Web Service Development

What we learned

What We Learned

- HTML
- CSS
- Chrome developer tools
- jQuery
- JavaScript
- AJAX
- Session
- Flask
- GET/POST
- Client/Server
- Database
- ORM
- MVC
- Python
- Git
- Crawler
- Jinja
- Bootstrap
- Pusher
- BeautifulSoup
- Debugging
- Python data types
- DB data types
- JSON
- Functions
- Variables
- If-else
- While loop
- For loop
- jQuery plugins
- Python modules
- API

Web Service Development

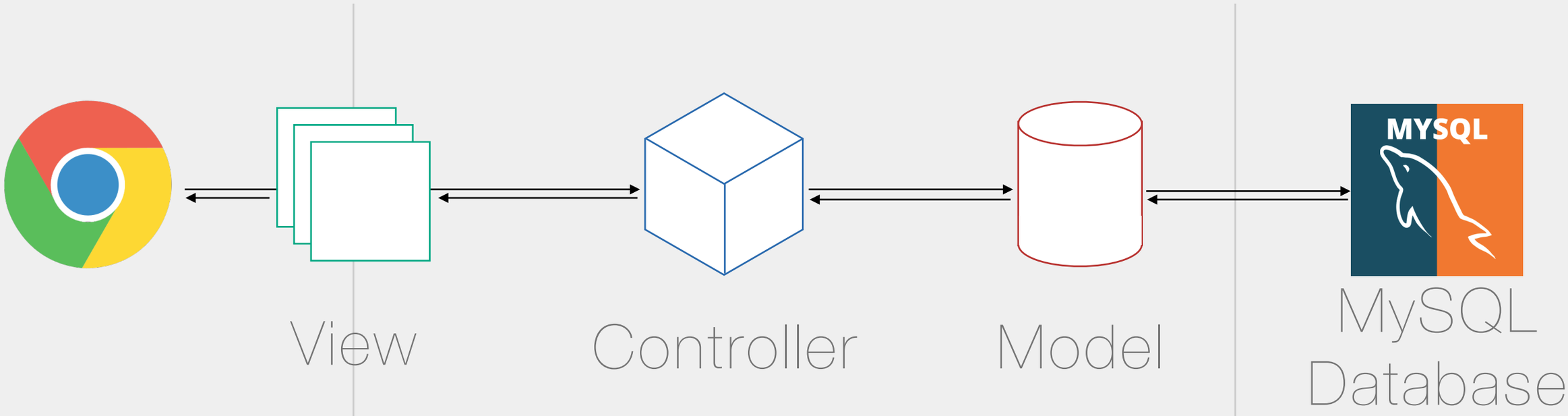
- Client - Server
- Request
- HTML, CSS, JavaScript(jQuery)
- Python
- HTML Form
- Session
- AJAX
- JSON
- MVC
- Database - ORM

Web Service Structure

Client

Web Server

DB Server



Client-Server

Client-Server

in Web Services

Web Client

Client Computer + Client Program

Web Server

Server Computer + Server Program

Client-Server

in Web Services

Web Client

Client Computer + Client Program

Web Server

Server Computer + Server Program

Client-Server

in Web Services

Web Client

Web Server

Request

Response

Client Computer + Client Program

Server Computer + Server Program

Client-Server

in Web Services

- Client : 서버로 요청(request) 전송

- Server : 클라이언트의 요청을 받아서 처리한 후 응답(response) 돌려줌

Client-Server

Client

- Web Browser (Chrome)

Server

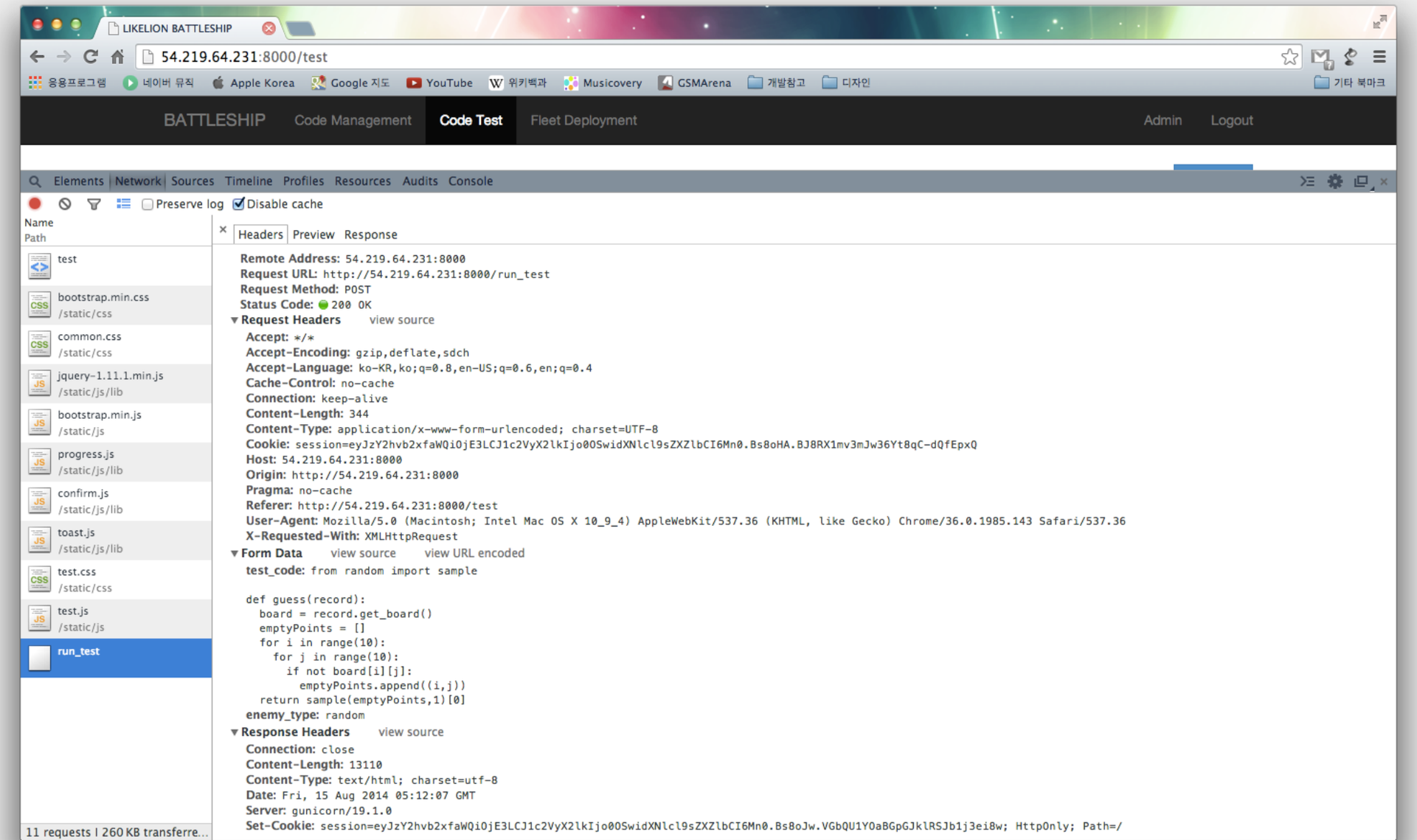
- Web Server (Flask)

Request

HTTP Request

HTTP Request : URL + Data

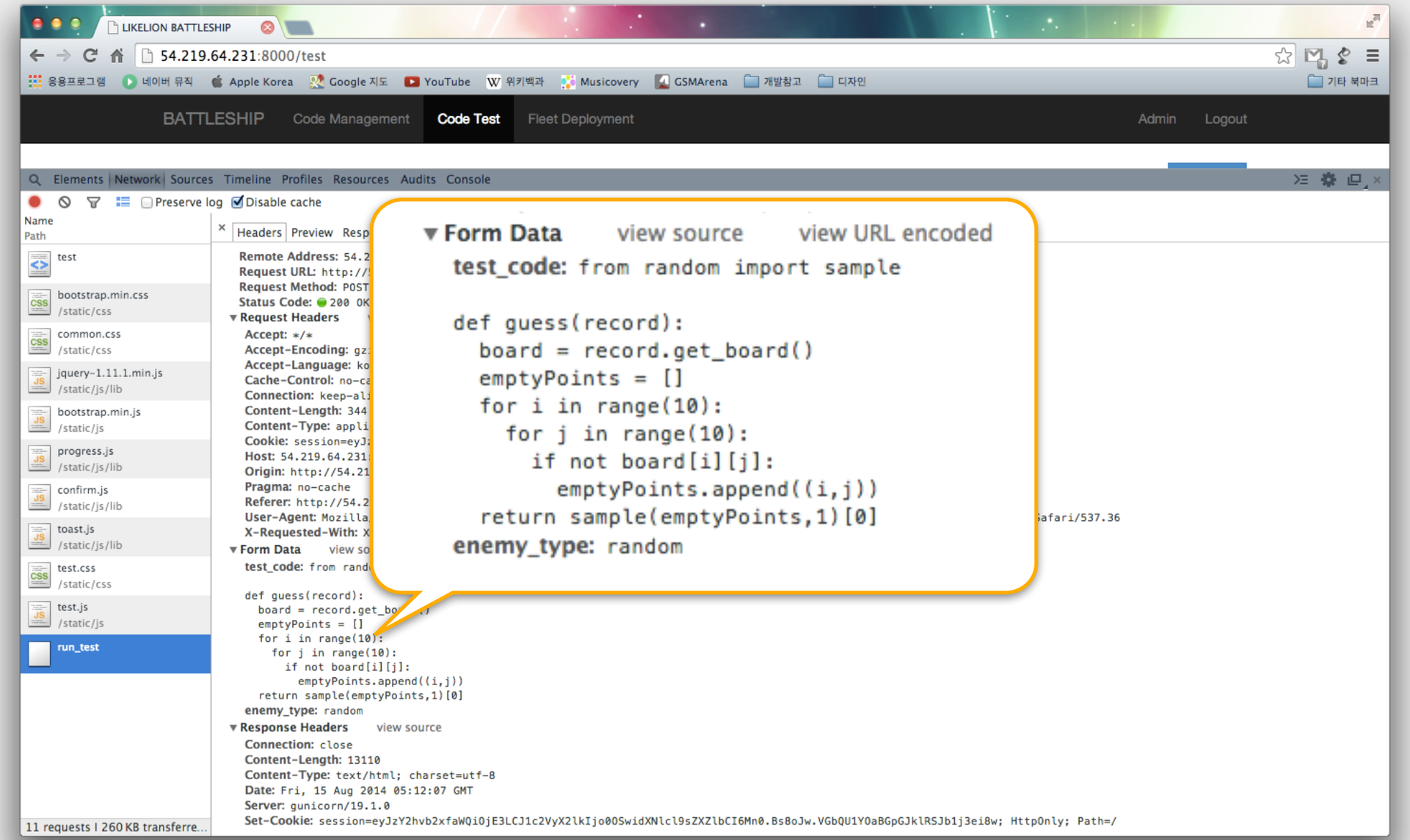
Request


LIKELION BATTLESHIP
54.219.64.231:8000/test
응용프로그램
네이버 뮤직
Apple Korea
Google 지도
YouTube
위키백과
Musicovery
GSMArena
개발참고
디자인
기타 북마크
BATTLESHIP
Code Management
Code Test
Fleet Deployment
Admin
Logout
Elements
Network
Sources
Timeline
Profiles
Resources
Audits
Console
Preserve log
Disable cache
Name
Path
test
bootstrap.min.css
/static/css
common.css
/static/css
jquery-1.11.1.min.js
/static/js/lib
bootstrap.min.js
/static/js
progress.js
/static/js/lib
confirm.js
/static/js/lib
toast.js
/static/js/lib
test.css
/static/css
test.js
/static/js
run_test
11 requests | 260 KB transferre...
Headers
Preview
Response
Remote Address: 54.219.64.231:8000
Request URL: http://54.219.64.231:8000/run_test
Request Method: POST
Status Code: 200 OK
Request Headers
view source
Accept: */*
Accept-Encoding: gzip,deflate,sdch
Accept-Language: ko-KR,ko;q=0.8,en-US;q=0.6,en;q=0.4
Cache-Control: no-cache
Connection: keep-alive
Content-Length: 344
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Cookie: session=eyJzY2hvb2xfaWQiOjE3LCJ1c2VyX2lkIjo0OSwidXNlcl9sZXZlbCI6Mn0.Bs8oHA.BJ8RX1mv3mJw36Yt8qC-dQfEpxQ
Host: 54.219.64.231:8000
Origin: http://54.219.64.231:8000
Pragma: no-cache
Referer: http://54.219.64.231:8000/test
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/36.0.1985.143 Safari/537.36
X-Requested-With: XMLHttpRequest
Form Data
view source
view URL encoded
test_code: from random import sample
def guess(record):
board = record.get_board()
emptyPoints = []
for i in range(10):
for j in range(10):
if not board[i][j]:
emptyPoints.append((i,j))
return sample(emptyPoints,1)[0]
enemy_type: random
Response Headers
view source
Connection: close
Content-Length: 13110
Content-Type: text/html; charset=utf-8
Date: Fri, 15 Aug 2014 05:12:07 GMT
Server: gunicorn/19.1.0
Set-Cookie: session=eyJzY2hvb2xfaWQiOjE3LCJ1c2VyX2lkIjo0OSwidXNlcl9sZXZlbCI6Mn0.Bs8oJw.VGbQU1YOaBGpGJklRSJb1j3ei8w; HttpOnly; Path=/

Request


LIKELION BATTLESHIP
54.219.64.231:8000/test
BATTLESHIP
Code Management
Code Test
Fleet Deployment
Admin
Logout
Request URL: http://54.219.64.231:8000/run_test
Elements
Network
Sources
Timeline
Profiles
Preserve log
Disable cache
Name
Path
test
bootstrap.min.css
/static/css
common.css
/static/css
jquery-1.11.1.min.js
/static/js/lib
bootstrap.min.js
/static/js
progress.js
/static/js/lib
confirm.js
/static/js/lib
toast.js
/static/js/lib
test.css
/static/css
test.js
/static/js
run_test
11 requests | 260 KB transferre...
Headers
Preview
Response
Remote Address: 54.219.64.231:8000
Request URL: http://54.219.64.231:8000/run_test
Request Method: POST
Status Code: 200 OK
Request Headers
view source
Accept: */*
Accept-Encoding: gzip,deflate,sdch
Accept-Language: ko-KR,ko;q=0.8,en-US;q=0.6,en;q=0.4
Cache-Control: no-cache
Connection: keep-alive
Content-Length: 344
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Cookie: session=eyJzY2hvb2xfaWQiOjE3LCJ1c2VyX2lkIjo0OSwidXNlcl9sZXZlbCI6Mn0.Bs8oHA.BJ8RX1mv3mJw36Yt8qC-dQfEpxQ
Host: 54.219.64.231:8000
Origin: http://54.219.64.231:8000
Pragma: no-cache
Referer: http://54.219.64.231:8000/test
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/36.0.1985.143 Safari/537.36
X-Requested-With: XMLHttpRequest
Form Data
view source
view URL encoded
test_code: from random import sample
def guess(record):
board = record.get_board()
emptyPoints = []
for i in range(10):
for j in range(10):
if not board[i][j]:
emptyPoints.append((i,j))
return sample(emptyPoints,1)[0]
enemy_type: random
Response Headers
view source
Connection: close
Content-Length: 13110
Content-Type: text/html; charset=utf-8
Date: Fri, 15 Aug 2014 05:12:07 GMT
Server: gunicorn/19.1.0
Set-Cookie: session=eyJzY2hvb2xfaWQiOjE3LCJ1c2VyX2lkIjo0OSwidXNlcl9sZXZlbCI6Mn0.Bs8oJw.VGbQU1YOaBGpGJklRSJb1j3ei8w; HttpOnly; Path=/

Request


LIKELION BATTLESHIP
54.219.64.231:8000/test
BATTLESHIP
Code Management
Code Test
Fleet Deployment
Admin
Logout
Elements
Network
Sources
Timeline
Profiles
Resources
Audits
Console
Preserve log
Disable cache
Name
Path
test
bootstrap.min.css
/static/css
common.css
/static/css
jquery-1.11.1.min.js
/static/js/lib
bootstrap.min.js
/static/js
progress.js
/static/js/lib
confirm.js
/static/js/lib
toast.js
/static/js/lib
test.css
/static/css
test.js
/static/js
run_test
11 requests | 260 KB transferre...
Headers
Preview
Form Data
view source
view URL encoded
test_code: from random import sample
def guess(record):
board = record.get_board()
emptyPoints = []
for i in range(10):
for j in range(10):
if not board[i][j]:
emptyPoints.append((i,j))
return sample(emptyPoints,1)[0]
enemy_type: random
Request Headers
Form Data
Response Headers
view source
Connection: close
Content-Length: 13110
Content-Type: text/html; charset=utf-8
Date: Fri, 15 Aug 2014 05:12:07 GMT
Server: gunicorn/19.1.0

HTTP Request : URL + Data

Request > URL

URL : 요청을 보낼 위치

Request > URL

http://54.219.64.231:8000/test

서버 주소

Request > URL

http://54.219.64.231:8000/test

서버 내 위치

Request > URL

```python
# controllers/test.py

@app.route('/test')
def test():
    return render_template('test.html')
```

Request > URL

```
# controller
@app.route('/test')
def test():
    return render_template('test.html')
```

이 위치로 요청을 받으면

Request > URL

```python
# controllers/test.py

@app.route('/test')
def test():
    return render_template('test.html')
```

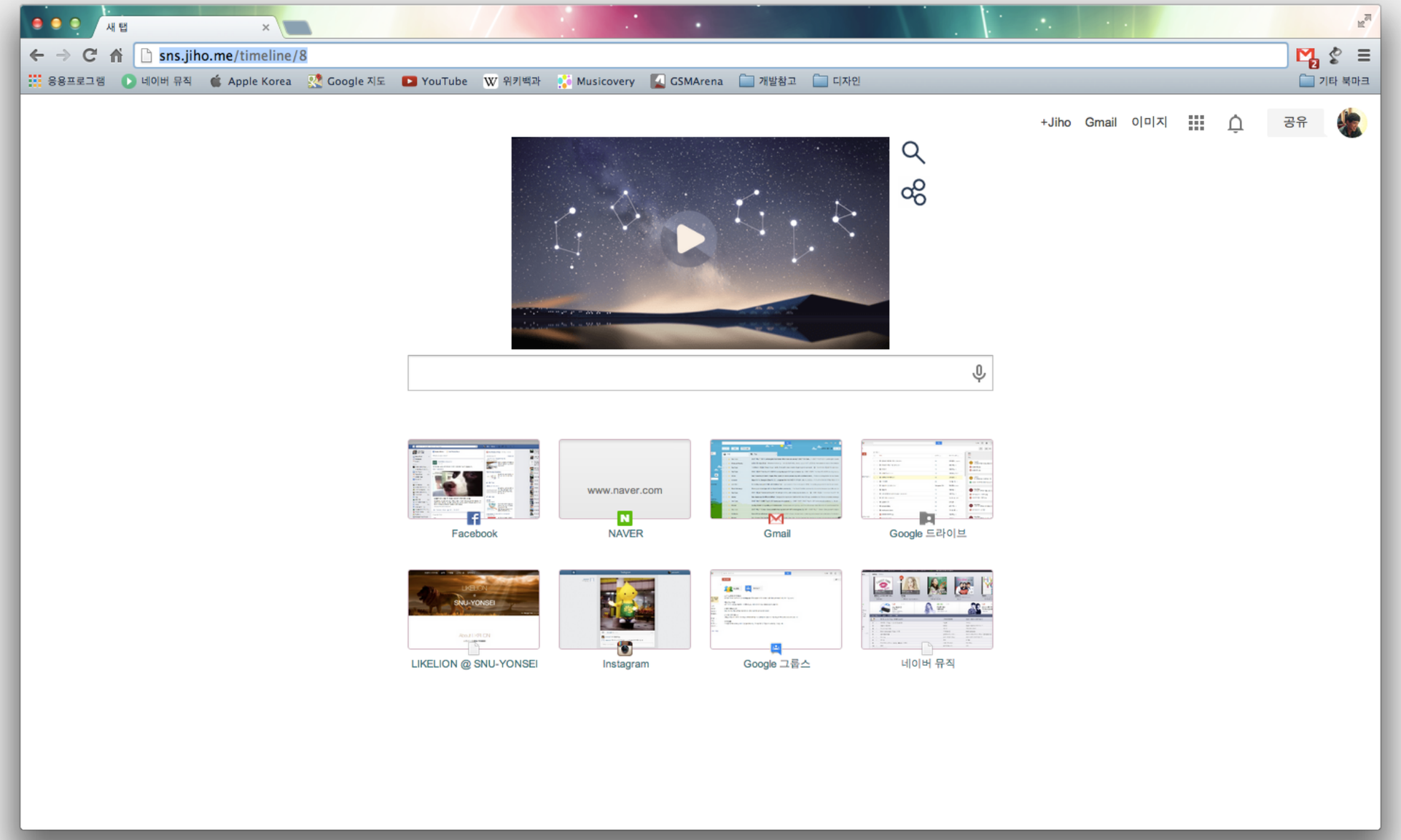
이 함수를 호출

Request > URL

```python
# controllers/test.py

@app.route('/test')
def test():
    return render_template('test.html')
```

return 값 : Response

Request Handling



Request Handling

```python
# sns.jiho.me/timeline/8
@app.route('/', defaults={'wall_id':0})
@app.route('/timeline/<int:wall_id>')
def timeline(wall_id) :
    if not is_login():
        return redirect(url_for('login'))
    if wall_id == 0:
        wall_id = session['user_id']

    session['wall_id'] = wall_id
    session['wall_username'] = get_user(wall_id).username

    context = {
        'posts':get_wall_posts(wall_id,0),
        'BS':BS
    }
    postList = render_template('post_list.html',context=context)
    context = {
        'post_list':postList
    }
    return render_template('timeline.html', context=context)
```

