

## 8. aJax

영진전문대학  
컴퓨터정보계열  
김종율

# 학습내용

- aJax
- JSONP

# aJax

- <http://ko.wikipedia.org/wiki/Ajax>
- Asynchronous JavaScript And XML
  - 비동기 JavaScript와 XML
  - XML: Data 교환 ML
  - JSON & JSONP
- 배경
  - JS는 Client에서 동작하며, user의 Web Browser를 통해 해석되어 실행
  - Server와 Data 송수신을 위해서는 특정 페이지로 이동해야 함
  - 전통적인 Web App에서 Server로 Data를 보내려면 Form태그로 Submit시켜서 처리, 서버에서는 수신 데이터를 처리하고 응답 결과를 Client에게 반환
    - 페이지를 이동하여 Data를 보내야 하기 때문에 사용성이 떨어지고 페이지를 새로 불러오므로 속도가 느림
  - aJax: Page를 이동시키지 않고 Server와 Data 송수신할 수 있는 기법
    - 비동기 & 동기 모드
      - 동기 모드
        - » 요청시 서버에서 응답이 올 때까지 Web App이 정지
      - 비동기 모드
        - » 서버에서 응답이 올 때까지 정지하지 않음, 다른 작업 수행 가능
    - 일반적으로 응답으로 XML 또는 JSON, 그리고 JSONP 형식으로 받음
  - jQuery의 aJax함수 활용

# jQuery의 ajax함수

- `$.ajax(option);`

- `option`:

- json형태 지정

- <http://api.jquery.com/jQuery.ajax/>

```
{
  url: "remote_url",
  type: "GET",
  data: "parameters",
  dataType: "json",
  async: false,
  timeout: millisecond,
  cache: true,
  success: function(data){},
  error: function(xhr,
                    textStatus,
                    errorThrown)
  {}
}
```

option	설명
url	로드할 파일의 서버 주소 또는 파일 경로 타입: 문자열
type	데이터 연동방식 지정 (GET or POST) 타입: 문자열, 생략시 기본값 GET
data	서버로 보낼 데이터 (key=value&key=value) 타입: 문자열
dataType	로드될 데이터의 타입 지정 (text, json, xml, jsonp) 타입: 문자열
async	true - 비동기식 호출, false - 동기식 호출 타입: 불리언, 생략시 기본값 true
timeout	밀리세컨드 단위 응답에 대한 대기 제한시간 타입: 숫자
cache	이전 요청에 대한 캐시저장 여부 (true-사용) 타입: 불리언
success	데이터 읽기 완료시 실행될 함수 타입: 함수
error	데이터 읽기 실패시 실행될 함수 타입: 함수

# 간단 예제

- Echo 예제

- client에서 서버로 Hello송신, 서버에서 수신한 Hello를 Client로 송신

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset='utf-8' />
5   <title>jQuery 예제</title>
6   <style>    </style>
7   <script
8     src="https://code.jquery.com/jquery-3.3.1.min.js"
9     integrity="sha256-FgpCb/KJQlLNfOu91ta32o/NMZxltwRo8QtmkMRdAu8="
10    crossorigin="anonymous">
11  </script>
12  <script>
13
14  </script>
15 </head>
16 <body>
17   <div class="Response"></div>
18   <input type="button" id="ButtonRequest" value="AJAX 시작" />
19 </body>
20 </html>
```

express 프로젝트 폴더내부에서  
ajax\_test폴더내 www 폴더내에  
index.html작성

# 간단 예제

- script 작성

```
<script>
$(document).ready(function(){
    $('#ButtonRequest').click(function(){
        $.ajax({
            url: 'http://localhost:3000/Echo',
            data:{
                myData: "Hello안녕",
            },
            dataType: 'jsonp',
            success: function(data){
                console.log(data);
                $('#.Response').text('서버로 부터의 응답 : ' + data.result);
            },
            error: function(){
                $('#.Response').text('오류가 발생하였습니다. ');
            }
        });
    });
});
</script>
```

body-parser 모듈 설치

npm install body-parser --save

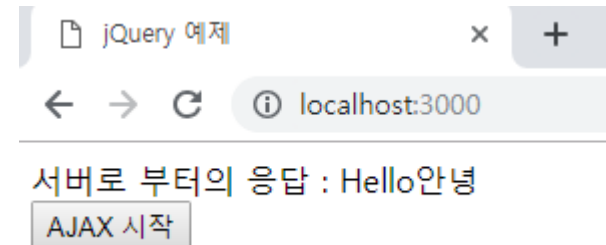
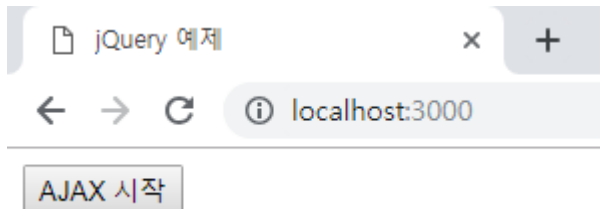
# 간단 예제

- Node.js 서버 소스 작성 ajax\_srv\_test.js

