1. **Ajax : 서버와 데이터 교환을 하는 기술.**

<http://api.jquery.com/category/ajax/> ajax관련된 API 정보

웹페이지의 특정부분(한파트)를 업데이트하고 서버와 데이터를 교환하는 기술.

우리가 기존의 웹페이지의 내용을 변경하기 위해서는 웹페이지에 떠있는 페이지를 다시 로딩해야 했었다.

그런데 Ajax를 이용하게 되면 특정부분의 영역에 있는 데이터 수정시 전체 페이지를 로딩하지 않고 서버와 데이터 교환이 가능하다

샘플을 보면 기존에 있었던 텍스트 대신 바꿔 치기 한다.

전체가 로딩되지 않고 일부분만 텍스트가 변경되었다.

Ajax는 웹페이지의 전체내용을 다시 불러들이지 않고 웹페이지의 일부분을 배경에서 서버와 데이터 교환이 일어나 원하는 부분에 뿌려 줄 수 있는 기술이다.

jQuery에서는 Ajax와 관련하여 몇 개의 메소드를 제공하고 있다.

jQuery Ajax메소드를 사용하게 되면 외부데이터를 로딩해서 보여질 수 있다.

jQuery를 이용하지 않으면 Ajax이용이 어려울 수 있으나 jQuery를 이용하면 쉬운 코딩이 될 수 있다.

1. Http Request : GET vs. POST

$.get(URL, callback);

$.post(URL, data, callback);

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Insert title here</title>

<script src=*"jquery-1.12.3.js"*></script>

<script>

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

$("button").click(**function** () {

$.get("demo\_test.jsp?id=aa",**function**(data, status){

alert("data : " + data + "\nstatus : "+status);

});

});

});

</script>

</head>

<body>

<button>aa 아이디 중복 체크</button>

</body>

</html>

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=UTF-8"*>

<title>Insert title here</title>

<script src=*"jquery-1.12.3.js"*></script>

<script>

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

$("button").click(**function** () {

$.post("demo\_test.jsp",{id:"aa"},

**function**(data, status){

alert("data : " + data + "\nstatus : "+status);

});

});

});

</script>

</head>

<body>

<button>aa 아이디 중복 체크</button>

</body>

</html>

가장 기본적인 API

jQery.ajax([settings])

jQery.ajax(url[, settings])

* Setting은 Ajax 통신을 위한 옵션을 담고 있는 객체가 들어간다.주요한 옵션은 다음과 같은 것들이 들어갈 수 있다.
  + url : 요청 자원을 호출.
  + Data : 서버로 데이터를 전송할 때 사용
  + datatype : 서버측에서 전송한 데이터를 어떤 형식의 데이터로 해석할 것인가를 지정한다. 값으로 올 수 있는 것은 xml, json, script, html. 형식을 지정하지 않으면 jQuery가 알아서 판단
  + success : 성공했을 때 호출할 콜백함수를 지정
  + function(plainObject data, String testStatus, jqXHR jqXHR)

idCon.jsp

<%@page import=*"com.ch.ex.dao.MembersDao"*%>

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=UTF-8"*>

<title>Insert title here</title>

</head>

<body>

<%

String id = request.getParameter("id");

MembersDao dao = MembersDao.getInstance();

**int** result = dao.idConfirm(id);

**if**(result == 0) {

%>

중복된 ID입니다. 다른 ID를 쓰세요

<% } **else** { %>

사용가능한 ID입니다

<% } %>

</body>

</html>

aGet.html

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Insert title here</title>

<script src=*"jquery-1.12.3.min.js"*></script>

<script>

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

$('button').click(**function**(){

**var** id = $('#id').val();

// Ajax 이용 - get, post, ajax함수

$.get('idCon.jsp?id='+id, **function**(data, status){

$('span').html(data);

});

});

/\* $('span').click(function(){

$('span').html('<button>ID 중복 체크</button>');

}); \*/

});