Request Handling

```python
# sns.jiho.me/timeline/8
@app.route('/', defaults={'wall_id':0})
@app.route('/timeline/<int:wall_id>')
def timeline(wall_id) :
    if not is_login():
        return redirect(url_for('login'))
    if wall_id == 0:
        wall_id = session['user_id']

    session['wall_id'] = wall_id
    session['wall_username'] = get_user(wall_id).username

    context = {
        'posts':get_wall_posts(wall_id,0),
        'BS':BS
    }
    postList = render_template('post_list.html',context=context)
    context = {
        'post_list':postList
    }
    return render_template('timeline.html', context=context)
```

클라이언트의 요청을 받아서

Request Handling

처리한 후

```python
# sns.jiho.me/timeline/8
@app.route('/', defaults={'wall_id':0})
@app.route('/timeline/<int:wall_id>')
def timeline(wall_id) :
    if not is_login():
        return redirect(url_for('login'))
    if wall_id == 0:
        wall_id = session['user_id']

    session['wall_id'] = wall_id
    session['wall_username'] = get_user(wall_id).username

    context = {
        'posts':get_wall_posts(wall_id,0),
        'BS':BS
    }
    postList = render_template('post_list.html',context=context)
    context = {
        'post_list':postList
    }
    return render_template('timeline.html', context=context)
```

Request Handling

```python
# sns.jiho.me/timeline/8
@app.route('/', defaults={'wall_id':0})
@app.route('/timeline/<int:wall_id>')
def timeline(wall_id) :
    if not is_login():
        return redirect(url_for('login'))
    if wall_id == 0:
        wall_id = session['user_id']

    session['wall_id'] = wall_id
    session['wall_username'] = get_user(wall_id).username

    context = {
        'posts':get_wall_posts(wall_id,0),
        'BS':BS
    }
    postList = render_template('post_list.html',context=context)
    context = {
        'post_list':postList
    }
    return render_template('timeline.html', context=context)
```

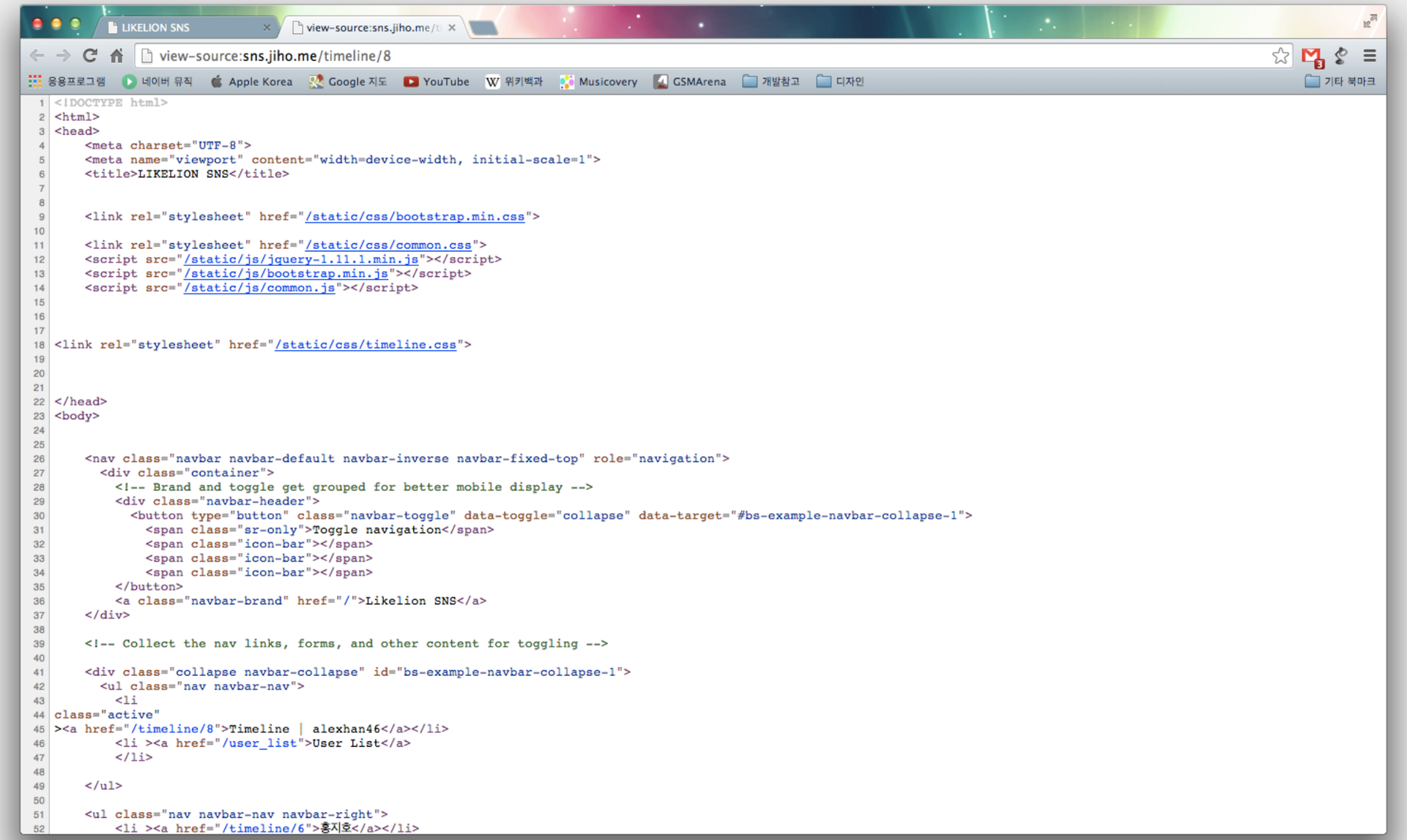
응답을 돌려줌

Request Handling

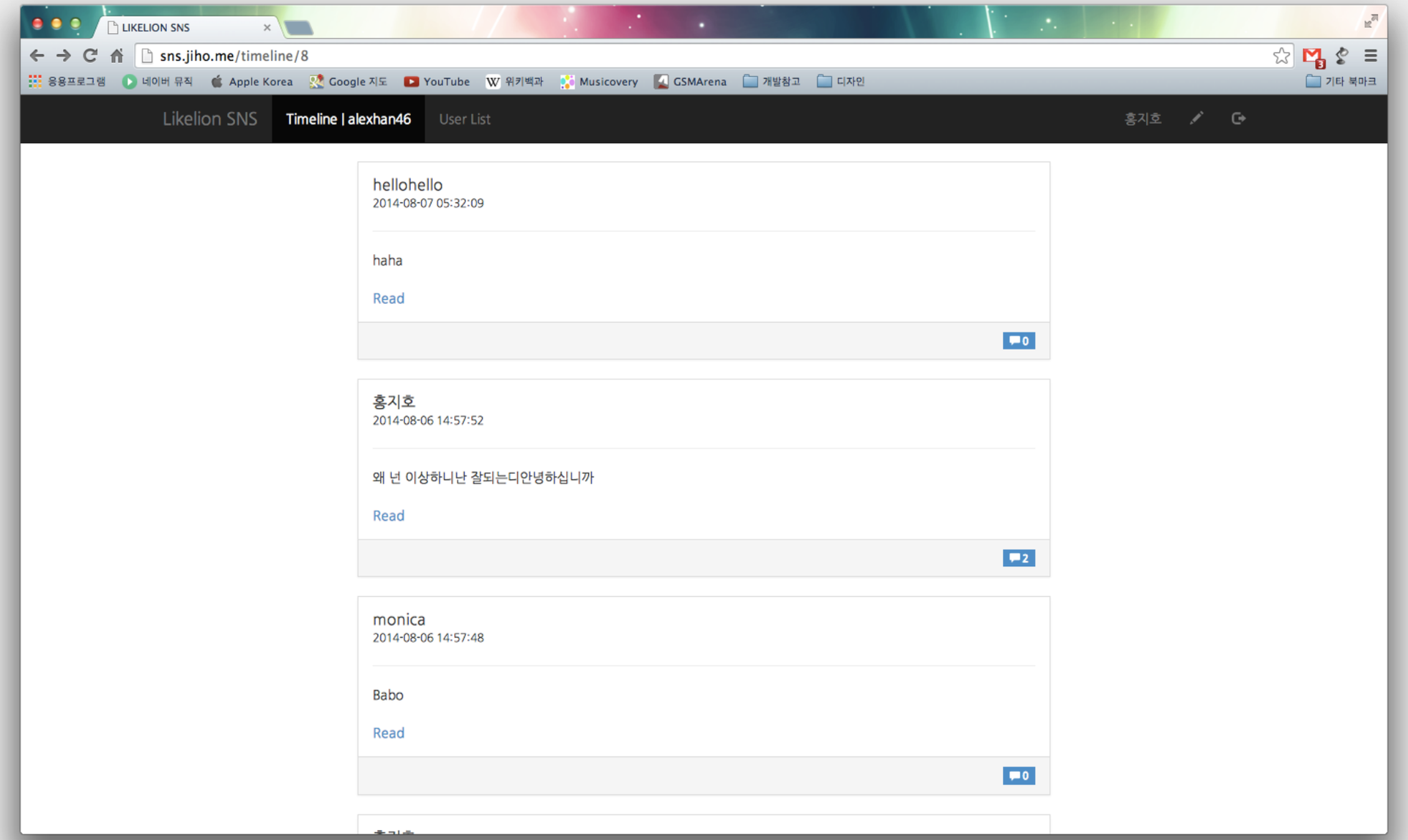
```python
# sns.jiho.me/timeline/8
@app.route('/', defaults={'wall_id':0})
@app.route('/timeline/<int:wall_id>')
def timeline(wall_id) :
    if not is_login():
        return redirect(url_for('login'))
    if wall_id == 0:
        wall_id = session['user_id']

    session['wall_id'] = wall_id
    session['wall_username'] = get_user(wall_id).username

    context = {
        'posts':get_wall_posts(wall_id,0),
        'BS':BS
    }
    postList = render_template('post_list.html',context=context)
    context = {
        'post_list':postList
    }
    return render_template('timeline.html', context=context)
```

Response

Request Handling



Request Handling



Client-Side

Front-end

HTML, CSS, JavaScript

Client-Side

HTML, CSS, JavaScript

Client-Side

```
    >
17      <input type="password" class="
        form-control" id="input-password
        -confirm" placeholder="Confirm
        password"
18      name="password-confirm" required
        >
19  </div>
20
21  <div class="form-group">
22      <label for="input-username">
        Username</label>
23      <input type="text" class="form-
        control" id="input-username"
24      name="username" placeholder="
        Username" required>
25  </div>
26
27  <div class="form-group">
28      <label class="radio-inline">
29        <input type="radio" name="
          gender" value="F" checked>
          Female
30    </label>
31    <label class="radio-inline">
32        <input type="radio" name="
          gender" value="M"> Male
33    </label>
34  </div>
35  <div class="form-group">
36      <label for="input-mobile">Phone<
        /label>
37      <input type="text" class="form-
        control" id="input-mobile"
38      name="mobile" placeholder="Phone
        " required>
39  </div>
```

```
1   @import url(http://fonts.googleapis.
        com/earlyaccess/nanumgothic.css)
        ;
2
3
4   body {
5       padding-top:70px;
6       font-family:"Nanum Gothic";
7
8   }
9
10
11  * {
12      border-radius: 0px !important;
13  }
```

```
17  });
18
19  function show_toast(message, level,
        style){
20
21      level = level || 'default';
22
23      // set custom styles
24      if(style){
25          for(s in defaultStyle){
26              if(!style[s]){
27                  style[s] =
                        defaultStyle[s];
28              }
29          }
30      }else{
31          style = defaultStyle;
32      }
33
34      style.backgroundColor = color[
            level];
35
36
37
38      if($('div#toast').length>0){
39          setTimeout('show_toast("'+
                message+'","'+level+'", '
                +JSON.stringify(style)+')
                ',300);
40      }else{
41          var toast = '<div id="toast"
                style="text-align:center;
            position:fixed;bottom:20%;wid
            th:100%;padding:0px;maring:0p
            x;display:none;z-index:9999;"
                ><span>'+message+'</span>
                </div>';
42
```

HTML, CSS, JavaScript

Client-Side

HTML, CSS, JavaScript

Client-Side

구조

HTML, CSS, JavaScript

Client-Side

구조 표현

HTML, CSS, JavaScript

Client-Side

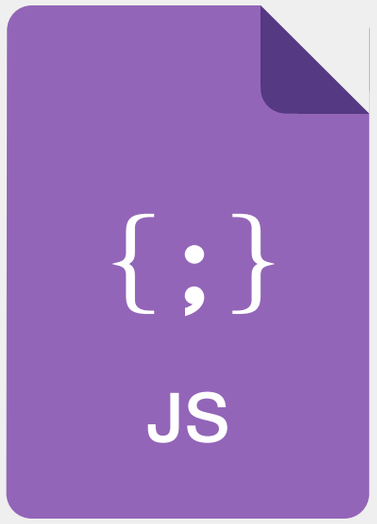
구조 표현 동작

HTML, CSS, JavaScript

Client-Side

구조

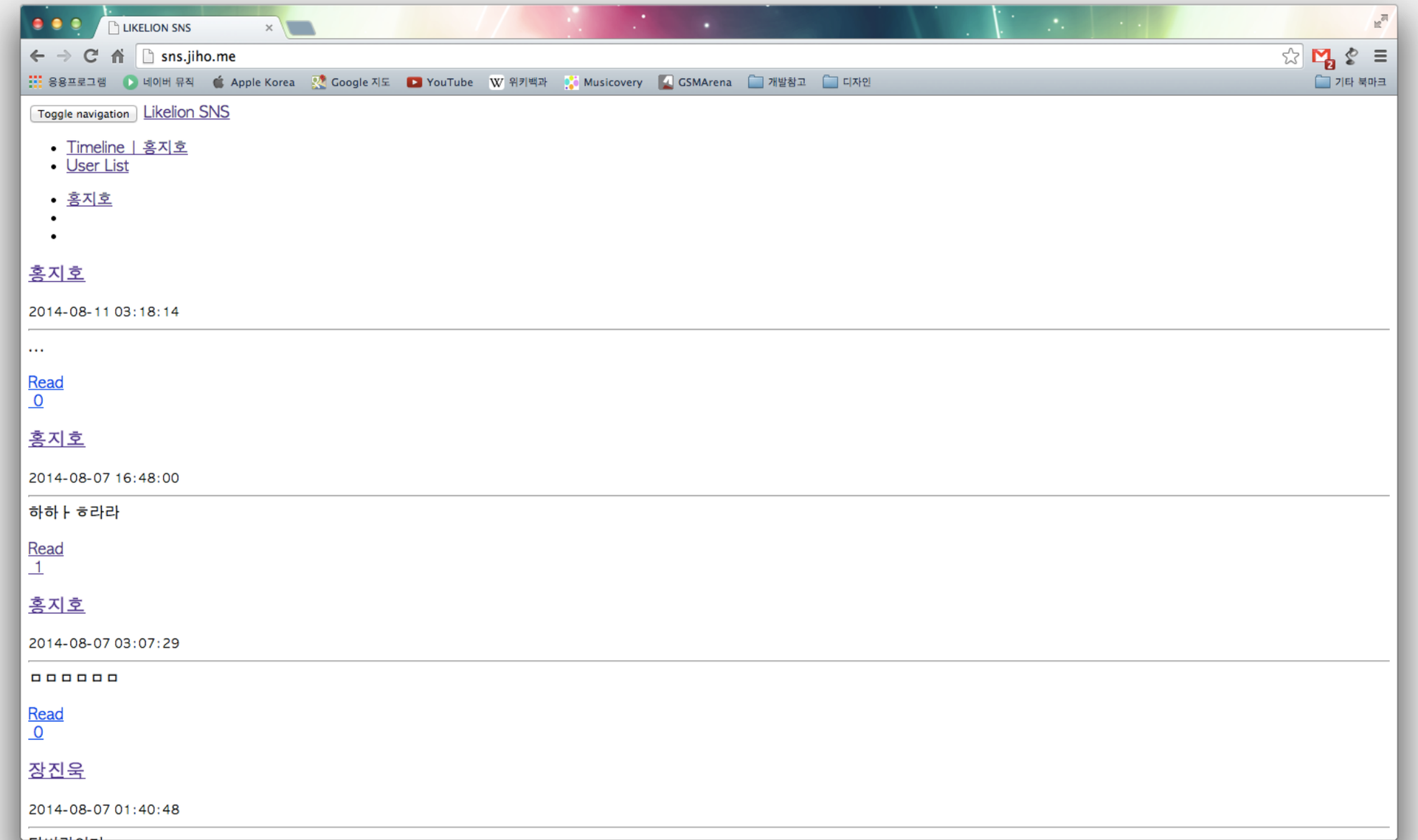
표현



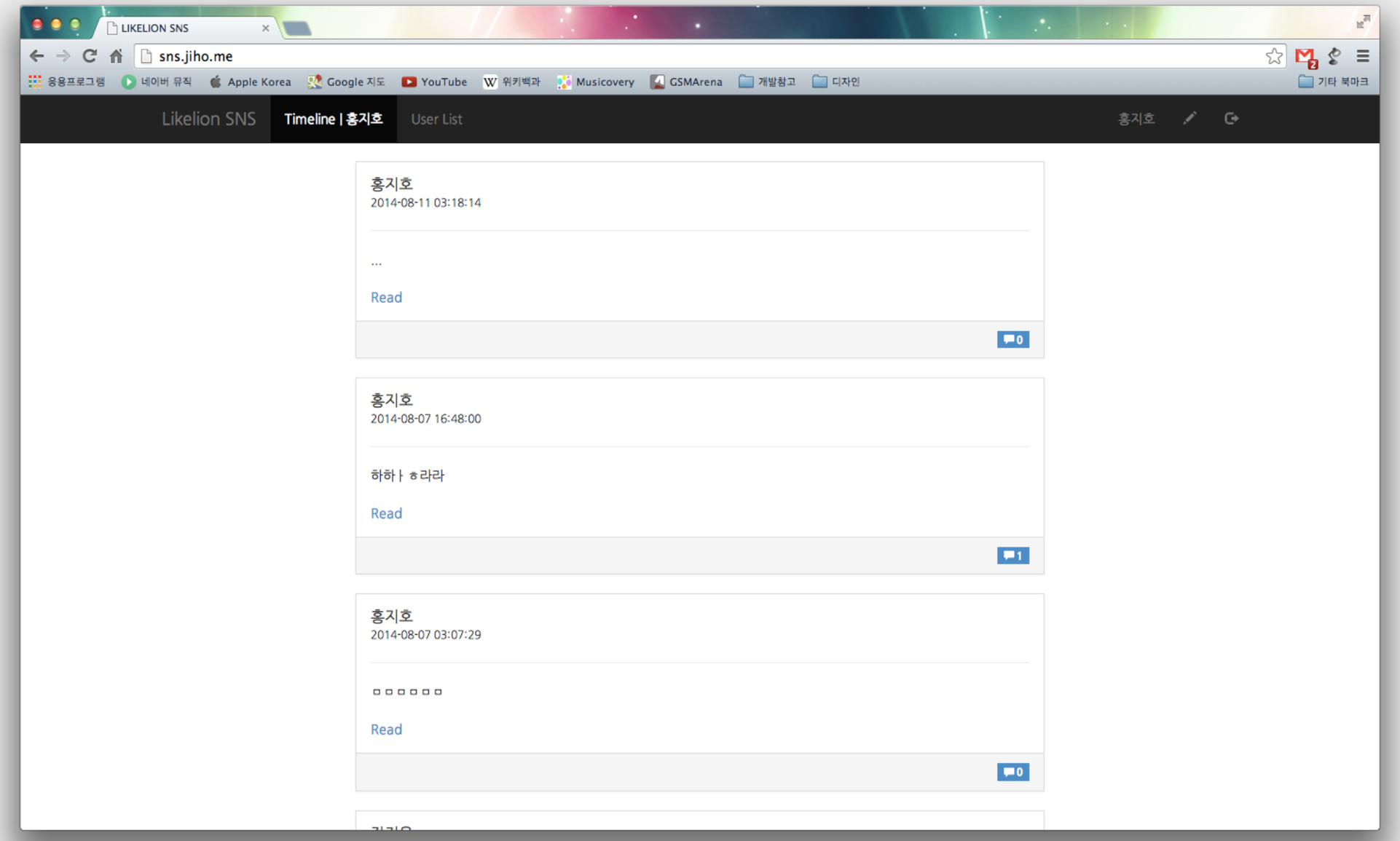
동작

jQuery :
JS 라이브러리

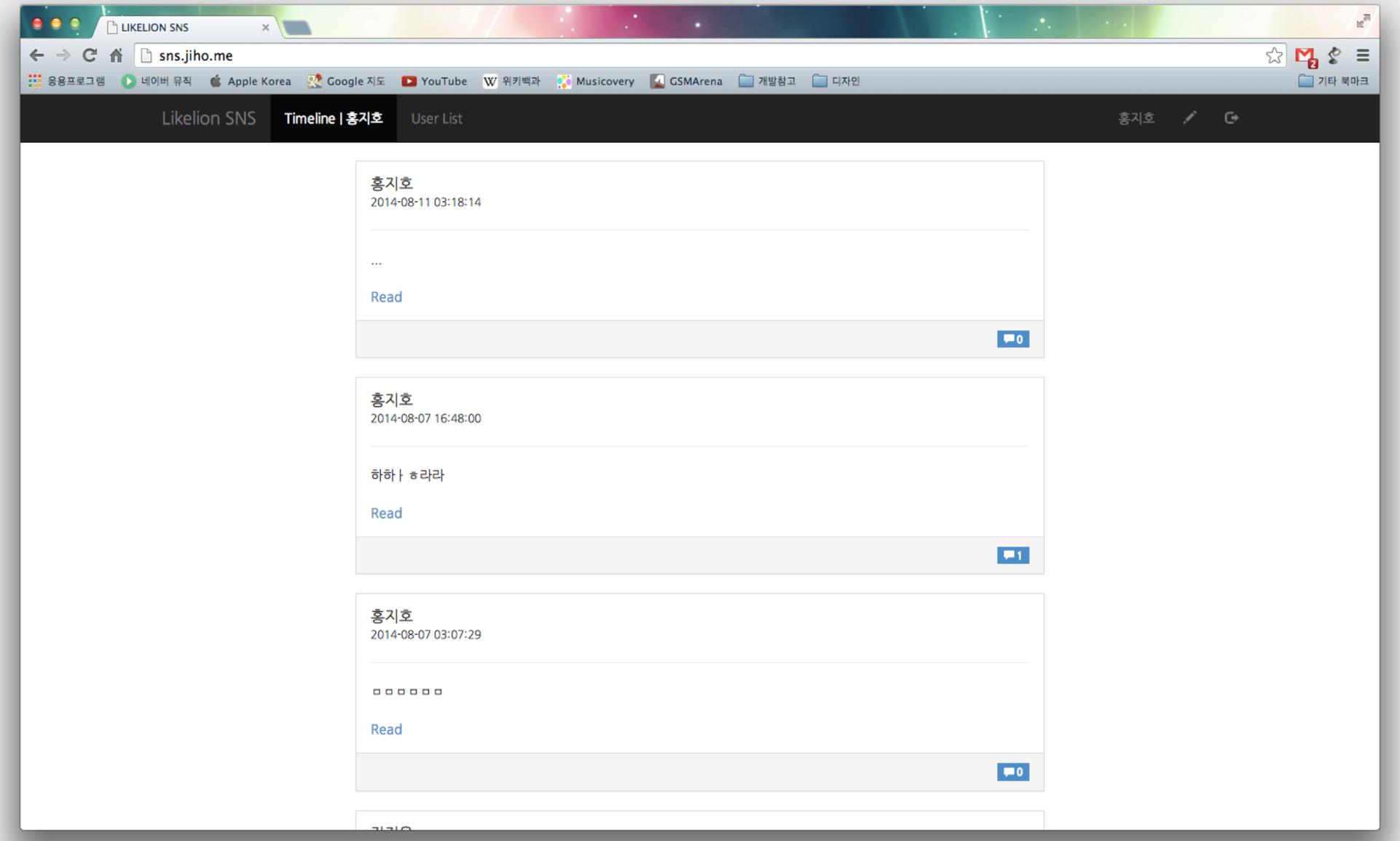
HTML


LIKELION SNS
sns.jiho.me
응용프로그램
네이버 뮤직
Apple Korea
Google 지도
YouTube
위키백과
Musicovery
GSMArena
개발참고
디자인
기타 북마크
Toggle navigation
Likelion SNS
Timeline | 홍지호
User List
홍지호
홍지호
2014-08-11 03:18:14
...
Read
0
홍지호
2014-08-07 16:48:00
하하ㅏ ㅎ라라
Read
1
홍지호
2014-08-07 03:07:29
ㅁㅁㅁㅁㅁㅁ
Read
0
장진욱
2014-08-07 01:40:48