```
1 var express = require('express'); var app = express();
2 var bodyParser = require('body-parser'); var path = require('path');
3
4 // 정적 파일 처리 경로 설정.
5 app.use(express.static(path.join(__dirname, 'ajax_test', 'www')));
6 app.use(bodyParser.json());
7 app.use(bodyParser.urlencoded({extended: true}));
8 // 서비스 메인 페이지 반환.
9 app.get('/', function(req, res) {
10   res.sendFile(path.join(__dirname, 'ajax_test', 'www', 'index.html'));
11 });
12 // Echo서비스
13 app.get('/Echo', function(req, res) {
14   var callbackName = req.query.callback;
15   var myData = req.query.myData;
16   var ResObj = {};
17   ResObj.result = myData;
18   res.header('Content-type', 'application/json');
19   res.header('Charset', 'utf8');
20   res.send(callbackName + '(' + JSON.stringify(ResObj) + ');');
21 });
22
23 var server = app.listen(3000, function() {
24   console.log("Express server has started on port 3000");
25 });
```

# 간단 예제

- 서버 실행 후 client실행
  - node ajax\_srv\_test.js
  - http://localhost:3000/





# aJax로 DB조회 예제

- 준비1: 서버 실행후 MySQL Cli실행
  - `mysql -uroot -p`
- 준비2: 사용할 DB선택
  - `use test;`
- 준비3: table 생성
  - `CREATE TABLE Person( PNAME CHAR(50),AGE INT )  
ENGINE=InnoDB DEFAULT CHARSET=utf8;`
- 준비4: Data 입력
  - `insert into Person values('김영진',23);`
  - `insert into Person values('이영진',21);`
  - `insert into Person values('최영진',22);`
  - `insert into Person values('박영진',23);`

# aJax로 DB조회 예제

- client 소스 작성 (html)

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <meta charset='utf-8' />
5     <title>jQuery 예제</title>
6     <style>    </style>
7     <script
8         src="https://code.jquery.com/jquery-3.3.1.min.js"
9         integrity="sha256-FgpCb/KJQlLNfOu91ta32o/NMZxltwRo8QtmkMRdAu8="
10        crossorigin="anonymous">
11    </script>
12    <script>
13
14    </script>
15 </head>
16 <body>
17     <input type="button" class="BtnRequest" value="데이터 불러오기" />
18     <div class="output"></div>
19 </body>
20 </html>
```

express 프로젝트 폴더내부에서  
ajaxdb\_test폴더내 www 폴더내에  
index.html작성

# ajax로 DB조회 예제

- script 작성

```
12 <script>
13 $(document).ready(function(){
14     $('.BtnRequest').click(function(){
15         var output = $('.output').empty();
16         $.ajax({
17             url: 'http://localhost:3000/ajaxDB',
18             data:{ },
19             dataType: 'jsonp',
20             success: function(responseData){
21                 var cnt = responseData.data.length;
22                 for(var i = 0; i < cnt; i++){
23                     $('<p>이름 : ' + responseData.data[i].PNAME + '</p>').appendTo(output);
24                     $('<p>나이 : ' + responseData.data[i].AGE + '</p>').appendTo(output);
25                 }
26             },
27             error: function(){
28                 $('<p>오류가 발생하였습니다.</p>').appendTo(output);
29             }
30         });
31     });
32 });
```

# ajax로 DB조회 예제

- Node.js 서버 소스 작성(ajaxdb\_srv\_test.js)

```
1 var express = require('express'); var app = express();
2 var bodyParser = require('body-parser'); var path = require('path');
3 var mysql = require('mysql');
4 var connection = mysql.createConnection({
5   host      : 'localhost',
6   user      : 'root',
7   password  : 'node',
8   port      : 3306,
9   database  : 'test'
10 });
11
12 // 정적 파일 처리 경로 설정.
13 app.use(express.static(path.join(__dirname, 'ajaxdb_test', 'www')));
14 app.use(bodyParser.json());
15 app.use(bodyParser.urlencoded({extended: true}));
16 // 서비스 메인 페이지 반환.
17 app.get('/', function(req, res) {
18   res.sendFile(path.join(__dirname, 'ajaxdb_test', 'www', 'index.html'));
19 });
```

