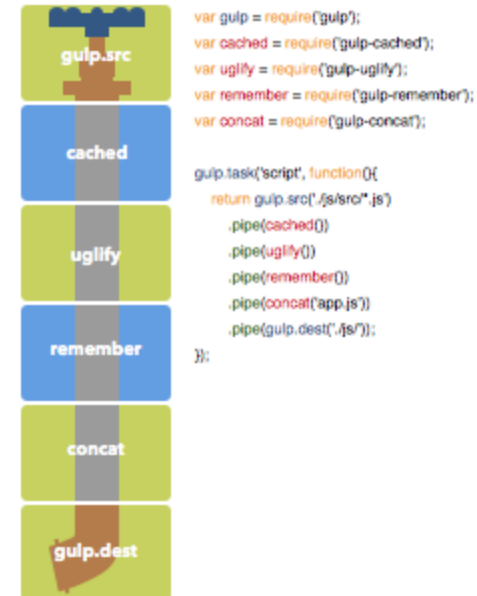
### single dest & watch



### multi dest



### incremental rebuilding



### task dependency

```
gulp.task('css', ['font', 'less'], function(){
  // do something after font and less
});
```

\$ npm install -g gulp install global

\$ npm install --save-dev gulp install local

\$ npm install --save-dev gulp-other-plugins

\$ gulp task\_name run task

\$ gulp task\_name other\_task run multi tasks

# gulp.js

cheatsheet  
The streaming waterworks



only changed

this sheet is made in **OpenSource Cafe**  
for OpenSource