HTML+CSS


LIKELION SNS
sns.jiho.me
응용프로그램
네이버 뮤직
Apple Korea
Google 지도
YouTube
위키백과
Musicovery
GSMArena
개발참고
디자인
기타 북마크
Likelion SNS
Timeline | 홍지호
User List
홍지호
홍지호
2014-08-11 03:18:14
...
Read
0
홍지호
2014-08-07 16:48:00
하하ㅏ ㅎ라라
Read
1
홍지호
2014-08-07 03:07:29
ㅁㅁㅁㅁㅁㅁ
Read
0

HTML + CSS + JavaScript



Client-Server

Client

- Web Browser
- HTML, CSS, JavaScript

Server

- Web Server

HTML

웹 페이지의 구조

http://www.w3schools.com/

HTML

웹 페이지의 구조

```
<!DOCTYPE html>
<html>
    <head>

    </head>
    <body>

    </body>
</html>
```

HTML

웹 페이지의 구조

```
<!DOCTYPE html>
<html>  이 문서는 HTML 문서다
    <head>

    </head>
    <body>


    </body>
</html>
```

HTML

웹 페이지의 구조

```
<!DOCTYPE html>
<html>
    <head>

    </head>
    <body>


    </body>
</html>
```

HTML 문서의 시작과 끝

HTML

웹 페이지의 구조

```
<!DOCTYPE html>
<html>
    <head>

    </head>
    <body>

    </body>
</html>
```

문서에 대한 정보가 들어있는 부분
브라우저에는 표시되지 않음

HTML

웹 페이지의 구조

```
<!DOCTYPE html>
<html>
    <head>
    </head>
    <body>


    </body>
</html>
```

브라우저에 보이는
실제 내용이 들어가는 부분

HTML

웹 페이지의 구조

- 태그
- 속성
- 내용

HTML

웹 페이지의 구조

```
<a id="link-daum" class="link active" href="http://www.naver.com">다음</a>
```

HTML

웹 페이지의 구조

```
<a id="link-daum" class="link active" href="http://www.naver.com">다음</a>
```

시작 태그

HTML

웹 페이지의 구조

```
<a id="link-daum" class="link active" href="http://www.naver.com">다음</a>
```

종료 태그

HTML

웹 페이지의 구조

```
<a id="link-daum" class="link active" href="http://www.naver.com">다음</a>
```

태그 이름

HTML

웹 페이지의 구조

```
<a id="link-daum" class="link active" href="http://www.naver.com">다음</a>
```

속성명

HTML

웹 페이지의 구조

```
<a id="link-daum" class="link active" href="http://www.naver.com">다음</a>
```

속성값

HTML

웹 페이지의 구조

속성값은 큰 따옴표로 묶기

```
<a id="link-daum" class="link active" href="http://www.naver.com">다음</a>
```

HTML

웹 페이지의 구조

```
<a id="link-daum" class="link active" href="http://www.naver.com">다음</a>
```

여러 개의 클래스는 공백으로 구분

HTML

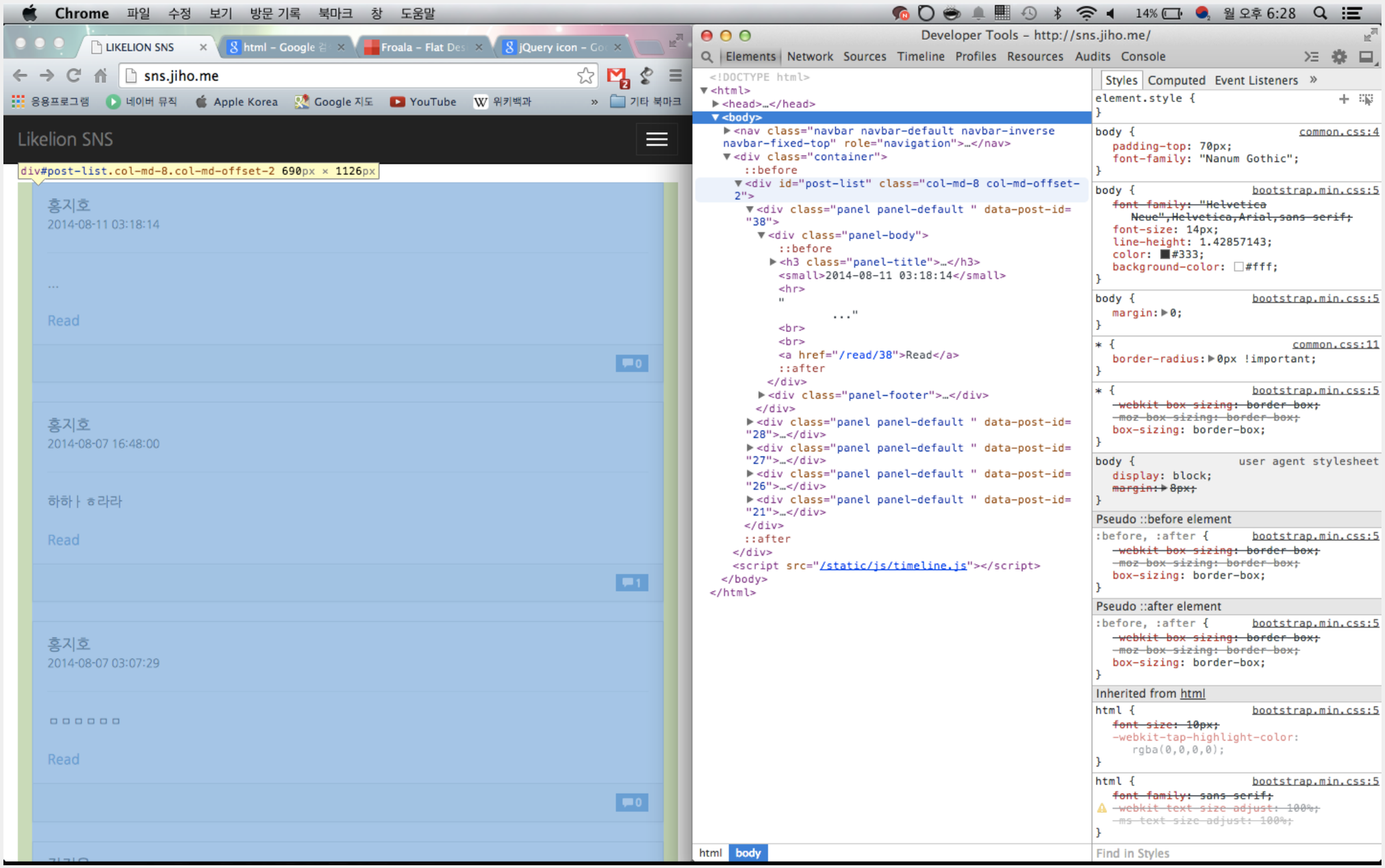
웹 페이지의 구조

```
<a id="link-daum" class="link active" href="http://www.naver.com">다음</a>
```

내용

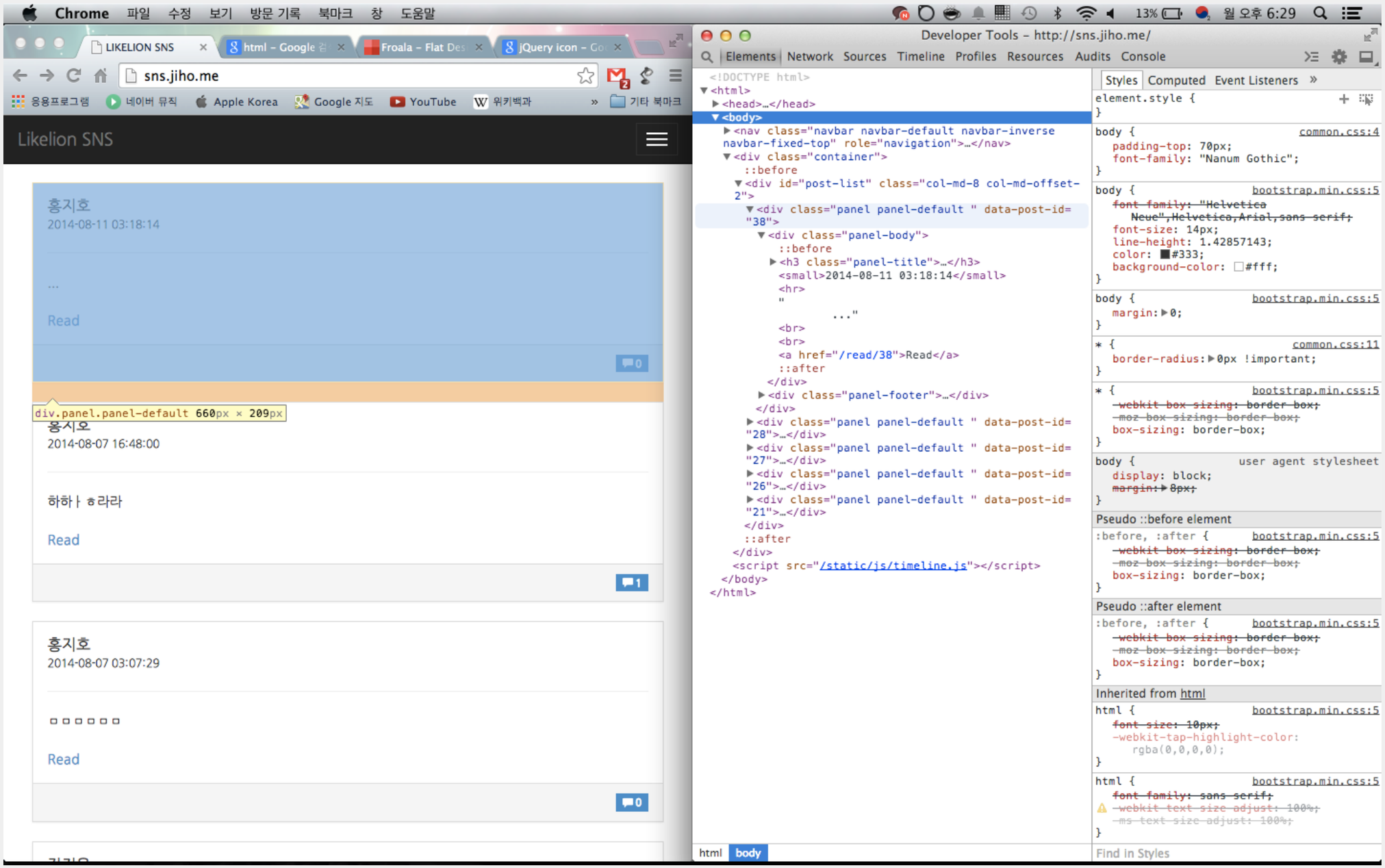
HTML

웹 페이지의 구조



HTML

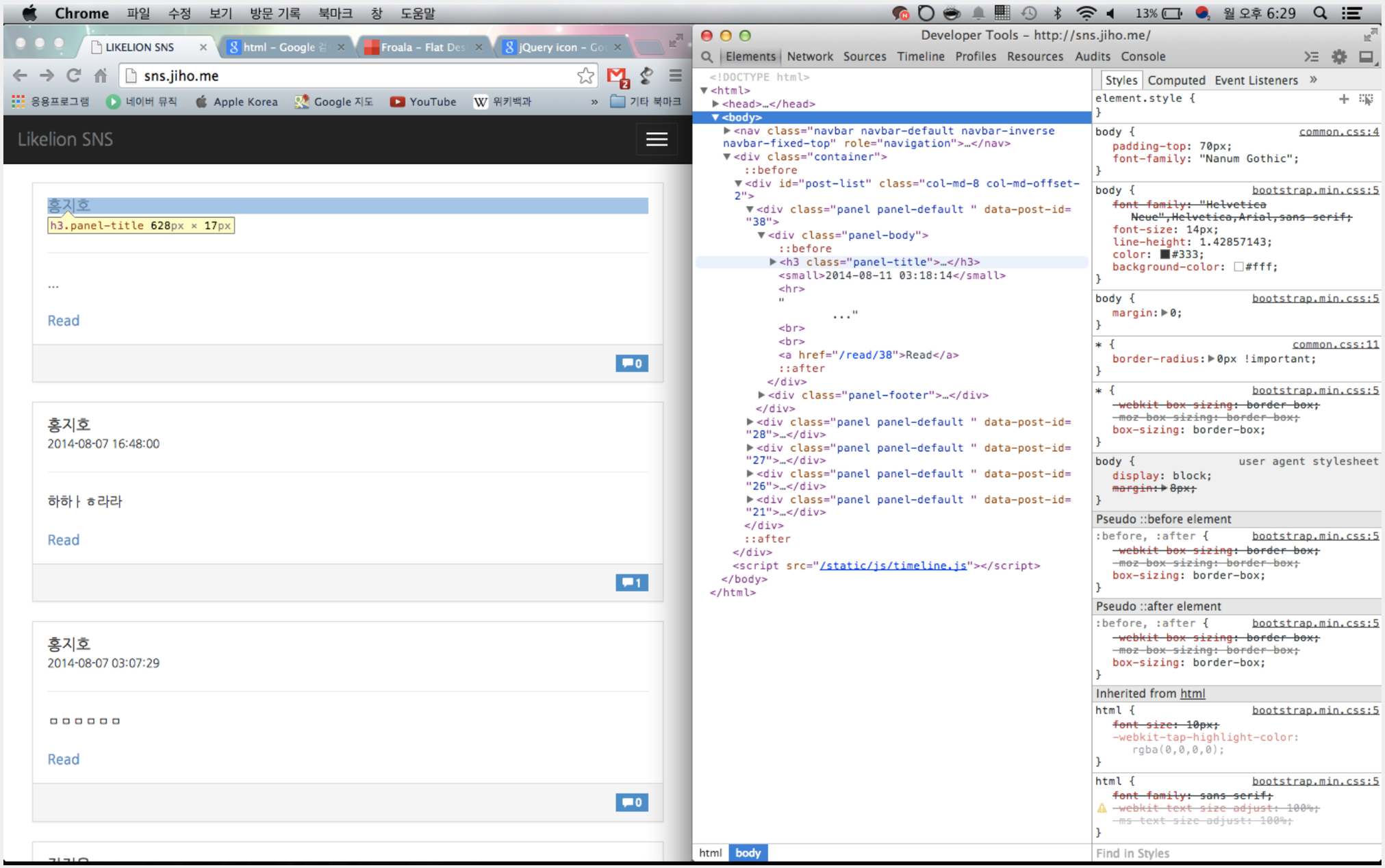
웹 페이지의 구조



Likelion

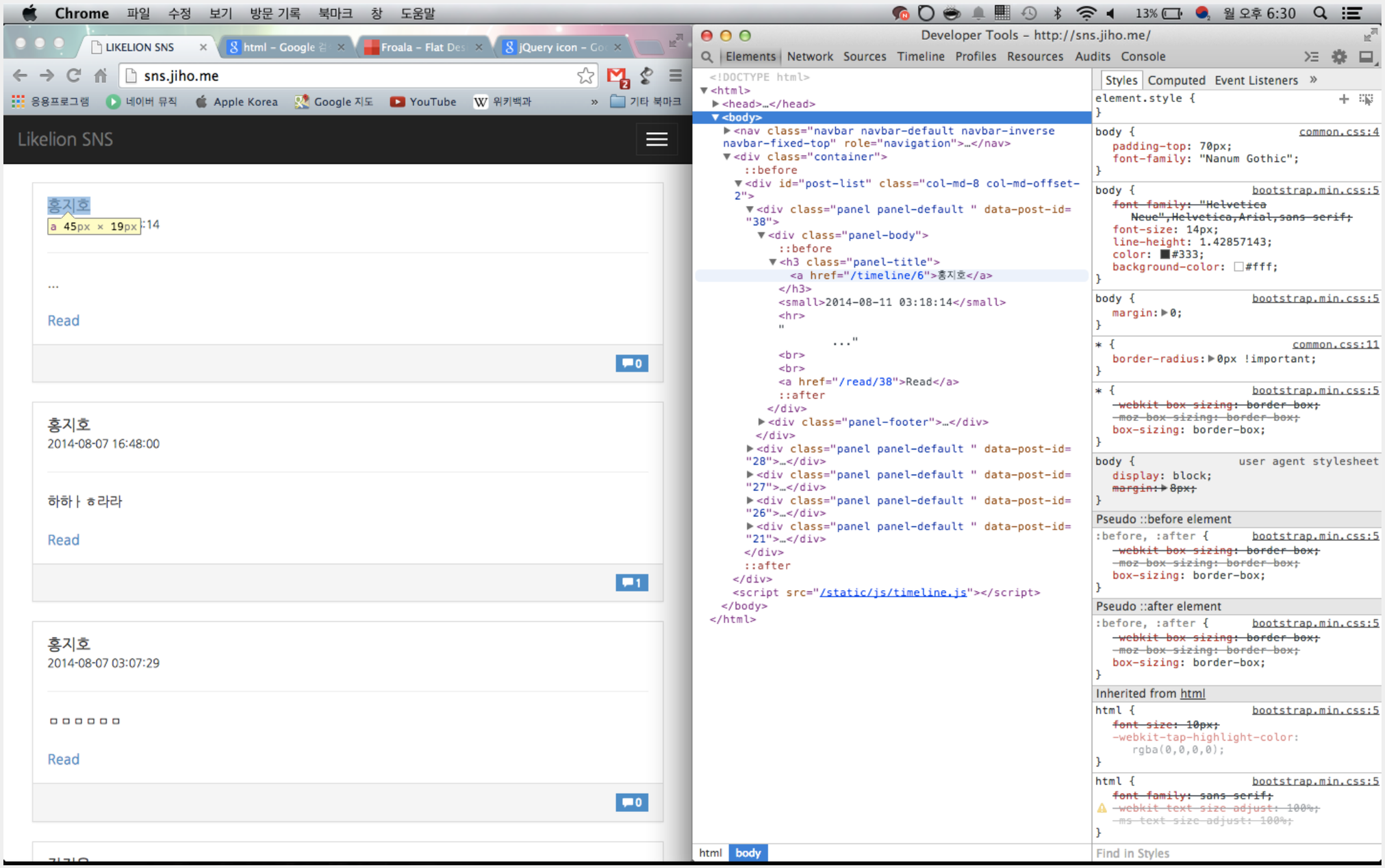
HTML

웹 페이지의 구조



HTML

웹 페이지의 구조



Client-Server

Client

- Web Browser
- HTML, CSS, JavaScript

Server

- Web Server

CSS

웹 페이지의 표현

http://www.w3schools.com/

CSS

웹 페이지의 표현

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Likelion SNS</title>

    <script src="//code.jquery.com/jquery-1.11.0.min.js"></script>

    <!-- Latest compiled and minified CSS -->
    <link rel="stylesheet" href="//maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap
    .min.css">

    <!-- Latest compiled and minified JavaScript -->
    <script src="//maxcdn.bootstrapcdn.com/bootstrap/3.2.0/js/bootstrap.min.js"></script
        >
    <link rel="stylesheet" href="{{ url_for('static',filename='css/common.css') }}">