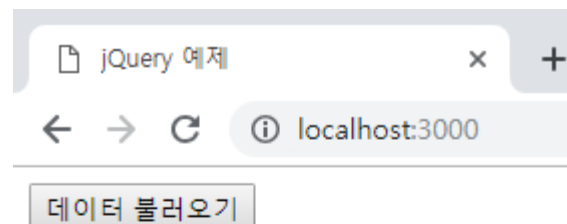
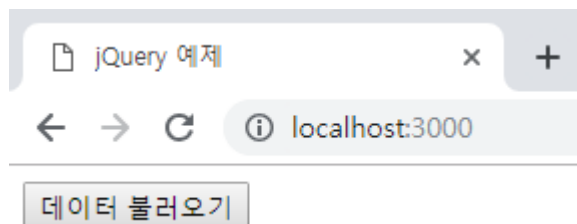
# aJax로 DB조회 예제

```
20 // ajaxDB 서비스
21 app.get('/ajaxDB', function(req, res) {
22     console.log('params: ' + JSON.stringify(req.params));
23     console.log('body: ' + JSON.stringify(req.body));
24     console.log('query: ' + JSON.stringify(req.query));
25     connection.query('SELECT * FROM person', function(err, results, fields) {
26         var callbackName = req.query.callback;
27         var ResObj = {};
28         res.header('Content-type', 'application/json');
29         res.header('Charset', 'utf8');
30         console.log(results);
31         if(err) {
32             ResObj.result = 'failed';
33             res.send(callbackName + '(' + JSON.stringify(ResObj) + ');');
34             //throw err;
35         } else if(!results.length) {
36             ResObj.result = 'wrong';
37             res.send(callbackName + '(' + JSON.stringify(ResObj) + ');');
38         } else {
```

# aJax로 DB조회 예제

```
39     ResObj.result = 'success';
40     ResObj.data = results;
41     res.send(
42         callbackName + '(' + JSON.stringify(ResObj) + ');'
43     );
44 }
45 });
46 });
47
48 var server = app.listen(3000, function() {
49     console.log("Express server has started on port 3000");
50 });
```

# ajax로 DB조회 예제



# 간단 로그인구현

- 준비1: 서버 실행후 MySQL Cli실행
  - `mysql -uroot -p`
- 준비2: 사용할 DB선택
  - `use test;`
- 준비3: table 생성
  - `CREATE TABLE member(  
    mno INT AUTO_INCREMENT PRIMARY KEY,  
    mid VARCHAR(50) NOT NULL,  
    mpw VARCHAR(80) NOT NULL,  
    mname VARCHAR(100) NOT NULL  
); ENGINE=InnoDB DEFAULT CHARSET=utf8;`
- 준비4: Data 입력
  - `INSERT INTO member (mid,mpw,mname) VALUES ('test', 't123', 'test_name');`



# 간단 로그인구현

```
1 <!DOCTYPE html>
2 <html lang="ko">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=no">
6   <meta http-equiv="X-UA-Compatible" content="ie=edge">
7   <style>
8     #Login{display:none;}
9   </style>
10  <script
11    src="https://code.jquery.com/jquery-3.3.1.min.js"
12    integrity="sha256-FgpCb/KJQlLNfOu91ta32o/NMZxltwRo8QtmkMRdAu8="
13    crossorigin="anonymous">
14  </script>
15  <script>
16
17  </script>
18  <title>TEST</title>
19 </head>
```

express 프로젝트 폴더내부에서  
login\_test폴더내 www 폴더내에  
index.html작성

# 간단 로그인구현

```
20 <body>
21   <!-- app wrapper -->
22   <div id="AppWrap">
23
24     <div id="BeforeLogin">
25       <img src='https://www.yju.ac.kr/sites/kr/images/img_symbol_mark.png'
26         width='300px' height='300px' /> <br/>
27       <button id="ToLogin">인트로화면에서로그인화면으로가기</button>
28     </div>
29
30   <!-- login page -->
31   <div id="Login">
32     <h1>My SNS Service</h1>
33     <br/>
34     <h3>로그인해주세요</h3>
35     <p>아이디</p>
36     <input type="text" id="loginTxtID" maxlength="50"/>
37     <p>비밀번호</p>
38     <input type="password" id="loginTxtPw" maxlength="50"/>
39     <br/><br/>
40     <button id="loginBtnLogin">로그인</button>
```

# 간단 로그인구현

```
40     <button id="loginBtnLogin">로그인</button>
41     <button id="loginBtnJoin">회원가입</button>
42 </div>
43 </div>
44 </body>
45 </html>
```