</script>

</head>

<body>

<h1>ID 중복 체크</h1>

ID : <input type=*"text"* name=*"id"* id=*"id"*>

<button>ID 중복 체크</button><span></span>

</body>

</html>

aPost.html

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Insert title here</title>

<script src=*"jquery-1.12.3.min.js"*></script>

<script>

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

$('button').click(**function**(){

$.post("idCon.jsp", {id : $('#id').val()},

**function**(data, status){

$('span').html(data);

});

});

});

</script>

</head>

<body>

<h1>ID 중복 체크</h1>

ID : <input type=*"text"* name=*"id"* id=*"id"*>

<span><button>ID 중복 체크</button></span>

</body>

</html>

aAjax.html

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Insert title here</title>

<script src=*"jquery-1.12.3.min.js"*></script>

<script>

/\* $.ajax(json형식으로 otion 설정)

ex) $.ajax({url:'', type:'get', data:~ , success:function(data,status){} });

1. url : 요청 자원(서버)를 호출,

일반적으로 호출된 자원에서 json형식으로 데이터 return 해준다

2. type : 요청처리방식

3. data: 요청시, 함께 전달되는 데이터

ex. $("form").serialize()

form안에 있는 데이터를 query로 변경하여 전달

4. dataType : 요청 후, 응답의 결과로 반환되는 데이터의 종류를 선언

ex) xml, html, json, text

5. success : 성공했을 때, 처리할 내용, 함수로 나타냄.

ex) function(data, status){}

6. error : 응답이 에러가 났을 때, 에러 상태에 대한 코드를 반환

ex) error : function(code){

code:에러에 대한 code 내용 전달...

}

\*/

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

$('button').click(**function**(){

$.ajax({url : 'idCon.jsp',

type : 'get',

datatype : ‘html’,

data : "id="+$("#id").val(),

success : **function**(data, status){

$('span').html(data);}

});

});

});

</script>

</head>

<body>

<h1>ID 중복 체크</h1>

ID : <input type=*"text"* name=*"id"* id=*"id"*>

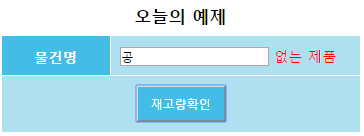
<span><button>ID 중복 체크</button></span>

</body>

</html>

<오늘의 예제>





**CREATE** SEQUENCE PRODUCT\_SEQ START **WITH** 1

INCREMENT **BY** 1

MAXVALUE 9999;

**CREATE** **TABLE** PRODUCT(

PNO NUMBER(4) **PRIMARY** **KEY**,

PNAME VARCHAR2(100) **NOT** **NULL**,

PS NUMBER **DEFAULT** 0

);

**INSERT** **INTO** PRODUCT **VALUES**(PRODUCT\_SEQ.NEXTVAL, '공책', 3);

**INSERT** **INTO** PRODUCT **VALUES**(PRODUCT\_SEQ.NEXTVAL, '볼펜', 10);

**INSERT** **INTO** PRODUCT **VALUES**(PRODUCT\_SEQ.NEXTVAL, 'A', 10);

**SELECT** \* **FROM** PRODUCT **WHERE** PNAME = '공책';

**COMMIT**;

**SELECT** \* **FROM** PRODUCT;

**public** **class** ProductDTO {

**private** **int** pno;

**private** String pname;

**private** **int** ps;

**public** ProductDTO() {}

**public** ProductDTO(**int** pno, String pname, **int** ps) {

**super**();

**this**.pno = pno;

**this**.pname = pname;

**this**.ps = ps;

}

**public** **int** getPno() {**return** pno;}

**public** **void** setPno(**int** pno) { **this**.pno = pno; }

**public** String getPname() { **return** pname; }

**public** **void** setPname(String pname) { **this**.pname = pname; }

**public** **int** getPs() { **return** ps; }

**public** **void** setPs(**int** ps) { **this**.ps = ps; }

}

**public** **class** ProductDAO {

**public** ProductDAO(){

**try** {

Class.*forName*("oracle.jdbc.driver.OracleDriver");

