

Back-End Programming for Front-End Developers

Day3

프론트엔드 개발 SCHOOL 2기, FASTCAMPUS

2016.11.23

YWL(Yesterday, We Learned..)

- node.js rendering
- package.json
- express
- routing, rendering

Serve Express One more Time!

Send Text(Hello Express) --> Home route with `routes/index.js` -
-> About route with `routes/about.js` --> `$ npm install ejs --`
`save`

partials with ejs

index.ejs(not layout.ejs)

```
<html>
  <head>
    <% include partials/head.ejs %>
  </head>
  <body>
    <% include partials/header.ejs %>

    <!-- content here -->

    <% include partials/footer.ejs %>
  </body>
</html>
```

partials with ejs in another way

index.ejs

```
<% include partials/header.ejs %>  
  
<!-- content here -->  
  
<% include partials/footer.ejs %>
```

Serve Static Files

in server.js

```
app.use("/public", express.static(__dirname +  
"/public"));
```

in head.ejs

```
<link rel="stylesheet" type="text/css"  
href="/public/src/css/style.css">
```

simple json

```
var users = [{  
  username:1,  
  name: kim,  
  email: kim@gmail.com  
}];
```

```
app.get('/users', function(req, res){  
  res.json(users);  
});
```

simple json

```
app.get('/users/random', function(req, res){  
    var n = Math.floor(Math.random() * users.length);  
    var u = users[n];  
    res.json(u);  
});
```


simple json

```
app.get('/users/:id', function(req, res){  
    if (users.length <= req.params.id || req.params.id < 0){  
        res.statusCode = 404;  
        return res.send("user not found");  
    }  
    var u = users[req.params.id];  
    res.json(u);  
});
```

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");  
});
```

```
<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));  
});
```