# 간단 로그인구현

- script 작성

```
16 $(document).ready(function() {
17     $('#ToLogin').click(function (){
18         $('#BeforeLogin').hide();
19         $('#Login').show();
20     });
21     $('#loginBtnLogin').click(function (){
22         var InputedId = $('#loginTxtID').val();
23         var InputedPw = $('#loginTxtPw').val();
24         if(!InputedId){
25             window.alert('아이디를 입력하세요!');
26             return false;
27         }else if(!InputedPw){
28             window.alert('비밀번호를 입력하세요!');
29             return false;
30         }
31         $.ajax({
32             type:"POST",
33             url:"http://localhost:3000/login",
34             data:{
35                 id: InputedId,
```

# 간단 로그인구현

```
36 pw: InputedPw
37 },
38 dataType: 'json',
39 success: function(responseData){
40     console.log(responseData);
41     console.log(JSON.stringify(responseData));
42     //로그인 성공
43     if(responseData.result == "success"){
44         alert('안녕하세요, <b>' + responseData.data[0].mname + '</b>님. ');
45         $('#AppWrap').append("<div>안녕하세요, <h1>"+
46             responseData.data[0].mname+"</h1>님. </div>");
47     }
48     //로그인 실패 - 잘못된 값
49     else if(responseData.result == "wrong"){
50         window.alert('잘못된 아이디나 비밀번호를 입력하셨습니다. ');
51     }
52     //로그인 실패 - 오류
53     else{
54         window.alert('오류가 발생하였습니다. ');
55     }
56 },
```

```
57     error: function(){
58         window.alert('오류가 발생하였습니다. ');
59     }
60 });
61 });
62 });
```

# 간단 로그인구현

- Node.js 서버 소스 작성(loginsrv\_test.js)

```
1 var express = require('express'); var app = express();
2 var path = require('path'); var bodyParser = require('body-parser');
3 var mysql = require('mysql');
4 var connection = mysql.createConnection({
5   host      : 'localhost',
6   user      : 'root',
7   password  : 'node',
8   port      : 3306,
9   database  : 'test'
10 });
11
12 // 정적 파일 처리 경로 설정.
13 app.use(express.static(path.join(__dirname, 'login_test', 'www')));
14 app.use(bodyParser.json());
15 app.use(bodyParser.urlencoded({extended: true}));
16
```

# 간단 로그인구현

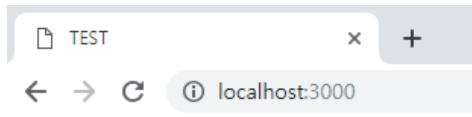
```
17 // 서비스 메인 페이지 반환.
18 app.get('/', function(req, res) {
19     res.sendFile(path.join(__dirname, 'login_test', 'www', 'index.html'));
20 });
21 // 로그인
22 app.post('/login', function(req, res) {
23     var inputId = req.body.id;
24     var inputPw = req.body.pw;
25
26     var query = "SELECT * FROM member WHERE mid = ? AND mpw = ?";
27     var test = [inputId, inputPw];
28
29     connection.query(query, test, function(error, rows, result) {
30         var ResObj = {};
31         res.header('Content-type', 'application/json');
32         res.header('Charset', 'utf8');
33         var resData = 'failed';
34         if(error) {
```

# 간단 로그인 구현

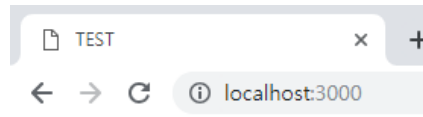
```
35     ResObj.result = resData;
36     res.send(JSON.stringify(ResObj));
37   } else if(!rows.length) {
38     resData = 'wrong';
39     ResObj.result = resData;
40     res.send(JSON.stringify(ResObj));
41   } else {
42     resData = 'success';
43     ResObj.result = resData;
44     ResObj.data = rows;
45     res.send(JSON.stringify(ResObj));
46   }
47   });
48 });
49
50 var server = app.listen(3000, function() {
51   console.log("Express server has started on port 3000");
52 });
```



# 간단 로그인구현



인트로화면에서로그인화면으로가기



## My SNS Service

로그인해주세요

아이디

test

비밀번호

....

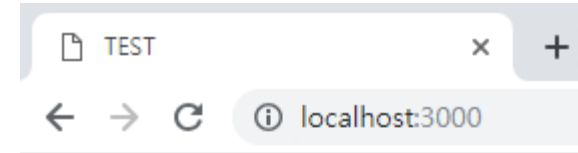
로그인

회원가입

localhost:3000 내용:

안녕하세요, <b>test\_name</b>님.

확인



## My SNS Service

로그인해주세요

아이디

test

비밀번호

....

로그인

회원가입

안녕하세요,

**test\_name**

님.