} **catch** (ClassNotFoundException e) {

e.printStackTrace();

}

}

**private** Connection getConnection(){

Connection conn= **null**;

String url = "jdbc:oracle:thin:@localhost:1521:xe";

String uid = "scott";

String upw = "tiger";

**try** {

conn = DriverManager.*getConnection*(url, uid, upw);

} **catch** (SQLException e) {

e.printStackTrace();

}

**return** conn;

}

**public** ProductDTO getProduct(String searchName){

ProductDTO dto = **null**;

Connection conn = **null**;

PreparedStatement pstmt = **null**;

ResultSet rs = **null**;

String query = "select \* from product where pName = ?";

**try** {

conn = getConnection();

pstmt = conn.prepareStatement(query);

pstmt.setString(1, searchName);

rs = pstmt.executeQuery();

**if**(rs.next()){

**int** pNo = rs.getInt("pNo");

String pName = rs.getString("pName");

**int** ps = rs.getInt("ps");

dto = **new** ProductDTO(pNo, pName, ps);

System.***out***.println("찾아서 dto에 넣었음");

}

}**catch**(SQLException e){

e.printStackTrace();

}**finally**{

**try**{

**if**(rs!=**null**) rs.close();

**if**(pstmt!=**null**)pstmt.close();

**if**(conn!=**null**) conn.close();

}**catch**(Exception e){e.printStackTrace();}

}

**return** dto;

}

}

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=UTF-8"*>

<title>Insert title here</title>

<style type=*"text/css"*>

*#wrap* { width:*400px*; margin:*10px auto*;}

**table** {width:*100%*;}

**table** **th,** **input**[type=button] { width:*100px*; background-color: *#43BDE6*; color:*white*; padding: *10px*; text-align: *center*;}

**table** **td** { background-color: *#AFE0F0*; padding: *10px*; text-align: *center*; }

**table** **caption**{padding: *5px 5px 10px*; font-size: *1.2em*; font-weight: *bold*;}

**input**[type=text]*:FOCUS* { background-color: *lightyellow*;}

*#msg*{color:*red*; text-align: *center*;}

</style>

<script src=*"http://code.jquery.com/jquery-1.10.2.js"*></script>

<script>

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

$('input[type=button]').click(**function**() {

$.ajax({

url : 'homework0530\_loadData.jsp',

data : "searchName="+$('input[name=searchName]').val(),

dataType : "text",

success : **function**(msg){

$('#msg').html(msg);

}

});

});

});

</script>

</head>

<body>

<div id=*"wrap"*>

<table>

<caption>오늘의 예제</caption>

<tr><th>물건명</th><td style="text-align:*left*;"><input type=*"text"* name=*"searchName"*><span id=*"msg"*></span></td></tr>

<tr><td colspan=*"2"*><input type=*"button"* value=*"재고량확인"*></td></tr>

</table>

</div>

</body>

</html>

<%@page import=*"jspexp.z05\_homework0518.ProductDTO"*%>

<%@page import=*"jspexp.z05\_homework0518.ProductDAO"*%>

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=UTF-8"*>

</head>

<body>

<%

String searchName = request.getParameter("searchName");

ProductDAO dao = **new** ProductDAO();

ProductDTO dto = dao.getProduct(searchName);

**if**(dto!=**null**){

out.println(dto.getPs()+"개 재고");

}**else**{

out.println("없는 제품");

}

%>

</body>

</html>

<오늘의 예제>

**CREATE** SEQUENCE EMP\_SEQ START **WITH** 7935

INCREMENT **BY** 1

MAXVALUE 999999;

**select** \* **from** emp;

**insert** **into** emp **values** (emp\_seq.nextval, '홍길동', 'superman', 7934, '17-05-10',5000,500,40);

rollback;

**select** max(empno)+1 **from** emp;

**select** empno, ename **from** emp;

**select** deptno, dname **from** dept **order** **by** dept.DEPTNO;

**update** emp **set** ename='정지훈', job='연예인', mgr=7937, hiredate='2017/06/03', sal = 7000, comm=10, deptno=10 **where** empno=7941;