</head>
<body>

    <nav class="navbar navbar-default" role="navigation">
        <div class="container">
            <!-- Brand and toggle get grouped for better mobile display -->
            <div class="navbar-header">
```

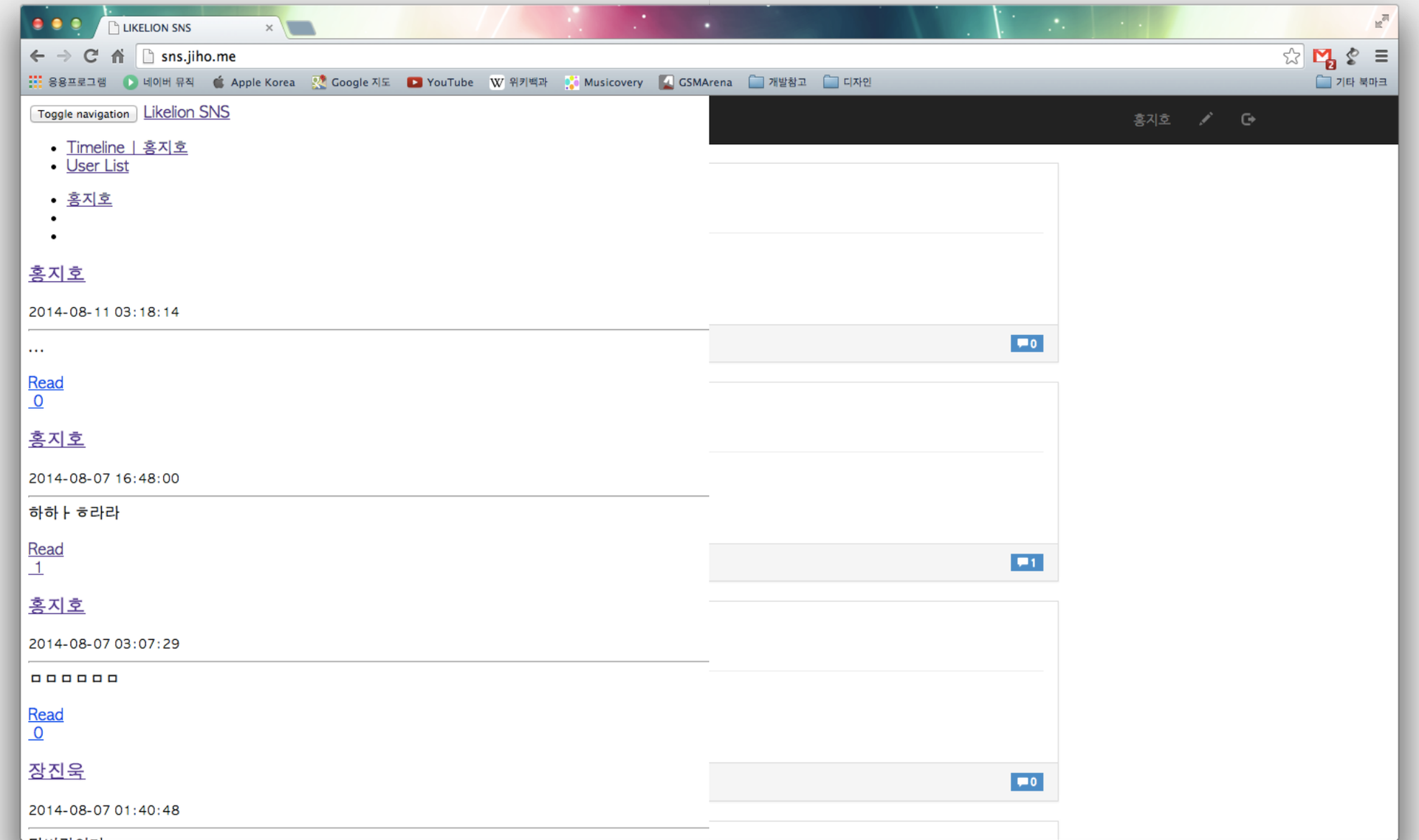
CSS

웹 페이지의 표현

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Likelion SNS</title>

    <script src="//code.jquery.com/jquery-1.11.0.min.js"></script>

    <!-- Latest compiled and minified CSS -->
    <link rel="stylesheet" href="//maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap
    .min.css">

    <!-- Latest compiled and minified JavaScript -->
    <script src="//maxcdn.bootstrapcdn.com/bootstrap/3.2.0/js/bootstrap.min.js"></script
        >
    <link rel="stylesheet" href="{{ url_for('static',filename='css/common.css') }}">

</head>
<body>

    <nav class="navbar navbar-default" role="navigation">
        <div class="container">
            <!-- Brand and toggle get grouped for better mobile display -->
            <div class="navbar-header">
```

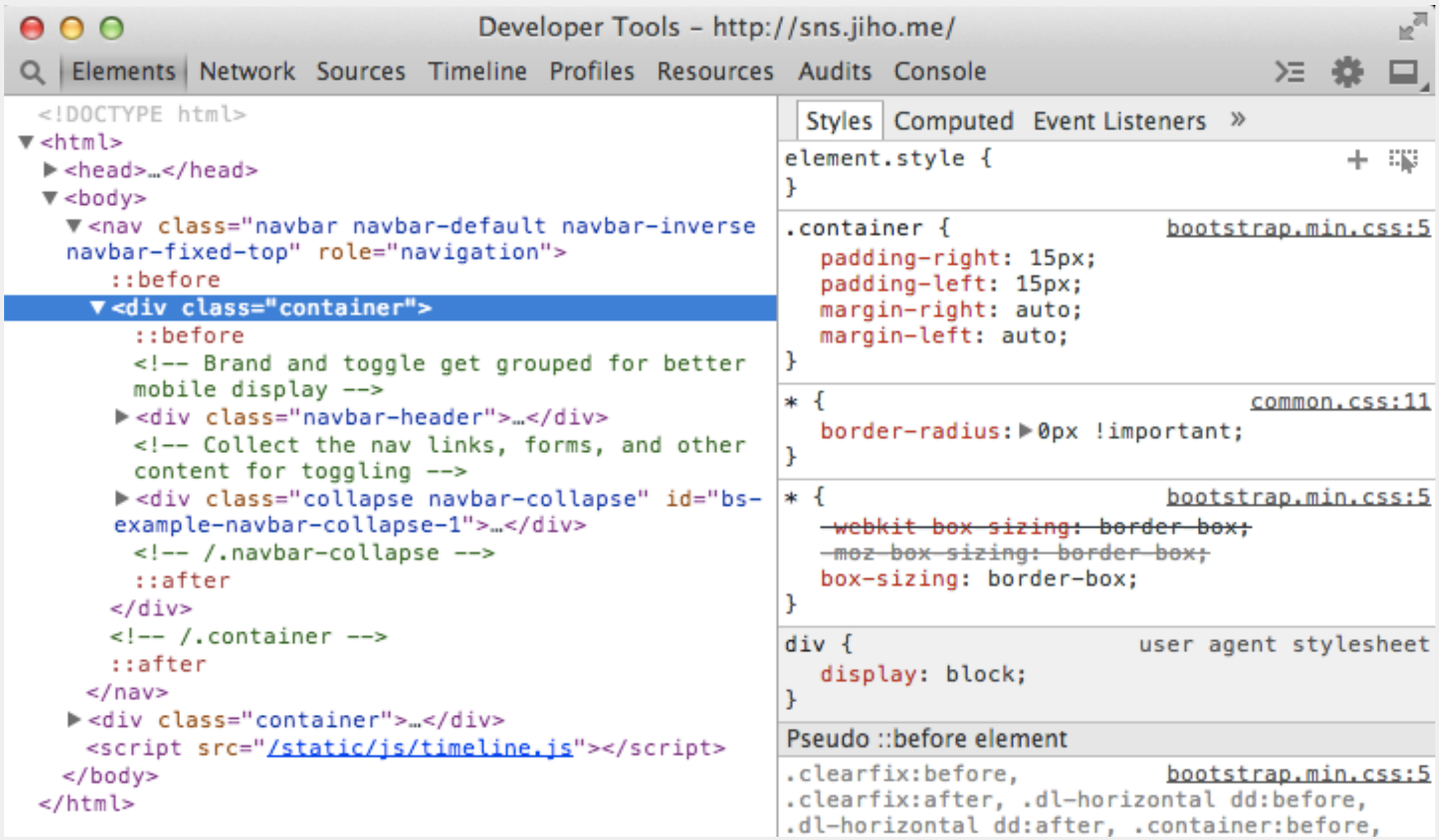
CSS

웹 페이지의 표현


LIKELION SNS
sns.jiho.me
응용프로그램
네이버 뮤직
Apple Korea
Google 지도
YouTube
위키백과
Musicovery
GSMArena
개발참고
디자인
기타 북마크
Toggle navigation
Likelion SNS
Timeline | 홍지호
User List
홍지호
홍지호
2014-08-11 03:18:14
...
Read
0
홍지호
2014-08-07 16:48:00
하하ㅏ ㅎ라라
Read
1
홍지호
2014-08-07 03:07:29
ㅁㅁㅁㅁㅁㅁ
Read
0
장진욱
2014-08-07 01:40:48
홍지호
0
1
0

CSS

웹 페이지의 표현

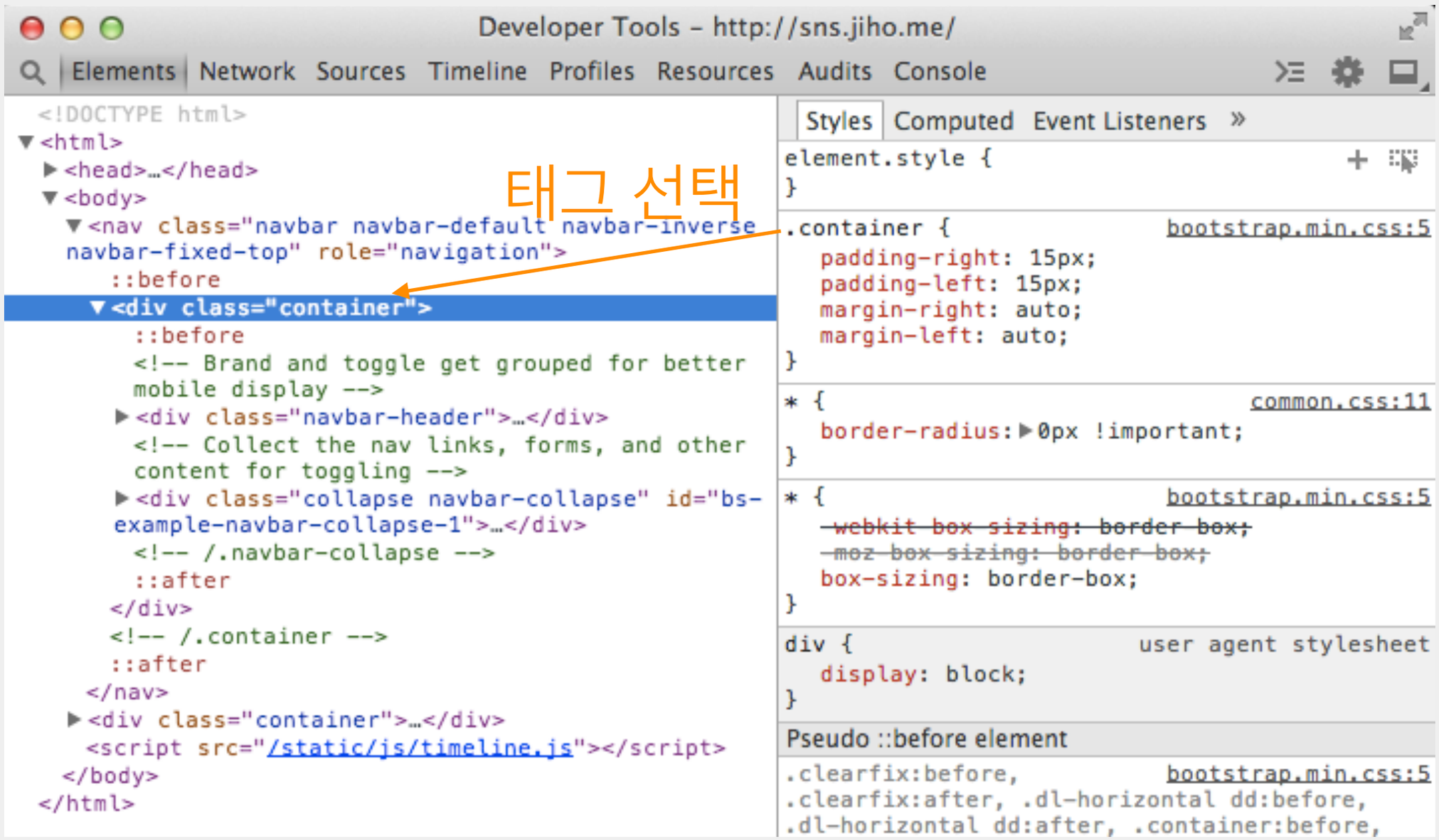

Developer Tools - http://sns.jiho.me/
Elements Network Sources Timeline Profiles Resources Audits Console
<!DOCTYPE html>
<html>
<head>…</head>
<body>
<nav class="navbar navbar-default navbar-inverse navbar-fixed-top" role="navigation">
::before

::before
<!-- Brand and toggle get grouped for better mobile display -->
…
<!-- Collect the nav links, forms, and other content for toggling -->
…
<!-- /.navbar-collapse -->
::after

<!-- /.container -->
::after
</nav>
…
<script src="/static/js/timeline.js"></script>
</body>
</html>
Styles Computed Event Listeners »
element.style {
}
.container { bootstrap.min.css:5
padding-right: 15px;
padding-left: 15px;
margin-right: auto;
margin-left: auto;
}
* { common.css:11
border-radius: 0px !important;
}
* { bootstrap.min.css:5
-webkit-box-sizing: border-box;
-moz-box-sizing: border-box;
box-sizing: border-box;
}
div { user agent stylesheet
display: block;
}
Pseudo ::before element
.clearfix:before, bootstrap.min.css:5
.clearfix:after, .dl-horizontal dd:before,
.dl-horizontal dd:after, .container:before,

CSS

웹 페이지의 표현

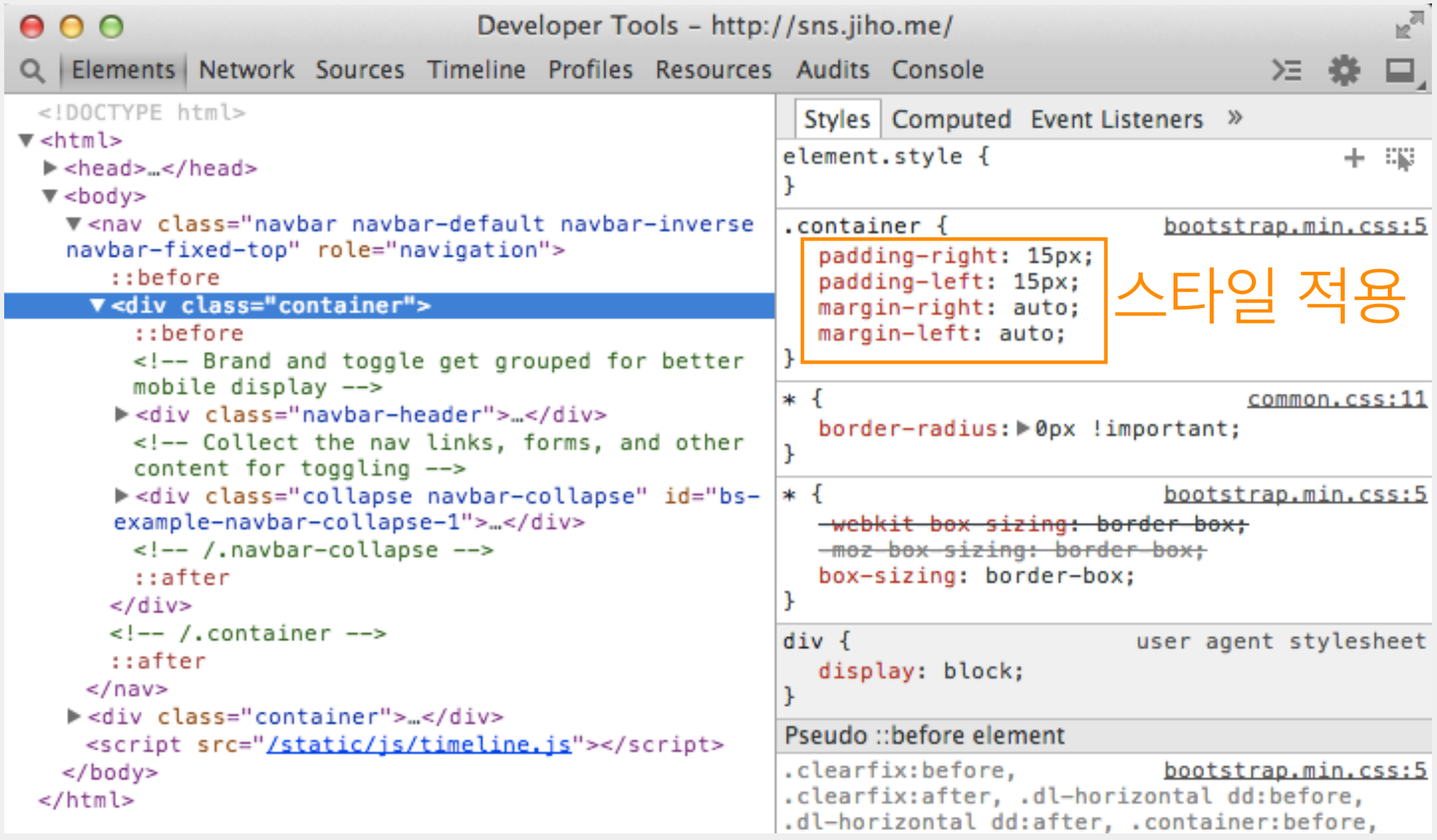

Developer Tools - http://sns.jiho.me/
Elements Network Sources Timeline Profiles Resources Audits Console
태그 선택
<!DOCTYPE html>
<html>
<head>…</head>
<body>
<nav class="navbar navbar-default navbar-inverse navbar-fixed-top" role="navigation">
::before

::before
<!-- Brand and toggle get grouped for better mobile display -->
…
<!-- Collect the nav links, forms, and other content for toggling -->
…
<!-- /.navbar-collapse -->
::after

<!-- /.container -->
::after
</nav>
…
<script src="/static/js/timeline.js"></script>
</body>
</html>
Styles Computed Event Listeners »
element.style {
}
.container { bootstrap.min.css:5
padding-right: 15px;
padding-left: 15px;
margin-right: auto;
margin-left: auto;
}
* { common.css:11
border-radius: 0px !important;
}
* { bootstrap.min.css:5
-webkit-box-sizing: border-box;
-moz-box-sizing: border-box;
box-sizing: border-box;
}
div { user agent stylesheet
display: block;
}
Pseudo ::before element
.clearfix:before, bootstrap.min.css:5
.clearfix:after, .dl-horizontal dd:before,
.dl-horizontal dd:after, .container:before,

CSS

웹 페이지의 표현


Developer Tools - http://sns.jiho.me/
Elements Network Sources Timeline Profiles Resources Audits Console
<!DOCTYPE html>
<html>
<head>…</head>
<body>
<nav class="navbar navbar-default navbar-inverse navbar-fixed-top" role="navigation">
::before

::before
<!-- Brand and toggle get grouped for better mobile display -->
…
<!-- Collect the nav links, forms, and other content for toggling -->
…
<!-- /.navbar-collapse -->
::after

<!-- /.container -->
::after
</nav>
…
<script src="/static/js/timeline.js"></script>
</body>
</html>
Styles Computed Event Listeners »
element.style {
}
.container { bootstrap.min.css:5
padding-right: 15px;
padding-left: 15px;
margin-right: auto;
margin-left: auto;
}
스타일 적용
* { common.css:11
border-radius: 0px !important;
}
* { bootstrap.min.css:5
-webkit-box-sizing: border-box;
-moz-box-sizing: border-box;
box-sizing: border-box;
}
div { user agent stylesheet
display: block;
}
Pseudo ::before element
.clearfix:before, bootstrap.min.css:5
.clearfix:after, .dl-horizontal dd:before,
.dl-horizontal dd:after, .container:before,

Client-Server

Client

- Web Browser
- HTML, CSS, JavaScript

Server

- Web Server

jQuery

JavaScript : 웹 페이지의 동작

jQuery

JavaScript : 웹 페이지의 동작

```
$("_____").____________();
```

jQuery

JavaScript : 웹 페이지의 동작

```
$(" CSS Selector ")._____();
```

jQuery

JavaScript : 웹 페이지의 동작

```
$("_____").____Function____();
```

http://api.jquery.com/

jQuery

JavaScript : 웹 페이지의 동작

```
$("_____").text();
$("_____").text("some text");
$("_____").html();
$("_____").html("<div>some html code</div>");
$("_____").val();
$("_____").val("7");
```

jQuery

JavaScript : 웹 페이지의 동작

```
$("_____").empty();
$("_____").remove();
$("_____").append("<div>some html code</div>");
```

jQuery > Event Handler

JavaScript : 웹 페이지의 동작

- Event Handler : 이벤트를 처리하는 **함수**
- 이벤트 : 클릭, 더블클릭, 키 입력, …
  - http://api.jquery.com/category/events/

jQuery > Event Handler

JavaScript : 웹 페이지의 동작

```
$(document).ready(function(){

});
```

jQuery > Event Handler

JavaScript : 웹 페이지의 동작

```
$(document).ready(function(){

});
```

대상

jQuery > Event Handler

JavaScript : 웹 페이지의 동작

```
$(document).ready(function(){

});
```

이벤트

jQuery > Event Handler

JavaScript : 웹 페이지의 동작

```
$(document).ready(function(){

});
```

핸들러

jQuery > Event Handler

JavaScript : 웹 페이지의 동작

```
$(document).ready(function(){

});
```

문서가

jQuery > Event Handler

JavaScript : 웹 페이지의 동작

```
$(document).ready(function(){
});
```

준비되면

jQuery > Event Handler

JavaScript : 웹 페이지의 동작

```
$(document).ready(function(){
});
```

이 함수를 실행해라

jQuery > Event Handler

JavaScript : 웹 페이지의 동작

```
$(document).ready(function(){
});
```

문서가 준비되면 이 함수를 실행해라

jQuery > Event Handler

JavaScript : 웹 페이지의 동작

```
$(document).ready(function(){

    $("button.btn-delete").click(function(){
        // do something
    });

    $("form").submit(function(){
        //do something

        return false;
    });

});
```

jQuery > Event Handler

JavaScript : 웹 페이지의 동작

```
$(document).ready(function(){

    $("button.btn-delete").click(function(){
        // do something
    });

    $("form").submit(function(){
        //do something

        return false;
    });

});
```

jQuery > Event Handler

JavaScript : 웹 페이지의 동작

```
$(document).ready(function(){

    $("button.btn-delete").click(function(){
        // do something
    });

    $("form").submit(function(){
        //do something

        return false;
    });

});
```

class 가 btn-delete 인 button 태그를 클릭하면 이 함수를 실행해라

jQuery > Event Handler

JavaScript : 웹 페이지의 동작

```
$(document).ready(function(){

    $("button.btn-delete").click(function(){
        // do something
    });

    $("form").submit(function(){
        //do something

        return false;
    });

});
```

form 을 전송하면
이 함수를 실행해라

Client-Server

Client

- Web Browser
- HTML, CSS, JavaScript

Server

- Web Server

...