**select** \* **from** emp **where** empno=7934;

**delete** **from** emp **where** empno=7949;

**import** java.sql.Date;

**public** **class** Emp {

**private** **int** empno;

**private** String ename;

**private** String job;

**private** **int** mgr;

**private** Date hiredate;

**private** **double** sal;

**private** **double** comm;

**private** **int** deptno;

**public** Emp() { }

**public** Emp(**int** empno, String ename, String job, **int** mgr, Date hiredate, **double** sal, **double** comm, **int** deptno) {

**super**();

**this**.empno = empno;

**this**.ename = ename;

**this**.job = job;

**this**.mgr = mgr;

**this**.hiredate = hiredate;

**this**.sal = sal;

**this**.comm = comm;

**this**.deptno = deptno;

}

**public** **int** getEmpno() { **return** empno; }

**public** **void** setEmpno(**int** empno) { **this**.empno = empno; }

**public** String getEname() { **return** ename; }

**public** **void** setEname(String ename) { **this**.ename = ename; }

**public** String getJob() { **return** job; }

**public** **void** setJob(String job) { **this**.job = job; }

**public** **int** getMgr() { **return** mgr; }

**public** **void** setMgr(**int** mgr) { **this**.mgr = mgr; }

**public** Date getHiredate() { **return** hiredate; }

**public** **void** setHiredate(Date hiredate) { **this**.hiredate = hiredate;}

**public** **double** getSal() {**return** sal; }

**public** **void** setSal(**double** sal) { **this**.sal = sal; }

**public** **double** getComm() { **return** comm; }

**public** **void** setComm(**double** comm) { **this**.comm = comm; }

**public** **int** getDeptno() { **return** deptno; }

**public** **void** setDeptno(**int** deptno) { **this**.deptno = deptno; }

}

**import** java.sql.\*;

**import** java.util.ArrayList;

**import** jspexp.z02\_dto.Dept;

**import** jspexp.z02\_dto.Emp;

**public** **class** EmpDAO {

**public** **static** **final** **int** ***SUCCESS*** = 1;

**public** **static** **final** **int** ***FAIL*** = 0;

**public** EmpDAO(){

**try** {

Class.*forName*("oracle.jdbc.driver.OracleDriver");

} **catch** (ClassNotFoundException e) {

e.printStackTrace();

}

}

**private** Connection setConn() **throws** ClassNotFoundException, SQLException{

String conInfo="jdbc:oracle:thin:@localhost:1521:xe";

Connection con = DriverManager.*getConnection*(conInfo, "scott", "tiger");

System.***out***.println("접속성공!!");

**return** con;

}

**public** ArrayList<Emp> empList(){

ArrayList<Emp> list = **new** ArrayList<Emp>();

Connection con=**null**;

Statement stmt=**null**;

ResultSet rs=**null**;

**try** {

con = setConn();

stmt=con.createStatement();

String sql="SELECT \* FROM EMP ORDER BY EMPNO DESC";

rs = stmt.executeQuery(sql);

Emp emp=**null**;

**while**(rs.next()){

emp = **new** Emp();

emp.setEmpno(rs.getInt("empno"));

emp.setEname(rs.getString("ename"));

emp.setJob(rs.getString("job"));

emp.setMgr(rs.getInt("mgr"));

emp.setHiredate(rs.getDate("hiredate"));

emp.setSal(rs.getDouble("sal"));

emp.setComm(rs.getDouble("comm"));

emp.setDeptno(rs.getInt("deptno"));

list.add(emp);

}

} **catch** (ClassNotFoundException e) {

e.printStackTrace();

} **catch** (SQLException e) {

e.printStackTrace();

}**finally**{

**try** {

**if**(rs!=**null**) rs.close();

**if**(stmt!=**null**) stmt.close();

**if**(con!=**null**) con.close();

} **catch** (SQLException e) {

e.printStackTrace();

}

}

**return** list;

}

**public** ArrayList<Emp> searchPrepared(Emp sch){

ArrayList<Emp> list = **new** ArrayList<Emp>();

Connection con=**null**;

Statement stmt=**null**;

PreparedStatement pstmt=**null**;

ResultSet rs=**null**;

**try** {

con = setConn();

/\*String sql="SELECT \* "

+ "FROM EMP "

+ "WHERE ENAME LIKE '%'||?||'%' "

+ "AND JOB LIKE '%'||?||'%' ";\*/

String sql="SELECT \* "

+ "FROM EMP "

+ "WHERE ENAME LIKE ? "

+ "AND JOB LIKE ? ORDER BY EMPNO DESC";

pstmt = con.prepareStatement(sql);

//pstmt.setString(1, sch.getEname());

//pstmt.setString(2, sch.getJob());

pstmt.setString(1, "%"+sch.getEname()+"%");

pstmt.setString(2, "%"+sch.getJob()+"%");

System.***out***.println(sql);

rs = pstmt.executeQuery();

Emp emp=**null**;

**while**(rs.next()){

emp = **new** Emp();

emp.setEmpno(rs.getInt("empno"));

emp.setEname(rs.getString("ename"));

emp.setJob(rs.getString("job"));

emp.setMgr(rs.getInt("mgr"));

emp.setHiredate(rs.getDate("hiredate"));

emp.setSal(rs.getDouble("sal"));

emp.setComm(rs.getDouble("comm"));

emp.setDeptno(rs.getInt("deptno"));

list.add(emp);

}

} **catch** (ClassNotFoundException e) {

e.printStackTrace();

} **catch** (SQLException e) {

e.printStackTrace();

}**finally**{

**try** {

**if**(rs!=**null**){rs.close();}

**if**(stmt!=**null**){stmt.close();}

**if**(con!=**null**){con.close();}

} **catch** (SQLException e) {

e.printStackTrace();

}

}

**return** list;

}

**public** **int** insertEmp(Emp emp){

**int** result = ***FAIL***;

Connection con=**null**;

PreparedStatement pstmt=**null**;

String sql1 = "INSERT INTO EMP VALUES (EMP\_SEQ.NEXTVAL, ?,?,?,?,?,?,?)";

String sql2 = "INSERT INTO EMP VALUES (EMP\_SEQ.NEXTVAL, ?,?,null,?,?,?,?)";

//insert into emp values (emp\_seq.nextval, '홍길동', 'superman', 7934, '17-05-10',5000,500,40);

**try** {

con = setConn();

con.setAutoCommit(**false**);

**if**(emp.getMgr()!=0){

pstmt = con.prepareStatement(sql1);

pstmt.setString(1, emp.getEname());

pstmt.setString(2, emp.getJob());

pstmt.setInt(3, emp.getMgr());

pstmt.setDate(4, emp.getHiredate());

pstmt.setDouble(5, emp.getSal());

pstmt.setDouble(6, emp.getComm());

pstmt.setInt(7, emp.getDeptno());

}**else**{

pstmt = con.prepareStatement(sql2);

pstmt.setString(1, emp.getEname());

pstmt.setString(2, emp.getJob());

pstmt.setDate(3, emp.getHiredate());

pstmt.setDouble(4, emp.getSal());

pstmt.setDouble(5, emp.getComm());

pstmt.setInt(6, emp.getDeptno());

}

result = pstmt.executeUpdate();

**if**(result>***FAIL***){

System.***out***.println("사원입력성공");

con.commit();

}**else**{

System.***out***.println("입력실패");

con.rollback();

}

} **catch** (ClassNotFoundException e) {

e.printStackTrace();

} **catch** (SQLException e) {

e.printStackTrace();

}**finally**{

**try** {

**if**(pstmt!=**null**) pstmt.close();

**if**(con!=**null**) con.close();

} **catch** (SQLException e) {

e.printStackTrace();