Server-Side

Back-end

Python

Server-Side Script

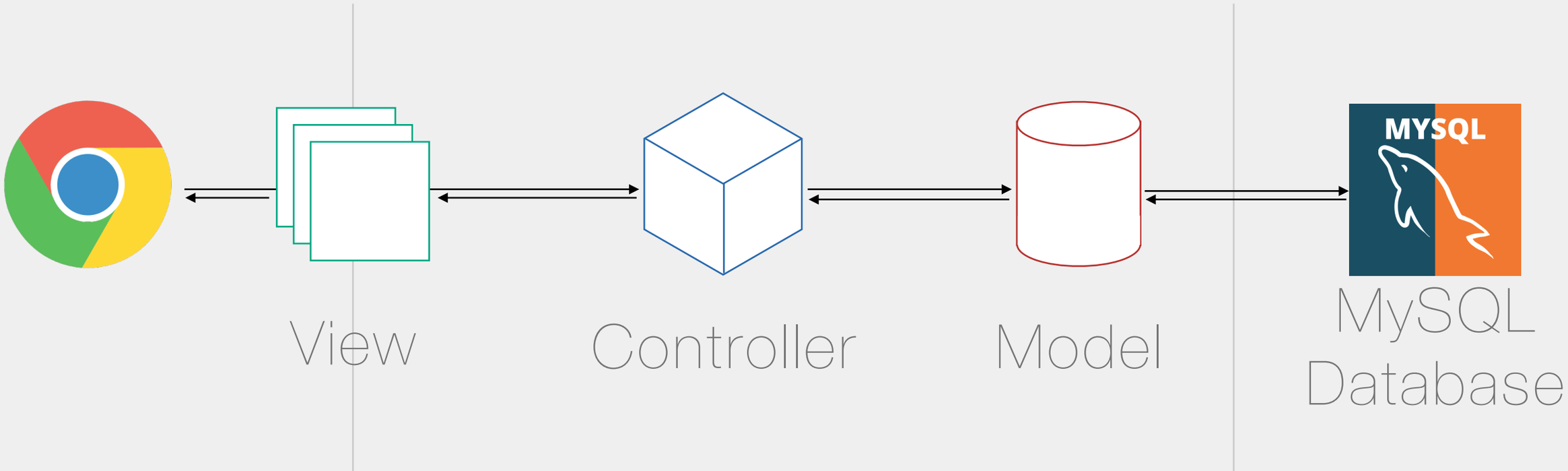
Likelion

Web Service Structure

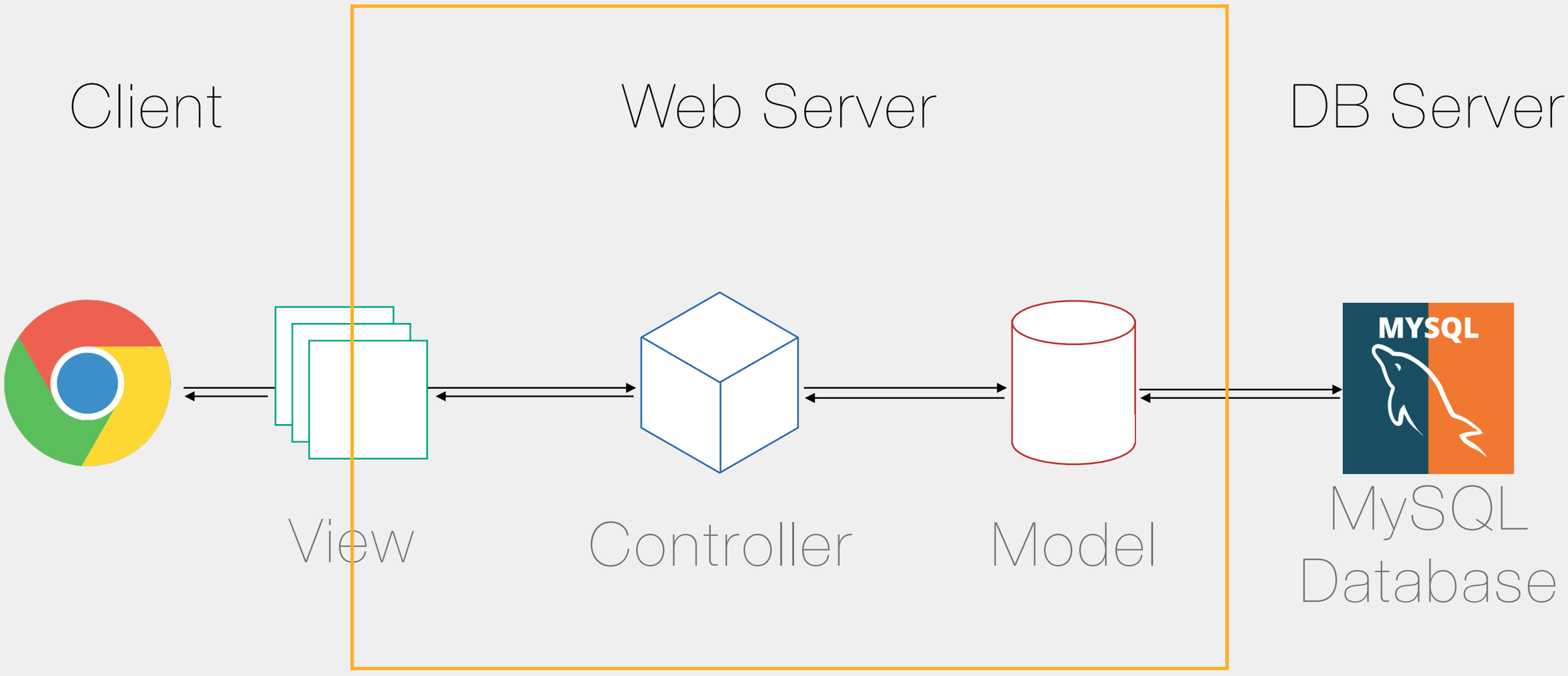
Client

Web Server

DB Server



Web Service Structure



Python : Programming Language

Likelion

Programming Language?

Programming

생각의 표현

Programming

생각의 표현

Programming?

Programming
생각의 표현

할 일을 순서대로 나열한 후

Programming
생각의 표현

할 일을 순서대로 나열한 후

프로그래밍 언어로 표현

Programming
생각의 표현

컴퓨터는 한 번에 하나의 일만 할 수 있다

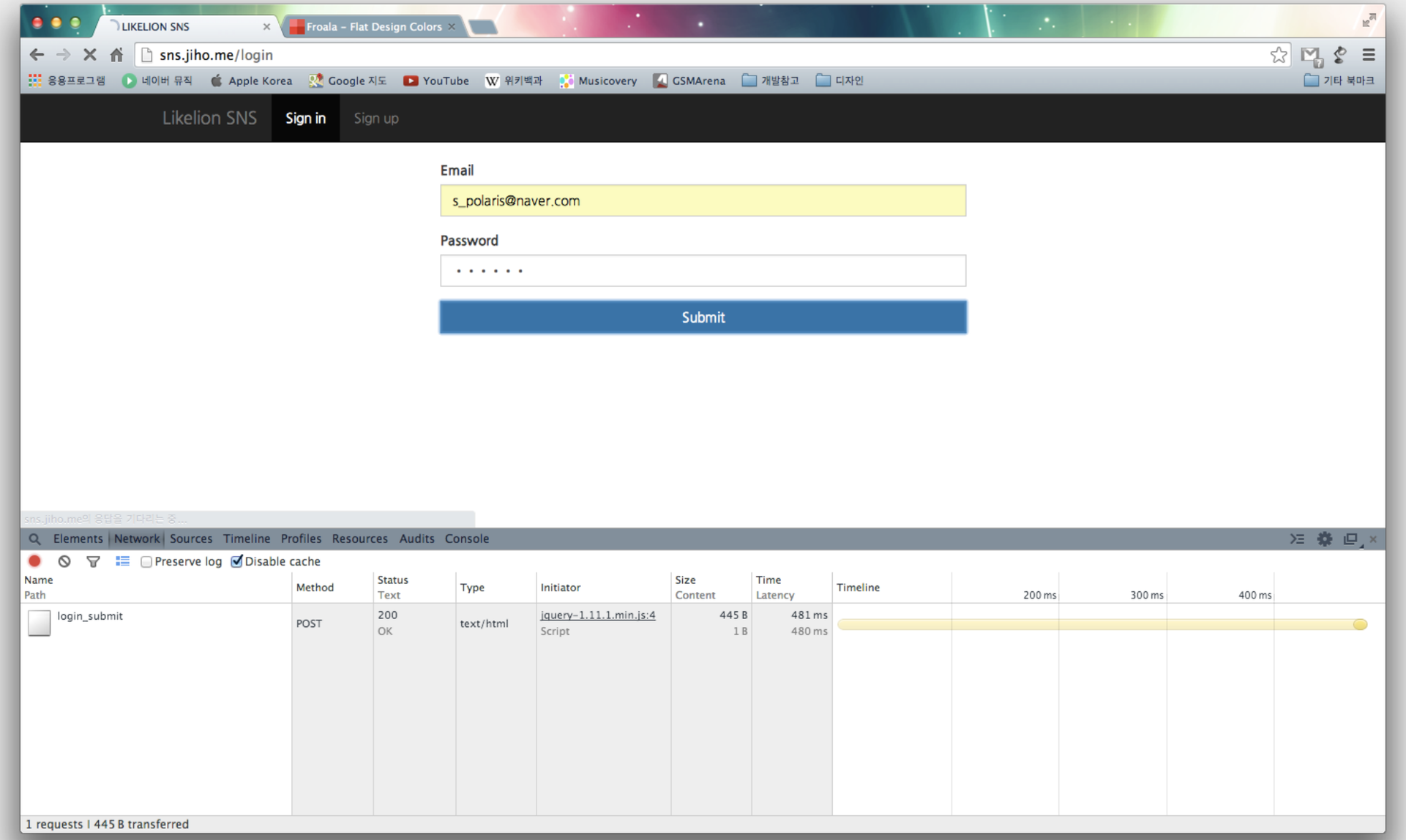
Programming

생각의 표현

예시 : 로그인

Programming

생각의 표현



Programming

생각의 표현

auth.py

```python
@app.route('/login_submit', methods = ['POST'])
def login_submit():
    email = request.form['email']
    password = request.form['password']

    user = login_check(email, password)
    if user.count() == 1:
        # login success
        user = user.one()

        session['username'] = user.username
        session['user_id'] = user.id
        return '0'
    else:
        return '1'
```

user_manager.py

```python
def login_check(email, password):
    return User.query.filter(User.email == email
        , User.password == db.func.md5(password)
        )
```

Client-Server

Client

- Web Browser
- HTML, CSS, JavaScript

Server

- Web Server
- Python

HTML Form

HTML Form

- 사용자가 입력한 데이터를 서버로 보낼 때 사용

HTML Form

```
<form role="form" method="post" action="{{ url_for('login_submit') }}">

    <div class="form-group">
        <label for="input-email">Email</label>
        <input type="email" id="input-email"
        name="email" class="form-control" placeholder="Email" autofocus
        required>
    </div>
    <div class="form-group">
        <label for="input-password">Password</label>
        <input type="password" class="form-control" id="input-password"
        name="password" placeholder="Password" required>
    </div>


    <button type="submit" class="btn btn-primary btn-block">Submit</button>

</form>
```

HTML Form

```
<form role="form" method="post" action="{{ url_for('login_submit') }}">

    <div class="form-group">
        <label for="input-email">Email</label>
        <input type="email" id="input-email"
        name="email" class="form-control" placeholder="Email" autofocus
        required>
    </div>
    <div class="form-group">
        <label for="input-password">Password</label>
        <input type="password" class="form-control" id="input-password"
        name="password" placeholder="Password" required>
    </div>


    <button type="submit" class="btn btn-primary btn-block">Submit</button>

</form>
```

HTML Form

```
<form role="form" method="post" action="{{ url_for('login_submit') }}">

    <div class="form-group">
        <label for="input-email">Email</label>
        <input type="email" id="input-email"
        name="email" class="form-control" placeholder="Email" autofocus
        required>
    </div>
    <div class="form-group">
        <label for="input-password">Password</label>
        <input type="password" class="form-control" id="input-password"
        name="password" placeholder="Password" required>
    </div>


    <button type="submit" class="btn btn-primary btn-block">Submit</button>

</form>
```

HTML Form

```
<form role="form" method="post" action="{{ url_for('login_submit') }}">

    <div class="form-group">
        <label for="input-email">Email</label>
        <input type="email" id="input-email"
        name="email" class="form-control" placeholder="Email" autofocus
        required>
    </div>
    <div class="form-group">
        <label for="input-password">Password</label>
        <input type="password" class="form-control" id="input-password"
        name="password" placeholder="Password" required>
    </div>


    <button type="submit" class="btn btn-primary btn-block">Submit</button>

</form>
```

HTML Form

```
<form role="form" method="post" action="{{ url_for('login_submit') }}">

    <div class="form-group">
        <label for="input-email">Email</label>
        <input type="email" id="input-email"
        name="email" class="form-control" placeholder="Email" autofocus
        required>
    </div>
    <div class="form-group">
        <label for="input-password">Password</label>
        <input type="password" class="form-control" id="input-password"
        name="password" placeholder="Password" required>
    </div>


    <button type="submit" class="btn btn-primary btn-block">Submit</button>

</form>
```

HTML Form

```
<form role="form" method="post" action="{{ url_for('login_submit') }}">

    <div class="form-group">
        <label for="input-email">Email</label>
        <input type="email" id="input-email"
        name="email" class="fo
         required>
    </div>
    <div class="form-group">
        <label for="input-pass
        <input type="password"
        name="password" placeh
    </div>


    <button type="submit" clas

</form>
```

```
@app.route('/login_submit', methods = ['POST'])
def login_submit():
    email = request.form['email']
    password = request.form['password']

    user = login_check(email, password)
    if user:
        # login success
        user = user.first()
        session['username'] = user.username
        session['user_id'] = user.id
        return '0'
    else:
        return '1'
```

HTML Form

```
<form role="form" method="post" action="{{ url_for('login_submit') }}">

    <div class="form-group">
        <label for="input-email">Email</label>
        <input type="email" id="input-email"
        name="email" class="fo
        required>
    </div>
    <div class="form-group">
        <label for="input-pass
        <input type="password"
        name="password" placeh
    </div>


    <button type="submit" clas

</form>
```

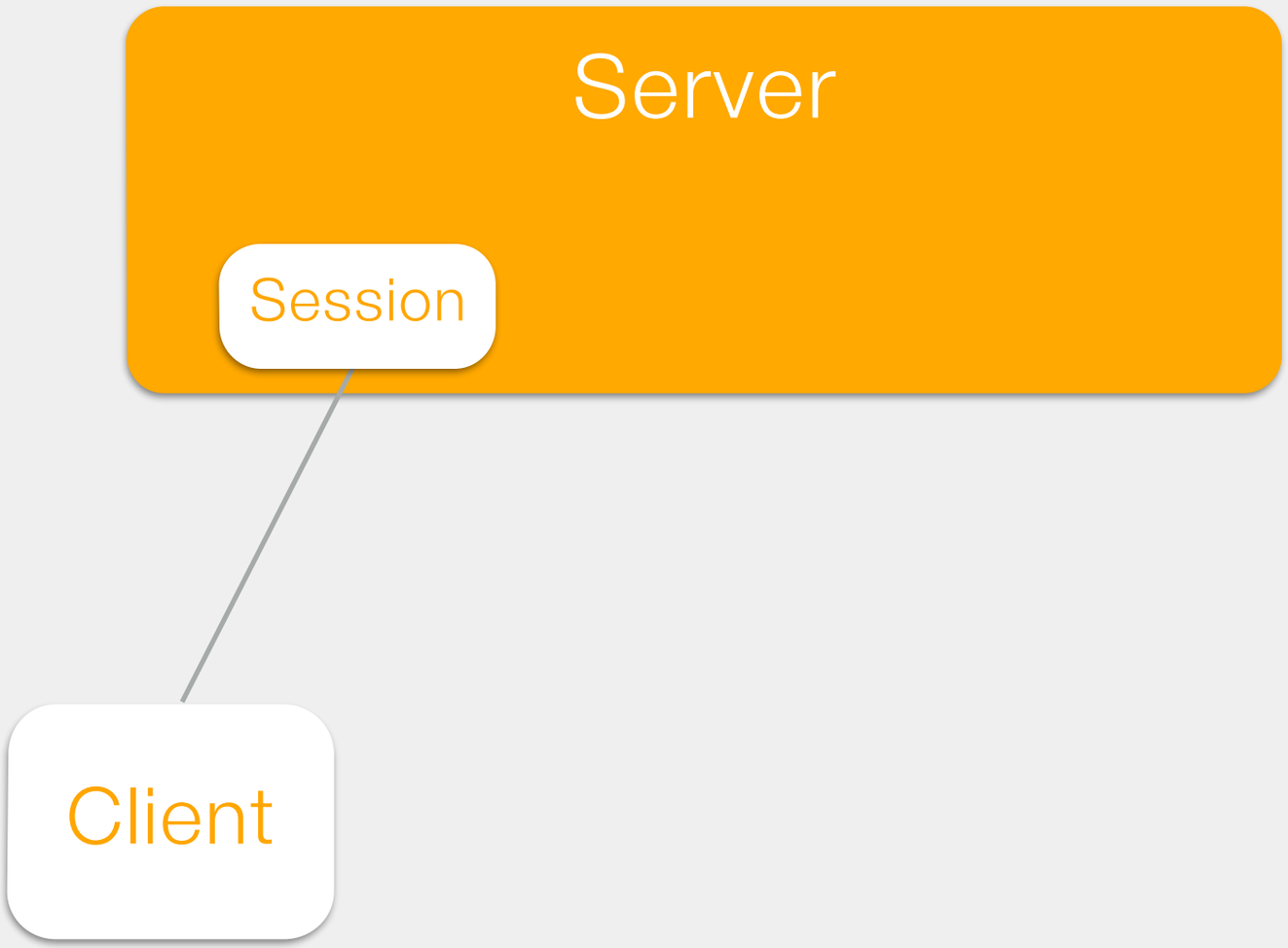
```
@app.route('/login_submit', methods = ['POST'])
def login_submit():
    email = request.form['email']
    password = request.form['password']

    user = login_check(email, password)
    if user:
        # login success
        user = user.first()
        session['username'] = user.username
        session['user_id'] = user.id
        return '0'
    else:
        return '1'
```

Session

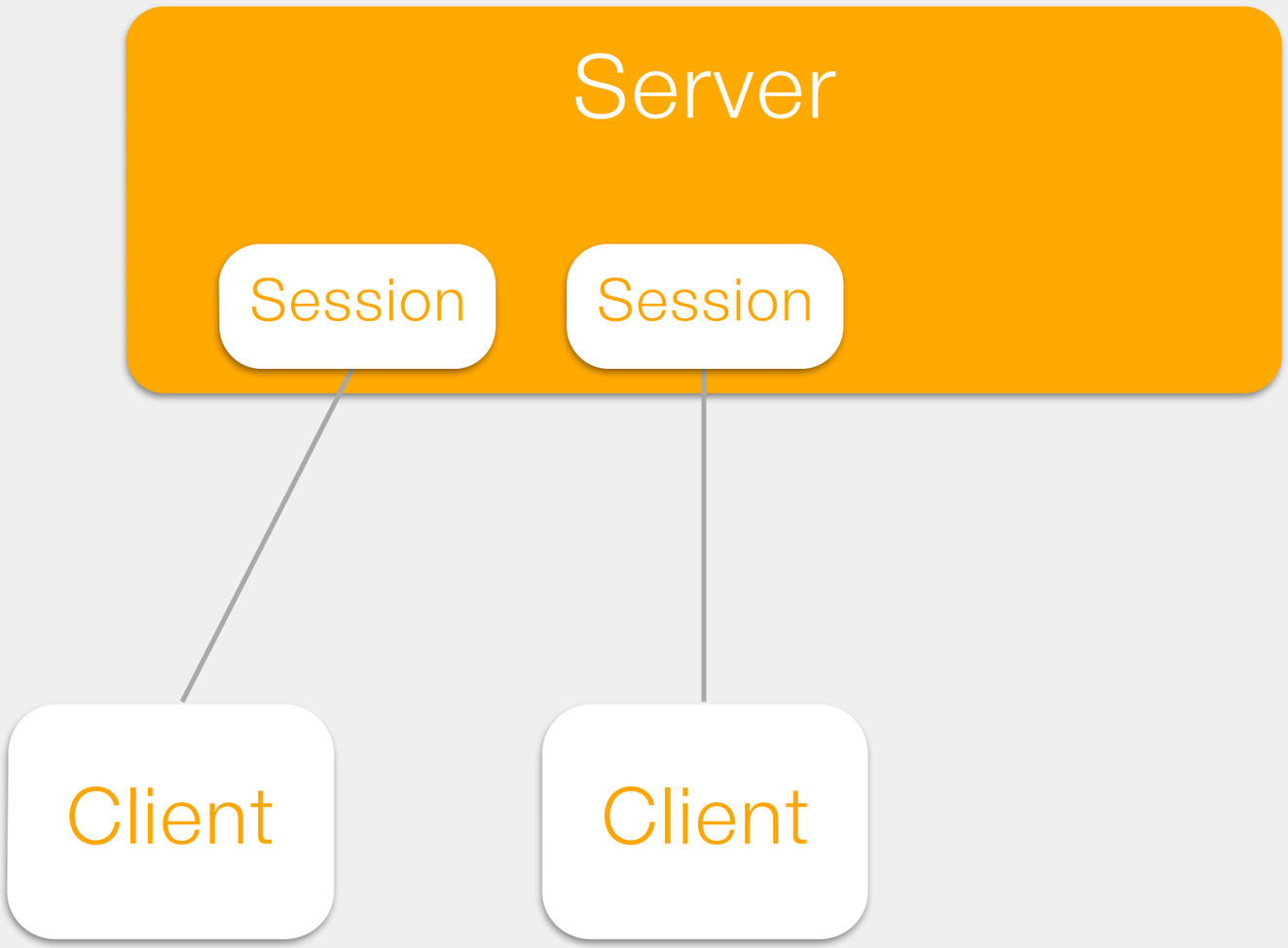
Session

Server

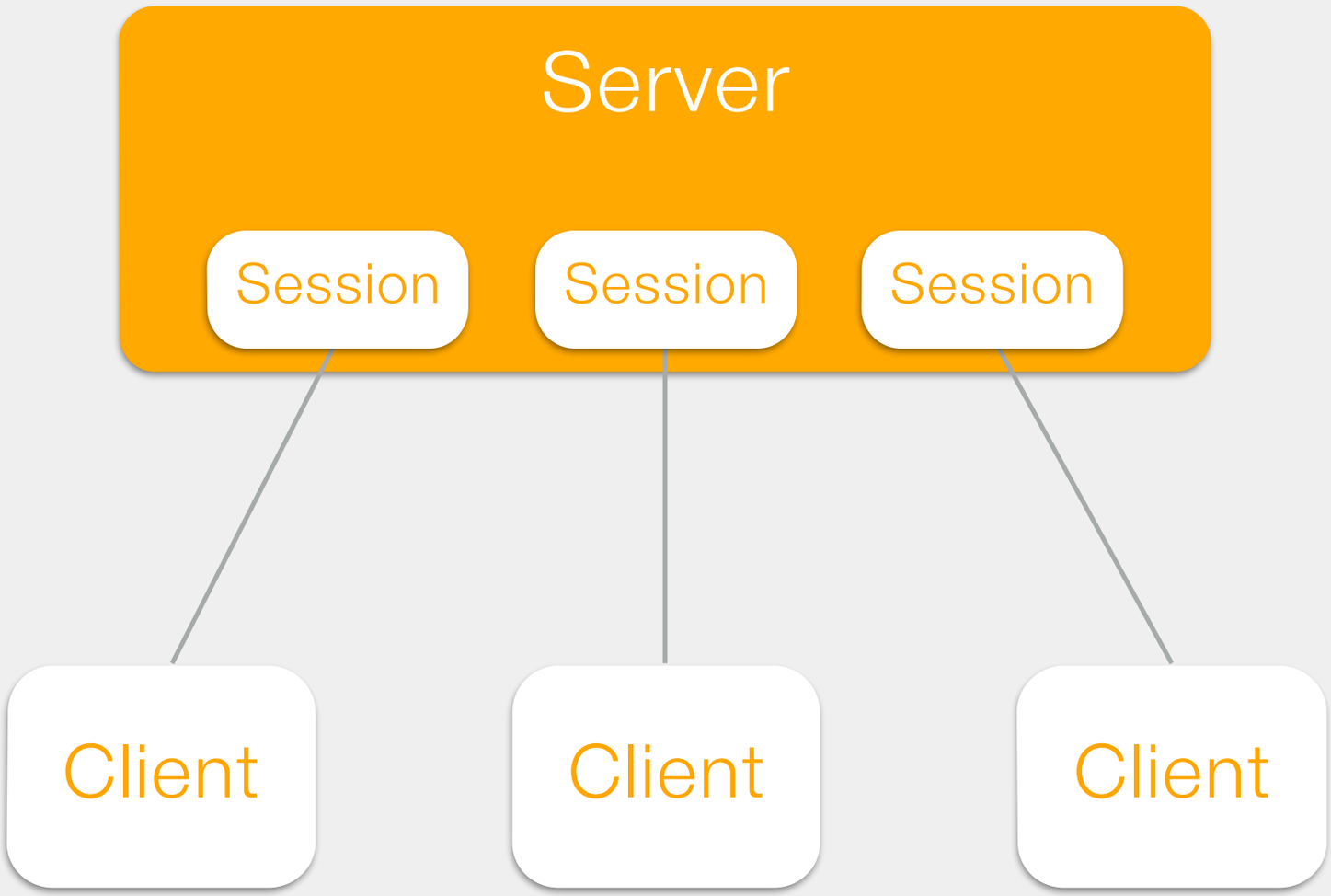
Session



Session



Session



Session

예시) 로그인

```python
@app.route('/login_submit', methods = ['POST'])
def login_submit():
    email = request.form['email']
    password = request.form['password']

    user = login_check(email, password)
    if user:
        # login success
        user = user.first()
        session['username'] = user.username
        session['user_id'] = user.id
        return '0'
    else:
        return '1'
```

Session

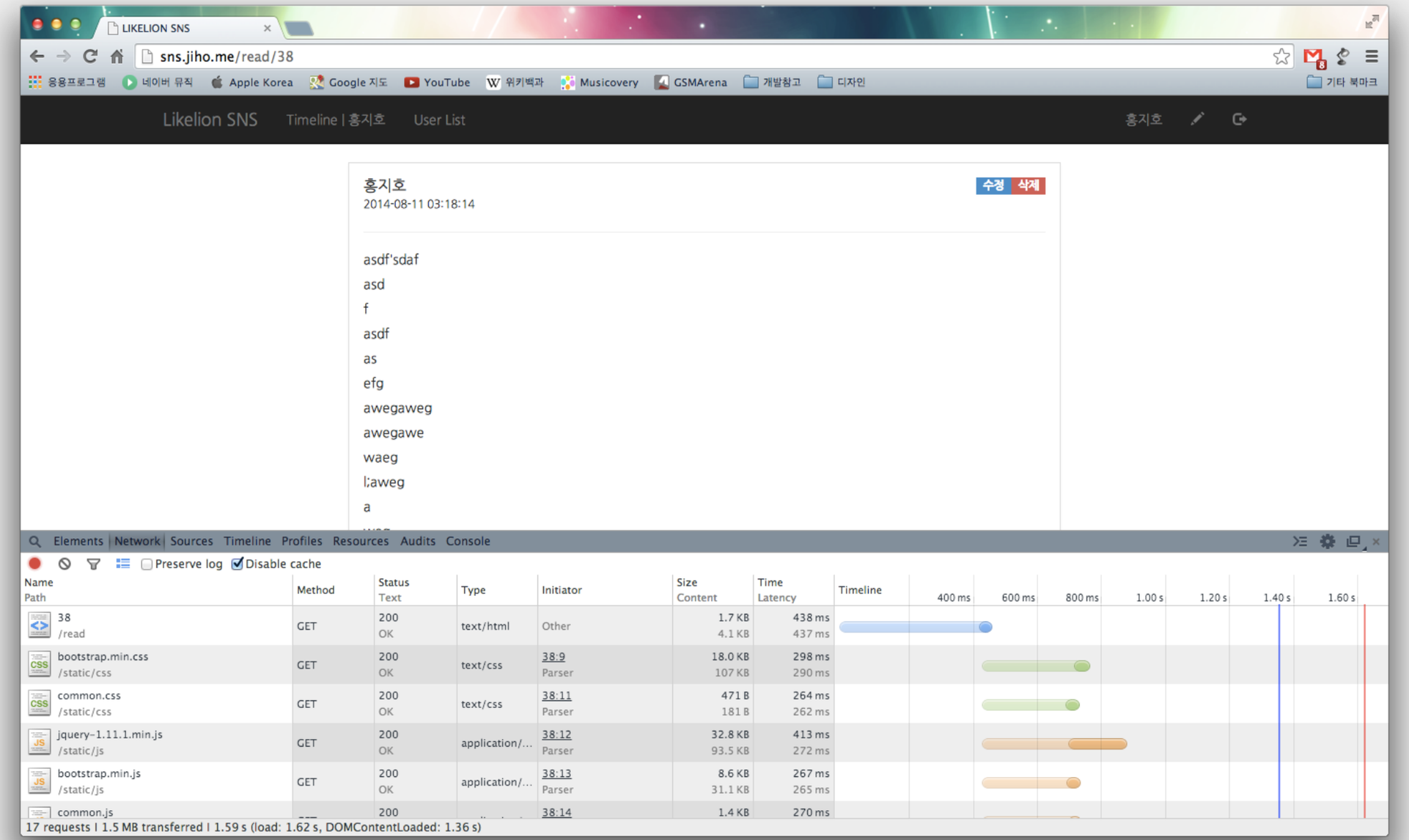
예시) 담벼락

```
@app.route('/', defaults={'wall_id':0})
@app.route('/timeline', defaults={'wall_id':0})
@app.route('/timeline/<int:wall_id>')
def timeline(wall_id) :
    if not is_login():
        return redirect(url_for('login'))
    if wall_id == 0:
        wall_id = session['user_id']

    session['wall_id'] = wall_id
    session['wall_username'] = get_user(wall_id).username

    context = {
        'posts':get_wall_posts(wall_id,0),
        'BS':BS
    }
    postList = render_template('post_list.html',context=context)
    context = {
        'post_list':postList
    }
    return render_template('timeline.html', context=context)
```

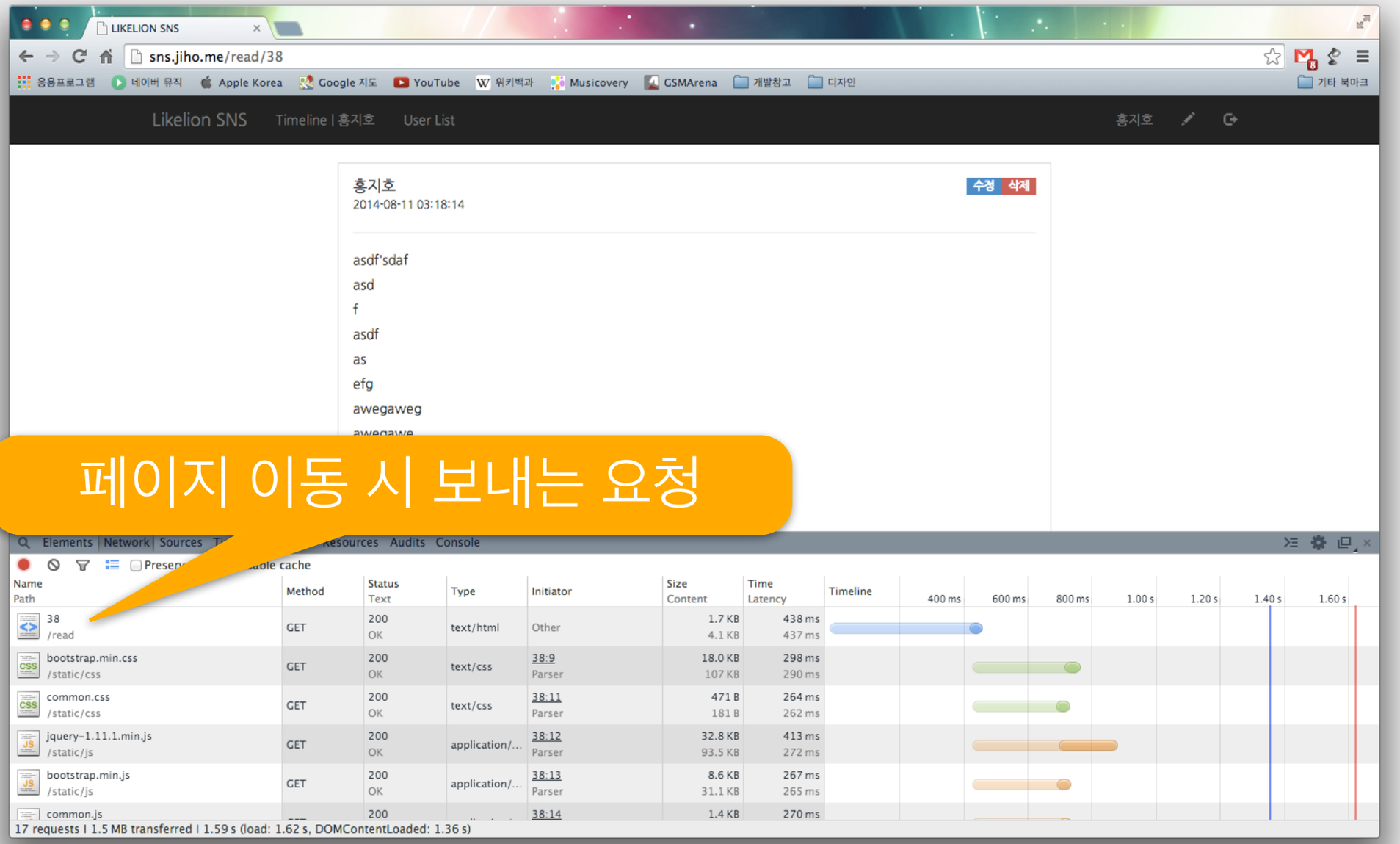
AJAX

Asynchronous JavaScript and XML

AJAX

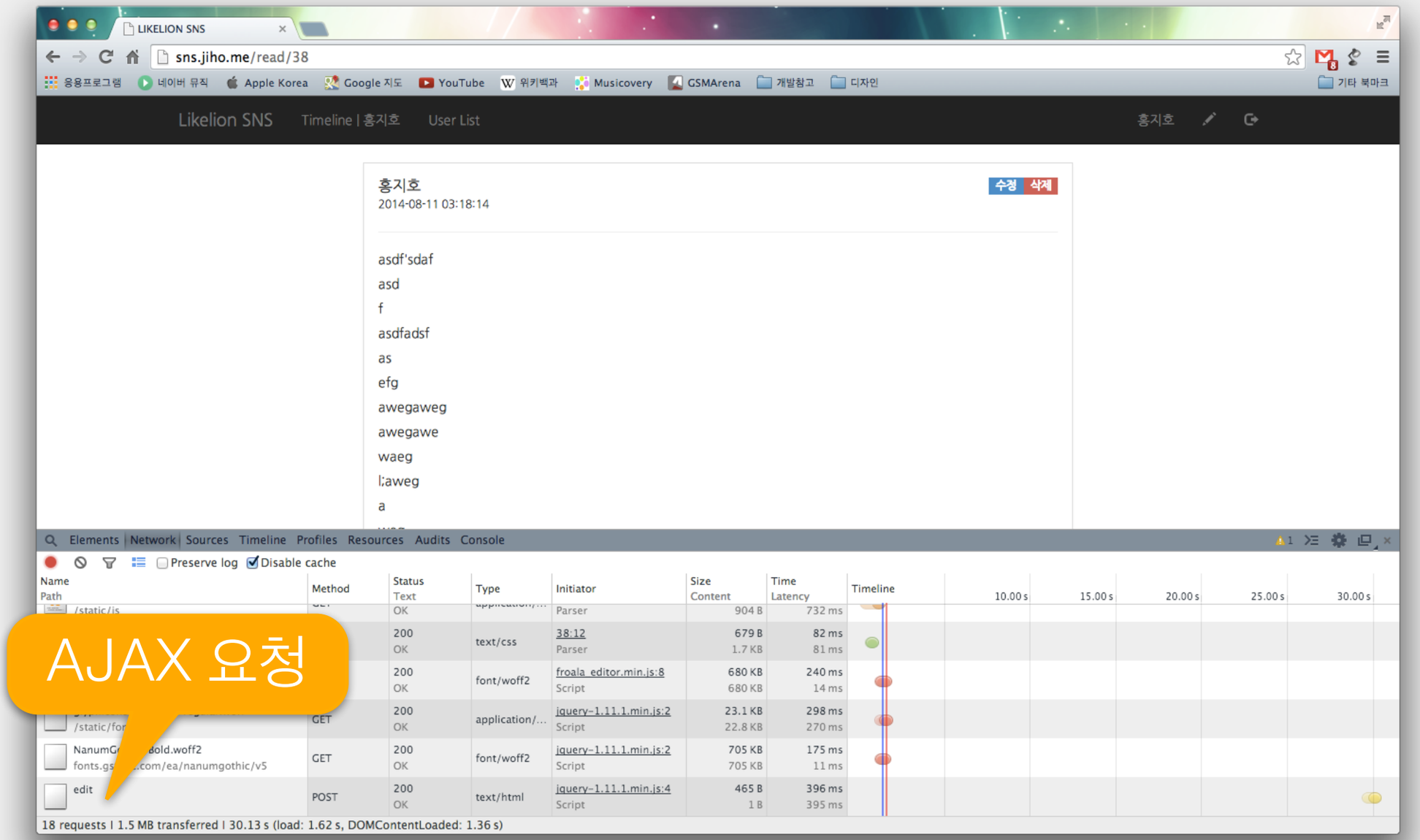

LIKELION SNS
sns.jiho.me/read/38
응용프로그램
네이버 뮤직
Apple Korea
Google 지도
YouTube
위키백과
Musicovery
GSMArena
개발참고
디자인
기타 북마크
Likelion SNS
Timeline | 홍지호
User List
홍지호
홍지호
2014-08-11 03:18:14
수정
삭제
asdf'sdaf
asd
f
asdf
as
efg
awegaweg
awegawe
waeg
l;aweg
a
Elements
Network
Sources
Timeline
Profiles
Resources
Audits
Console
Preserve log
Disable cache
Name
Path
Method
Status
Text
Type
Initiator
Size
Content
Time
Latency
Timeline
400 ms
600 ms
800 ms
1.00 s
1.20 s
1.40 s
1.60 s
38
/read
GET
200
OK
text/html
Other
1.7 KB
4.1 KB
438 ms
437 ms
bootstrap.min.css
/static/css
GET
200
OK
text/css
38:9
Parser
18.0 KB
107 KB
298 ms
290 ms
common.css
/static/css
GET
200
OK
text/css
38:11
Parser
471 B
181 B
264 ms
262 ms
jquery-1.11.1.min.js
/static/js
GET
200
OK
application/...
38:12
Parser
32.8 KB
93.5 KB
413 ms
272 ms
bootstrap.min.js
/static/js
GET
200
OK
application/...
38:13
Parser
8.6 KB
31.1 KB
267 ms
265 ms
common.js
200
38:14
1.4 KB
270 ms
17 requests | 1.5 MB transferred | 1.59 s (load: 1.62 s, DOMContentLoaded: 1.36 s)

AJAX

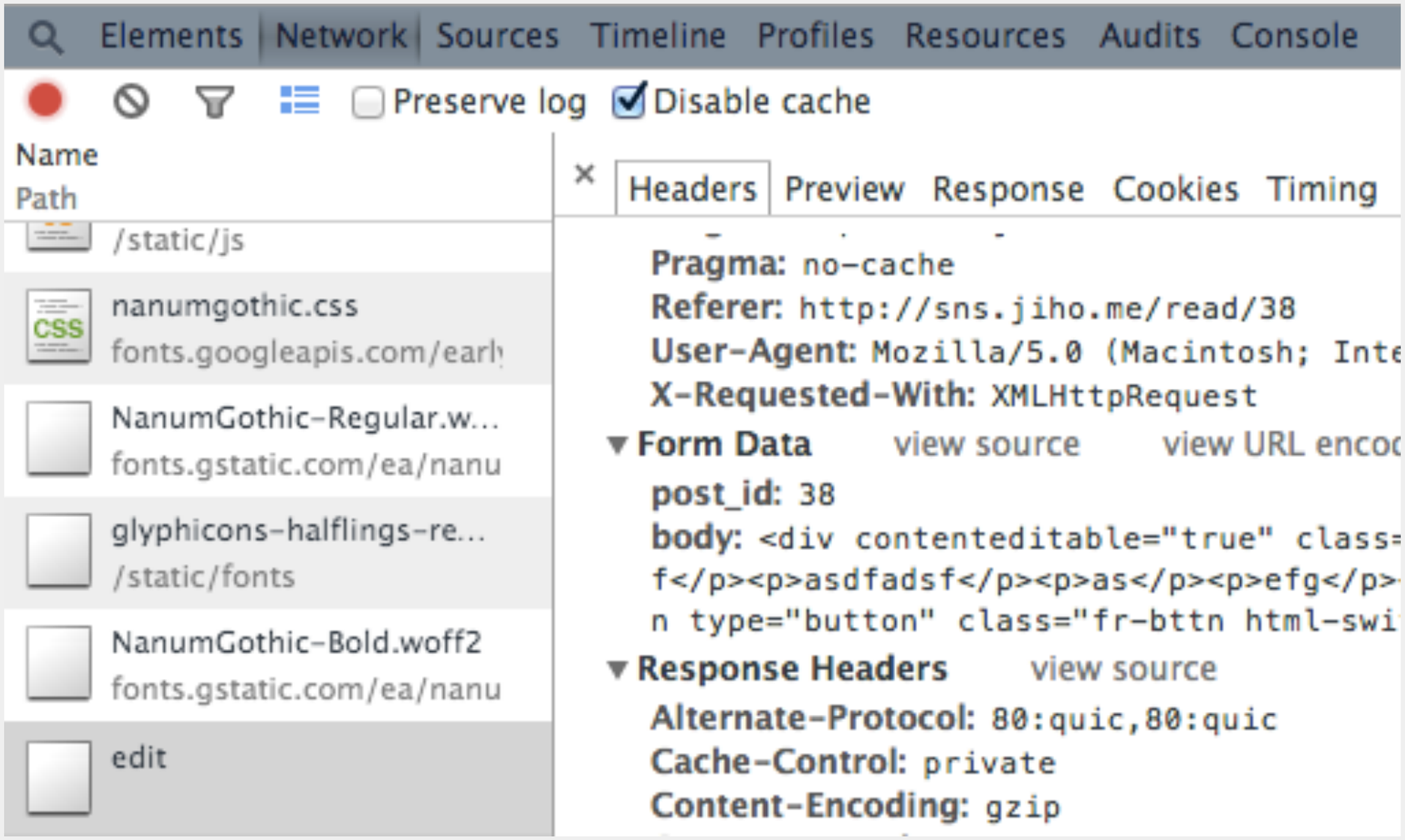


페이지 이동 시 보내는 요청

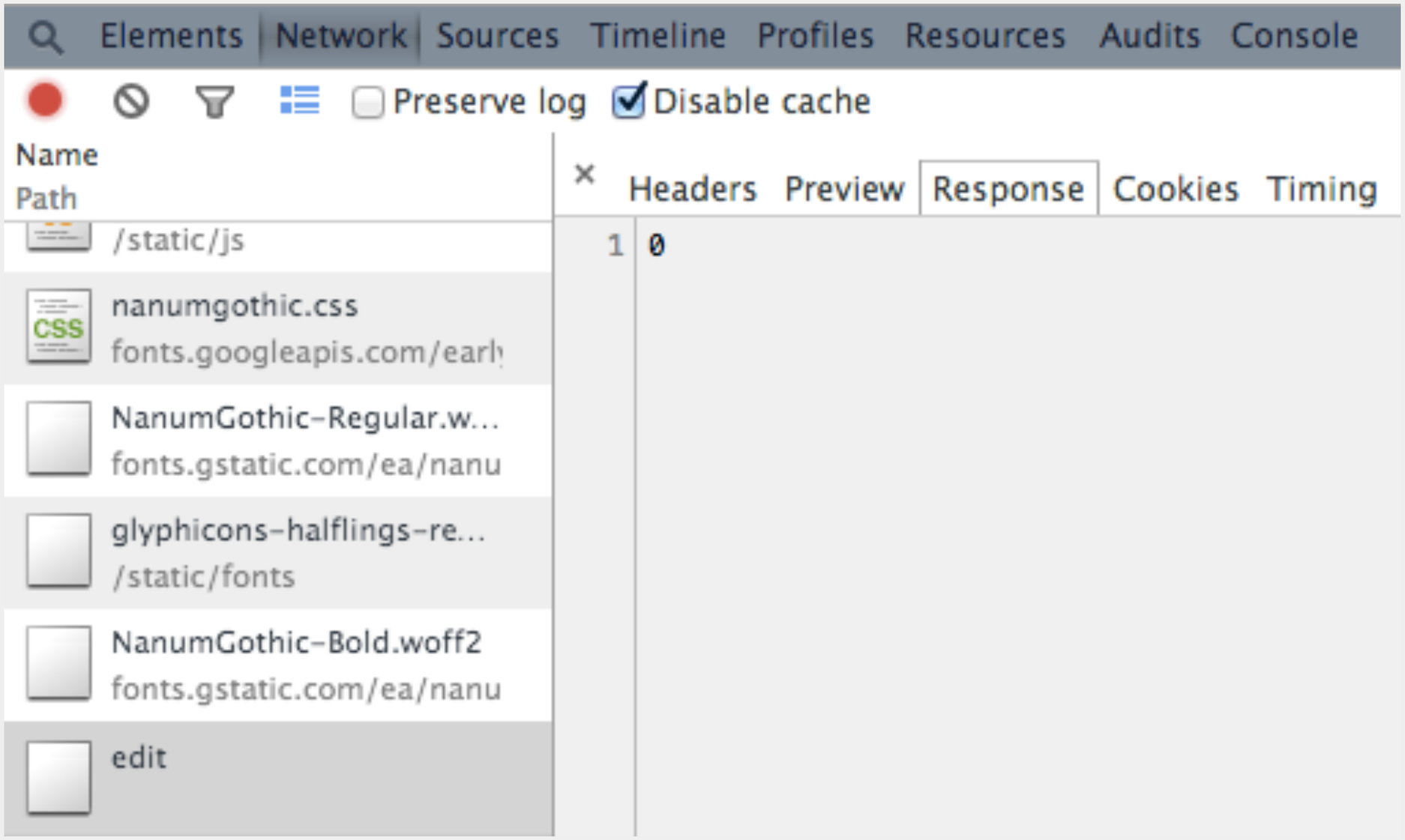
AJAX


LIKELION SNS
sns.jiho.me/read/38
Likelion SNS
Timeline | 홍지호
User List
홍지호
홍지호
2014-08-11 03:18:14
수정
삭제
asdf'sdaf
asd
f
asdfadsf
as
efg
awegaweg
awegawe
waeg
l;aweg
a
Elements
Network
Sources
Timeline
Profiles
Resources
Audits
Console
Preserve log
Disable cache
AJAX 요청
18 requests | 1.5 MB transferred | 30.13 s (load: 1.62 s, DOMContentLoaded: 1.36 s)

AJAX


Elements
Network
Sources
Timeline
Profiles
Resources
Audits
Console
Preserve log
Disable cache
Name
Path
/static/js
nanumgothic.css
fonts.googleapis.com/earl
NanumGothic-Regular.w...
fonts.gstatic.com/ea/nanu
glyphicons-halflings-re...
/static/fonts
NanumGothic-Bold.woff2
fonts.gstatic.com/ea/nanu
edit
Headers
Preview
Response
Cookies
Timing
Pragma: no-cache
Referer: http://sns.jiho.me/read/38
User-Agent: Mozilla/5.0 (Macintosh; Inte
X-Requested-With: XMLHttpRequest
Form Data
view source
view URL encoc
post_id: 38
body: <div contenteditable="true" class=
f asdfadsf as efg
n type="button" class="fr-bttn html-swi
Response Headers
view source
Alternate-Protocol: 80:quic,80:quic
Cache-Control: private
Content-Encoding: gzip

AJAX


Elements
Network
Sources
Timeline
Profiles
Resources
Audits
Console
Preserve log
Disable cache
Name
Path
/static/js
nanumgothic.css
fonts.googleapis.com/earl
NanumGothic-Regular.w...
fonts.gstatic.com/ea/nanu
glyphicons-halflings-re...
/static/fonts
NanumGothic-Bold.woff2
fonts.gstatic.com/ea/nanu
edit
Headers
Preview
Response
Cookies
Timing
1
0

AJAX

예시 : 글 수정

GROUP 1
read.html
edit.js
GROUP 2
post.py
FOLDERS
sns-jiho-me
application
controllers
__init__.py
auth.py
errorhandlers.py
index.py
post.py
models
static
css
editor
fonts
js
auth.js
bootstrap.min.js
common.js
edit.js
join.js
jquery-1.11.1.min.
login.js
read.js
timeline.js
write.js
res
templates
join.html
layout.html

read.html | edit.js

```
1  $(document).ready(function(){
2
3      $(document).on('click','#btn-edit-post-submit',
           function(){
4          var body = $("#post-body").html();
5          var postId = $("#input-post-id").val();
6
7          show_progress();
8          $.ajax({
9              url:'/edit',
10             data:{'post_id':postId,'body':body},
11             type:'post',
12             success:function(response){
13             if (response=='0') {
14                 show_toast('수정되었습니다','success');
15             } else if (response=='1') {
16                 show_toast('수정 권한이 없습니다','error');
17             } else {
18                 show_toast('오류가 발생하였습니다. 잠시 후에 다시
                       시도해주세요','error');
19             },
20             error:function(){
21                 show_toast('오류가 발생하였습니다. 잠시 후에 다시
                       시도해주세요','error');
22             },
23             complete:function(){
24                 hide_progress();
25             }
26         });
27     });
28 });
29
30
31
```

post.py

```
46 @app.route('/edit',methods=['POST'])
47 def edit():
48     targetPost = get_post(request.form['post_id'])
49     if targetPost.user_id != session['user_id']:
50         return '1'
51
52     edit_post(request.form)
53     return '0'
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76

77
78
79
80

81
82
83
```

AJAX

예시 : Autoload

GROUP 1
read.html
edit.js
timeline.js
GROUP 2
index.py
post.py
FOLDERS
sns-jiho-me
application
controllers
__init__.py
auth.py
errorhandlers.py
index.py
post.py
models
static
css
editor
fonts
js
auth.js
bootstrap.min.js
common.js
edit.js
join.js
jquery-1.11.1.min.
login.js
read.js
timeline.js
write.js
res
templates

read.html edit.js timeline.js

```
1  //autoload
2  var last=false;
3  var running=false;
4
5  $(document).ready(function(){
6      //get_posts(0);
7
8      $(document).on('scroll',function(){
9          if (last) return false;
10
11         var lastPost = $('#post-list>div.panel:last-child'
               );
12         var lastPostId = lastPost.data('postId');
13
14
15         if ($('body').scrollTop()+$(window).height() >
               lastPost.offset().top) {
16
17             get_posts(lastPostId);
18
19         }
20     });
21 });
22
23 function get_posts(start_id) {
24     if (!running) {
25         running = true;
26
27         show_progress();
28         $.post('/get_posts',{'start_id':start_id},function
               (response){
29             if (response.trim()=='') {
30                 last=true;
31                 return false;
32             }
33             $("#post-list").append(response);
34         }).always(function(){
35             hide_progress();
36             running = false;
37         });
38     }
```

index.py post.py

```
78 @app.route('/get_posts', methods=['POST'])
79 def get_posts():
80     posts = get_wall_posts(session['wall_id'],request.
           form['start_id'])
81
82     if posts.count() == 0:
83         return ''
84
85     context = {
86         'posts':posts,
87         'BS':BS
88     }
89
90
91
92     return render_template('post_list.html',context =
           context)
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
```

JSON

JavaScript Object Notation

JSON

Web Client

Web Server

URL + Data

Data

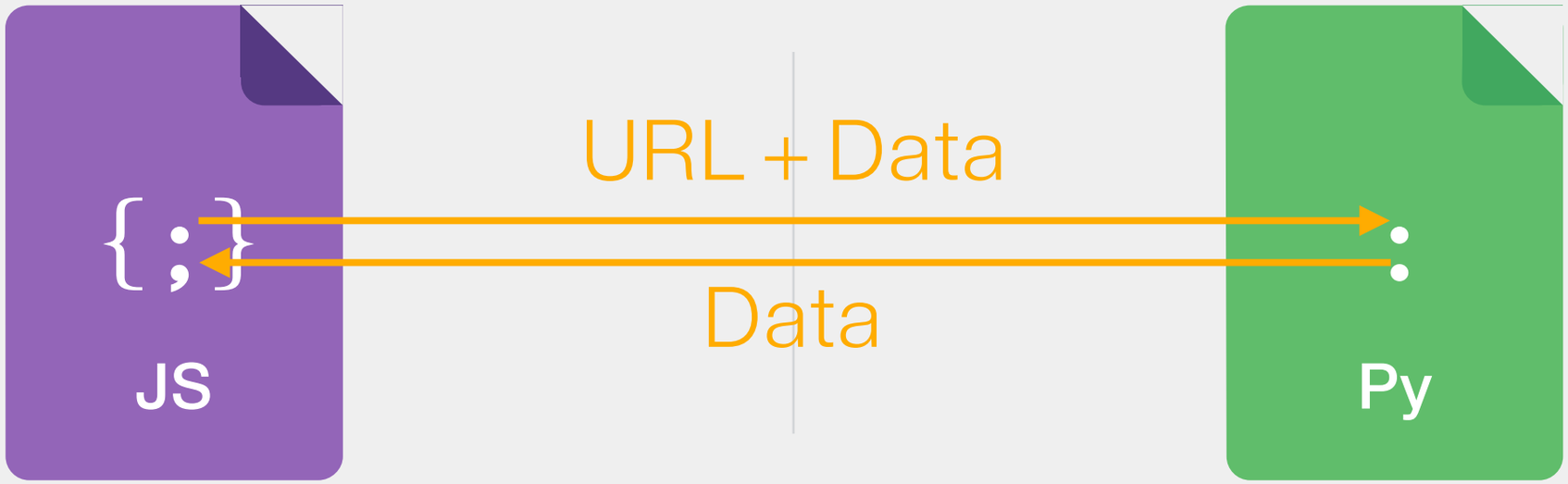
Client Computer + Client Program

Server Computer + Server Program

JSON

Web Client

Web Server



JSON

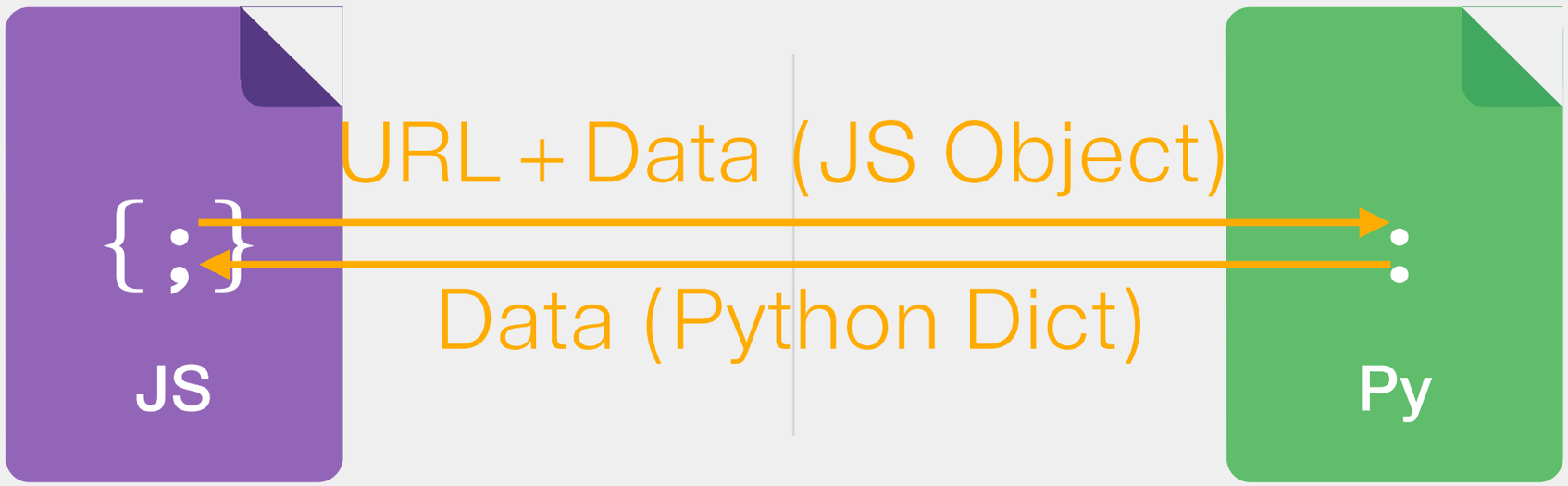
```javascript
// JavaScript - Object

var jiho = {
    name:'Jiho Hong',
    email:'jiho.hong@hotmail.com',
    phone:'010-4573-6906'
};

```

```python
# Python - Dict

jiho = {
    'name':'Jiho Hong',
    'email':'jiho.hong@hotmail.com',
    'phone':'010-4573-6906'
}

```

JSON

Web Client

Web Server



JSON

- JavaScript Object Notation
- 서로 다른 프로그래밍 언어 사이에서 데이터를 주고받기 위해 사용되는 데이터 표현 방식 중 하나
- 자바스크립트에서 object를 표현하는 문법을 그대로 **문자열**로 바꾼 것

JSON

Request

Web Client

Object or Array

↓

JSON String

Web Server

Dict or List

↑

JSON String

JSON
Response

Web Client

Object or Array

↑

JSON String

Web Server

Dict or List

↓

JSON String

←

...

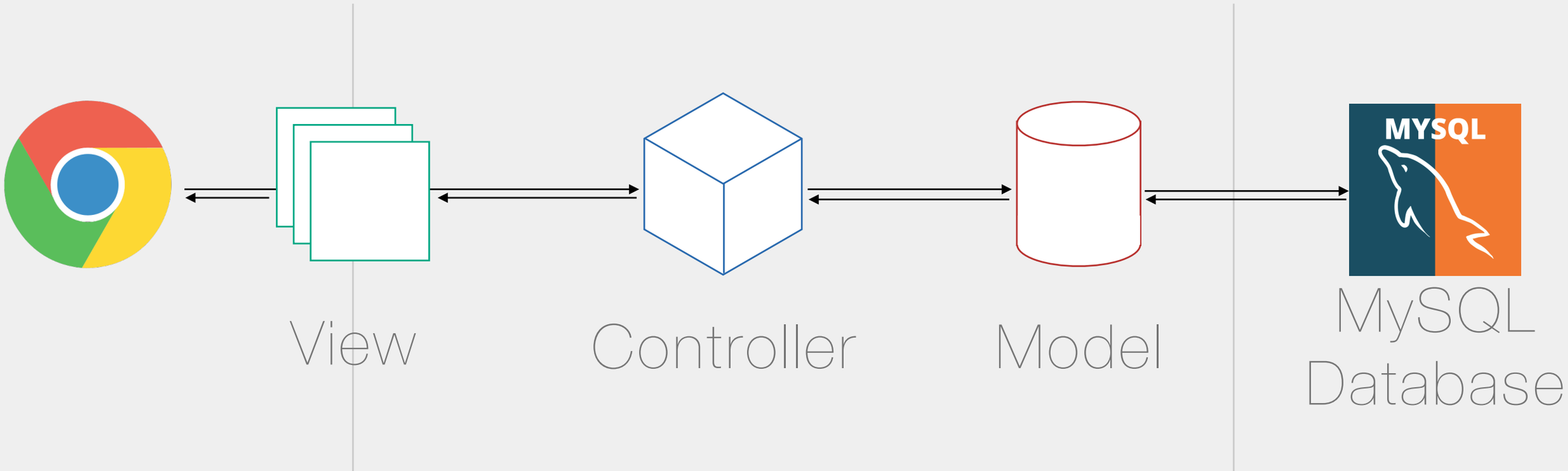
MVC

Web Service Structure

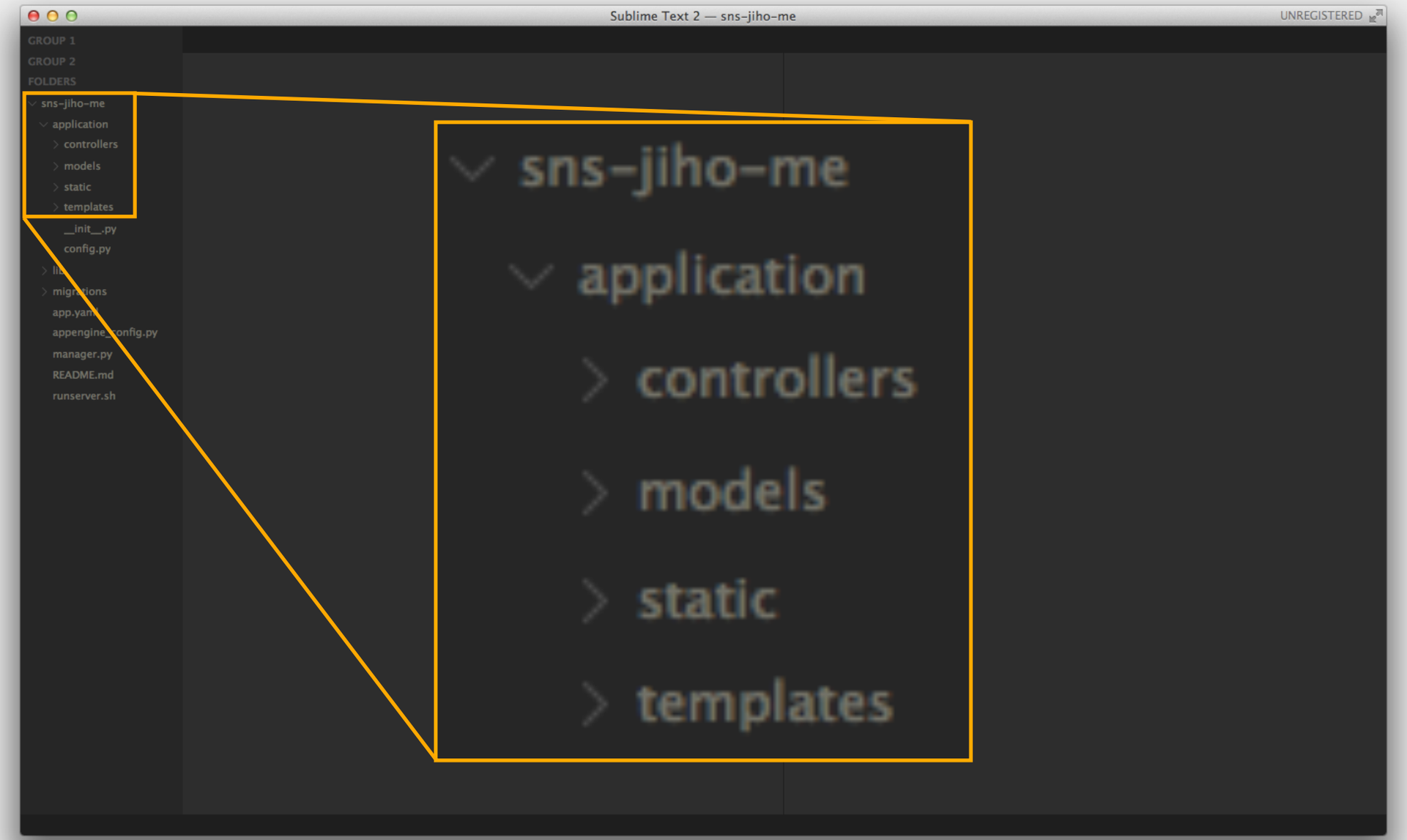
Client

Web Server

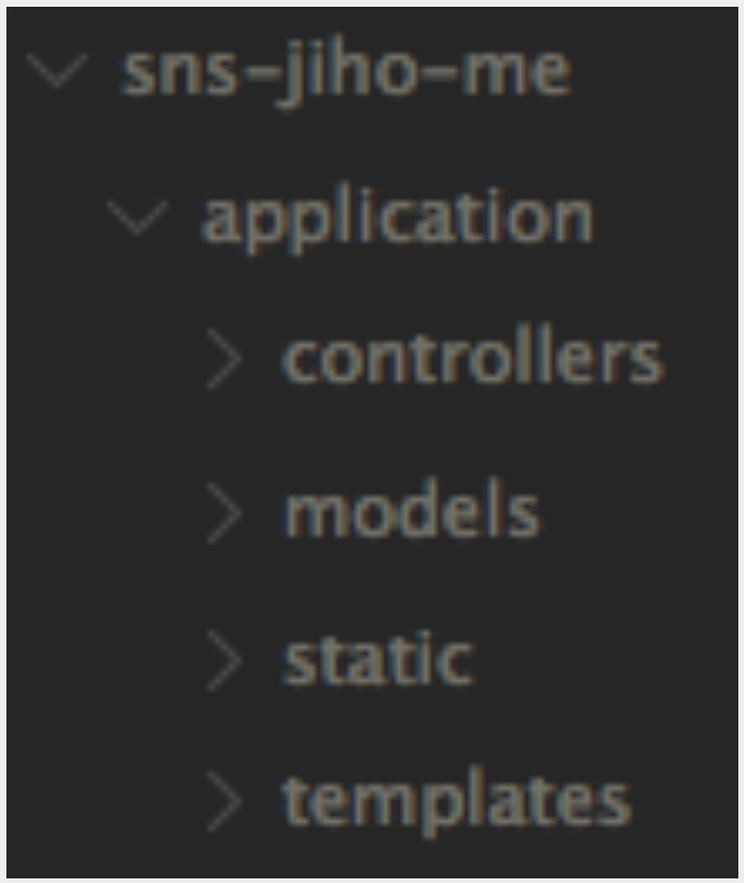
DB Server



MVC


Sublime Text 2 — sns-jiho-me
UNREGISTERED
GROUP 1
GROUP 2
FOLDERS
sns-jiho-me
application
controllers
models
static
templates
__init__.py
config.py
app.yaml
appengine_config.py
manager.py
README.md
runserver.sh
sns-jiho-me
application
controllers
models
static
templates

MVC


sns-jiho-me
application
controllers
models
static
templates

MVC > View

- 사용자의 요청에 대한 결과를 최종적으로 보여주는 역할
- Jinja : 템플릿 HTML 파일 + render_template()

MVC > View

GROUP 1
read.html
GROUP 2
post.py
FOLDERS
sns-jiho-me
application
controllers
__init__.py
auth.py
errorhandlers.py
index.py
post.py
models
static
templates
join.html
layout.html
login.html
post_list.html
read.html
timeline.html
user_list.html
write.html
__init__.py
config.py
lib
migrations
app.yaml
appengine_config.py
manager.py
README.md
runserver.sh

read.html

```
{% extends 'layout.html' %}

{% block head %}
{{ super() }}
<script src="{{ url_for('static', filename='js/read.js') }}"></script>
<link rel="stylesheet" href="{{ url_for('static',filename='css/read.css') }}">



{% if context.is_author %}
<link href="{{ url_for('static', filename='editor/css/font-awesome.min.css') }}" rel="stylesheet" type="text/css">

<link href="{{ url_for('static',filename='editor/css/froala_editor.min.css') }}" rel="stylesheet" type="text/css">

<link href="{{ url_for('static',filename='editor/css/themes/dark.min.css') }}" rel="stylesheet" type="text/css">

<script src="{{ url_for('static', filename='editor/js/froala_editor.min.js') }}"></script>

<script src="{{ url_for('static', filename='js/edit.js') }}"></script>
{% endif %}

{% endblock %}

{% block body %}
<div class="container">
    <div class="row">
        <div class="col-md-8 col-md-offset-2">
            <div class="panel panel-default">

            <div class="panel-body">
                <h3 class="panel-title"><a href="{{
                url_for('timeline',wall_id = context[
```

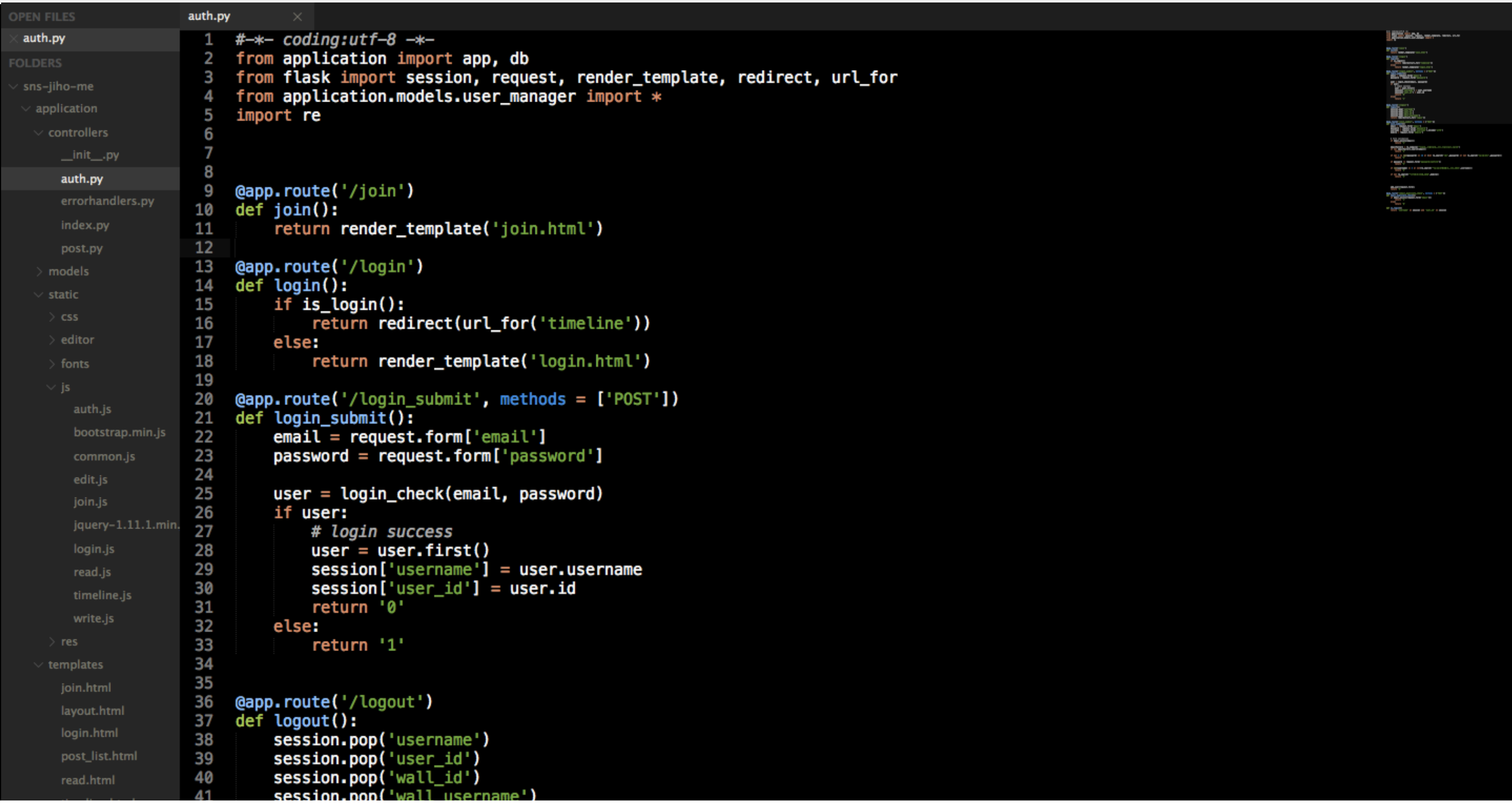
post.py

```
@app.route('/read/<int:post_id>')
def read(post_id) :
    if not is_login():
        return redirect(url_for('login'))

    post = get_post(post_id)
    context = {
        'post':post,
        'is_author':post.user_id == session['user_id']
    }
    return render_template('read.html', context = context)


@app.route('/write')
def write():
    if not is_login():
        return redirect(url_for('login'))

    return render_template('write.html')

@app.route('/post_submit', methods = ['POST'])
def post_submit():
    if not is_login():
        return redirect(url_for('login'))

    postData = {
        'user_id' : session['user_id'],
        'wall_id' : session['wall_id'],
        'is_secret' : '1' if 'is_secret' in request.form
            else '0',
        'body' : request.form['body']
    }
    add_post(postData)


    return redirect(url_for('timeline',wall_id = session['
        wall_id']))

@app.route('/edit',methods=['POST'])
def edit():
    targetPost = get_post(request.form['post_id'])
```

MVC > Controller

- 사용자 요청 분석
- 사용자가 제공한 데이터를 가공해 모델에게 전달
- 사용자가 요청한 데이터를 모델에서 가져와서 뷰로 전달

MVC > Controller



MVC > Model

- 데이터베이스 CRUD
  - Create
  - Read
  - Update
  - Delete
- 컨트롤러는 모델을 통해서 데이터베이스에 접근

MVC > Model

OPEN FILES

post_manager.py

FOLDERS

- sns-jiho-me
  - application
    - controllers
      - __init__.py
      - auth.py
      - errorhandlers.py
      - index.py
      - post.py
    - models
      - __init__.py
      - comment_manager.p
      - post_manager.py
      - schema.py
      - user_manager.py
    - static
      - css
      - editor
      - fonts
      - js
        - auth.js
        - bootstrap.min.js
        - common.js
        - edit.js
        - join.js
        - jquery-1.11.1.min.
        - login.js
        - read.js
        - timeline.js
        - write.js
      - res
    - templates

post_manager.py

```python
#-*- coding:utf-8 -*-
from schema import Post
from application import db


def add_post(data):
    post = Post(
        wall_id = data['wall_id'],
        user_id = data['user_id'],
        body = data['body'],
        is_secret = data['is_secret']
    )

    db.session.add(post)
    db.session.commit()

    return post

def get_post(id):
    return Post.query.get(id)

def get_all_posts():
    return Post.query.order_by(desc(Post.edited_time)).all()


def get_wall_posts(wall_id,start_id = 0,
    post_count = 5):
    start_id = int(start_id)

    if start_id == 0:
        return Post.query.filter(Post.wall_id == wall_id).order_by(db.desc(Post.edited_time)).limit(post_count)
    else:
        return Post.query.filter(Post.wall_id == wall_id, Post.id < start_id).order_by(db.desc(Post.edited_time)).limit(
            post_count)


def delete_post(id):
    db.session.delete(get_post(id))
    db.session.commit()

def edit_post(data):
```

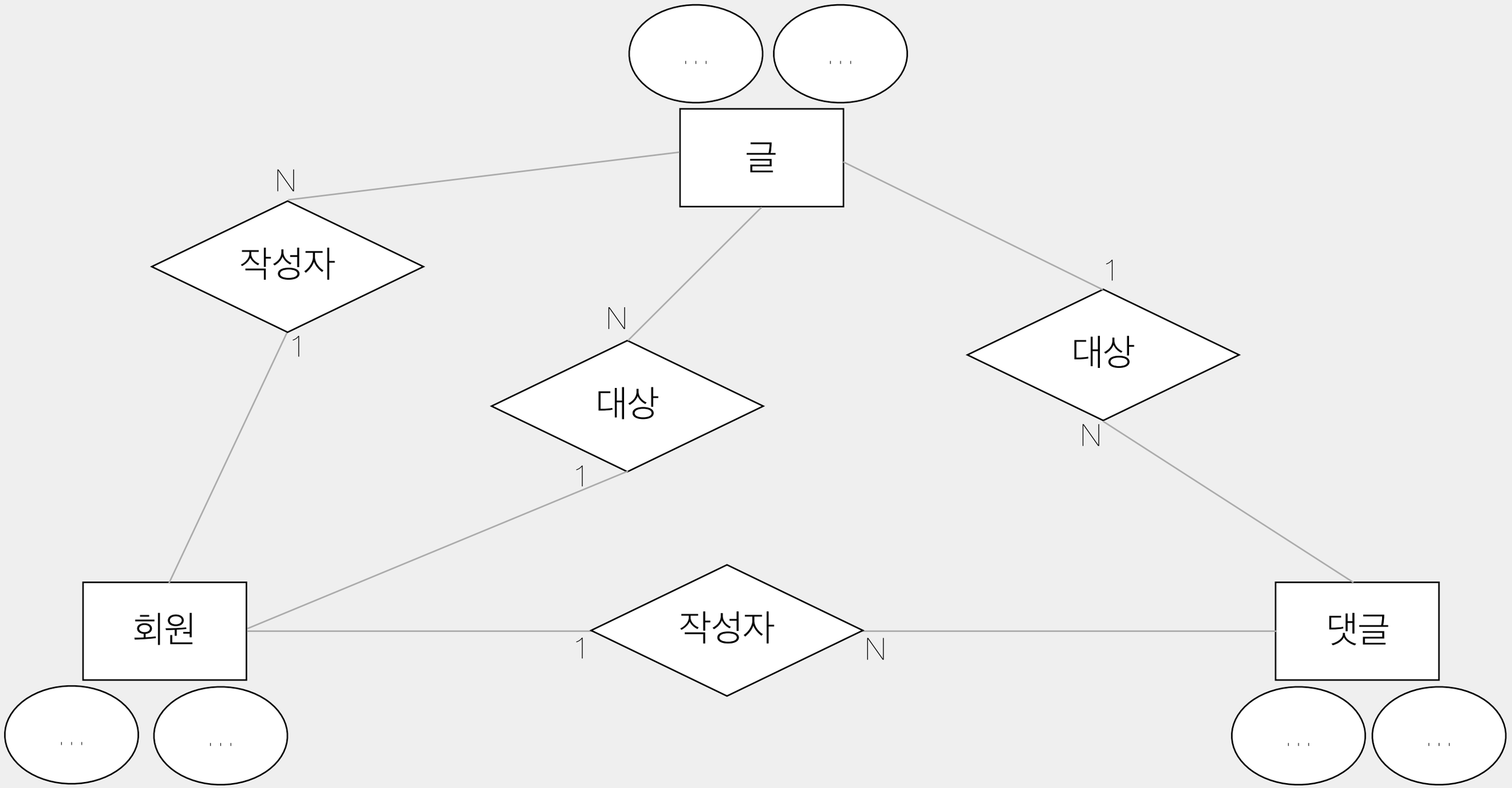
Likelion

Database - ORM

관계형 데이터베이스

- 개체
- 관계
- 속성

관계형 데이터베이스



관계형 데이터베이스

사용자

ID	이름	이메일	...

글

ID	내용	시간	...

댓글

ID	내용	시간	...

관계형 데이터베이스

사용자

ID	이름	이메일	...

글

ID	내용	시간	작성자ID	담벼락ID	...

댓글

ID	내용	시간	글ID	작성자ID	...

Database > CRUD

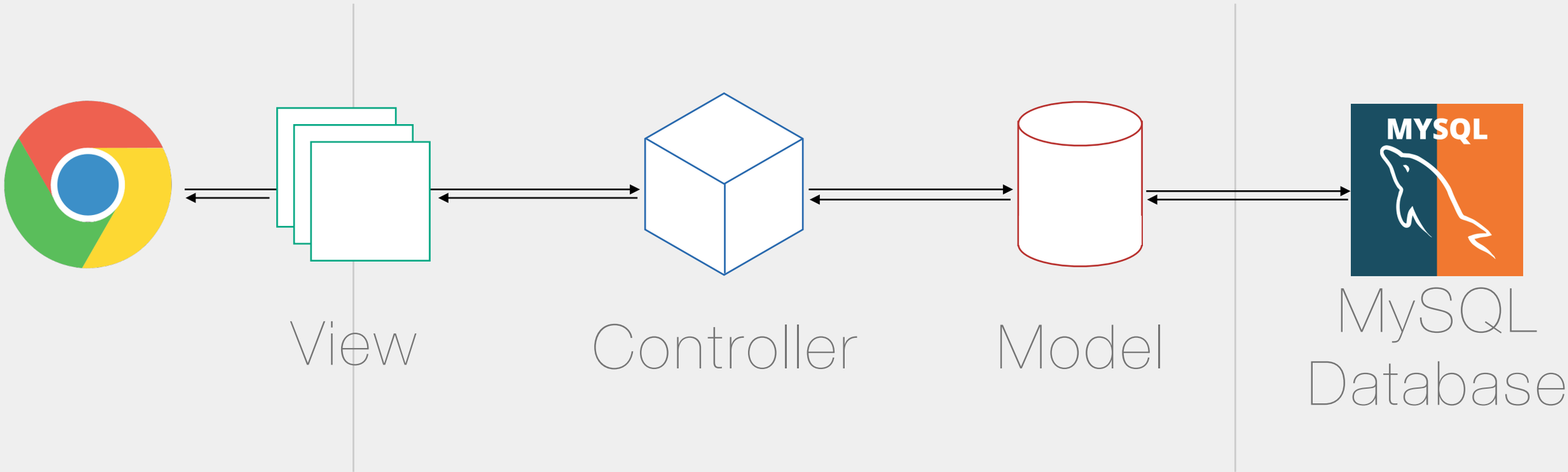
- Create : 데이터베이스의 테이블에 데이터 추가
- Read : 데이터베이스의 테이블에서 데이터를 가져옴
- Update : 데이터베이스 테이블의 데이터 수정
- Delete : 데이터베이스 테이블의 데이터 삭제

Database > CRUD

Client

Web Server

DB Server

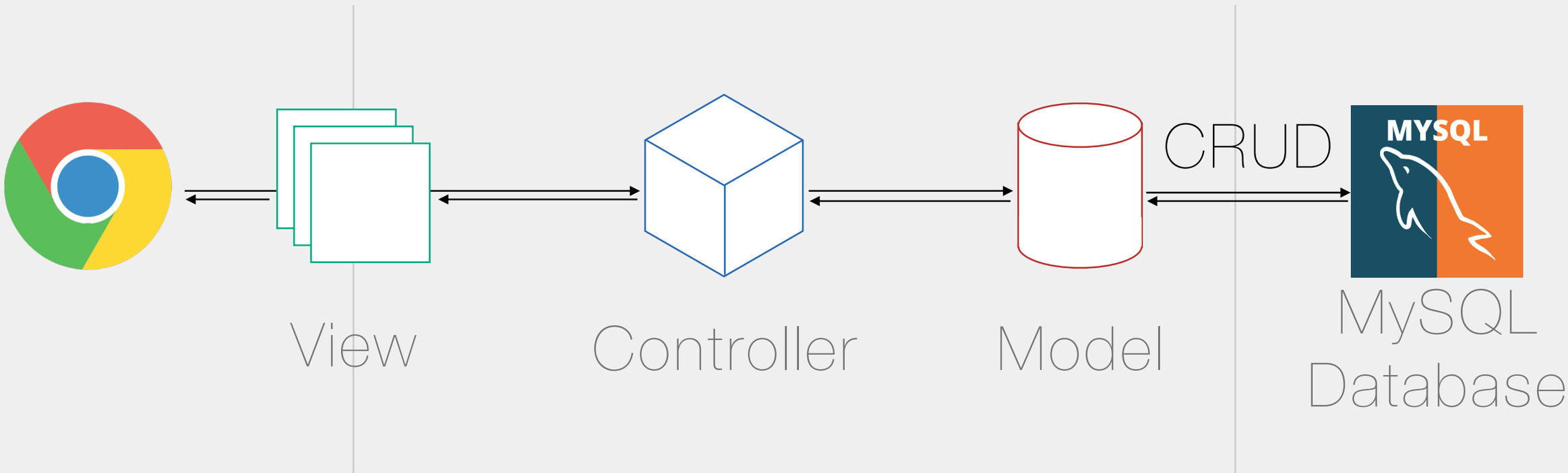


Database > CRUD

Client

Web Server

DB Server



ORM

Object-Relational Mapping

쿼리를 간단하게 수행할 수 있는 방법
CRUD에 해당하는 파이썬 함수 제공

ORM > Schema

```
class Post(db.Model):
    id           = db.Column(db.Integer, primary_key = True)
    user_id      = db.Column(db.Integer, db.ForeignKey('user.id'))
    user         = db.relationship('User',foreign_keys = [user_id]
       #,backref  = db.backref('posts', cascade = 'all, delete-orphan', lazy = 'dynamic')
    )
    wall_id      = db.Column(db.Integer, db.ForeignKey('user.id'))
    wall         = db.relationship('User',foreign_keys = [wall_id],
        backref = db.backref('wall_posts', cascade = 'all, delete-orphan', lazy = 'dynamic'))
    body         = db.Column(db.Text())
    edited_time = db.Column(db.DateTime, default = db.func.now(), onupdate = db.func.now())
    created_time = db.Column(db.DateTime, default = db.func.now())
    is_edited    = db.Column(db.Boolean, default = '0')
    is_secret    = db.Column(db.Boolean, default = '0')
```

ORM > Create

```
user = User(
    email = data['email'],
    username = data['username'],
    gender = data['gender'],
    password = db.func.md5(data['password']),
    mobile = data['mobile'],
    birthday = data['birthday']
)

db.session.add(user)
db.session.commit()
```

ORM > Read

```
Post.query.filter(Post.wall_id == 10).order_by(db.desc(Post.edited_time)).limit(5)
```

ORM > Update

```
targetPost = get_post(data['post_id'])
targetPost.body = data['body']

db.session.commit()
```

ORM > Delete

```
targetPost = get_post(id)
db.session.delete(targetPost)
db.session.commit()
```

...

Web Service Structure

Client

Web Server

DB Server