}

}

**return** result;

}

**public** **int** empno(){

String sql = "select max(empno)+1 from emp";

**int** empno=0;

Connection con=**null**;

Statement stmt=**null**;

PreparedStatement pstmt=**null**;

ResultSet rs=**null**;

**try** {

con = setConn();

stmt = con.createStatement();

rs = stmt.executeQuery(sql);

rs.next();

empno = rs.getInt(1);

} **catch** (ClassNotFoundException e) {

e.printStackTrace();

} **catch** (SQLException e) {

e.printStackTrace();

} **finally**{

**try** {

**if**(rs!=**null**) rs.close();

**if**(pstmt!=**null**) pstmt.close();

**if**(con!=**null**) con.close();

} **catch** (SQLException e) {

e.printStackTrace();

}

}

**return** empno;

}

**public** ArrayList<Emp> getMgr(){

String sql = "select empno, ename from emp";

ArrayList<Emp> list = **new** ArrayList<Emp>();

Connection con=**null**;

Statement stmt=**null**;

PreparedStatement pstmt=**null**;

ResultSet rs=**null**;

**try** {

con = setConn();

stmt = con.createStatement();

rs = stmt.executeQuery(sql);

**while**(rs.next()){

**int** empno = rs.getInt("empno");

String ename = rs.getString("ename");

list.add(**new** Emp(empno, ename, **null**, 0, **null**, 0, 0, 0));

}

} **catch** (ClassNotFoundException e) {

e.printStackTrace();

} **catch** (SQLException e) {

e.printStackTrace();

} **finally**{

**try** {

**if**(rs!=**null**) rs.close();

**if**(pstmt!=**null**) pstmt.close();

**if**(con!=**null**) con.close();

} **catch** (SQLException e) {

e.printStackTrace();

}

}

**return** list;

}

**public** ArrayList<Emp> getMgr(**int** empnoNot){

String sql = "select empno, ename from emp where empno!=?";

ArrayList<Emp> list = **new** ArrayList<Emp>();

Connection con=**null**;

PreparedStatement pstmt=**null**;

ResultSet rs=**null**;

**try** {

con = setConn();

pstmt = con.prepareStatement(sql);

pstmt.setInt(1, empnoNot);

rs = pstmt.executeQuery();

**while**(rs.next()){

**int** empno = rs.getInt("empno");

String ename = rs.getString("ename");

list.add(**new** Emp(empno, ename, **null**, 0, **null**, 0, 0, 0));

}

} **catch** (ClassNotFoundException e) {

e.printStackTrace();

} **catch** (SQLException e) {

e.printStackTrace();

} **finally**{

**try** {

**if**(rs!=**null**) rs.close();

**if**(pstmt!=**null**) pstmt.close();

**if**(con!=**null**) con.close();

} **catch** (SQLException e) {

e.printStackTrace();

}

}

**return** list;

}

**public** ArrayList<Dept> getDeptno(){

String sql = "select deptno, dname from dept order by DEPTNO";

ArrayList<Dept> list = **new** ArrayList<Dept>();

Connection con=**null**;

Statement stmt=**null**;

PreparedStatement pstmt=**null**;

ResultSet rs=**null**;

**try** {

con = setConn();

stmt = con.createStatement();

rs = stmt.executeQuery(sql);

**while**(rs.next()){

**int** deptno = rs.getInt("deptno");

String dname = rs.getString("dname");

list.add(**new** Dept(deptno, dname, **null**));

}

} **catch** (ClassNotFoundException e) {

e.printStackTrace();

} **catch** (SQLException e) {

e.printStackTrace();

} **finally**{

**try** {

**if**(rs!=**null**) rs.close();

**if**(pstmt!=**null**) pstmt.close();

**if**(con!=**null**) con.close();

} **catch** (SQLException e) {

e.printStackTrace();

}

}

**return** list;

}

**public** **void** updateEmp(Emp emp){

String sql1 = "UPDATE EMP SET ENAME=?, JOB=?, MGR=?, HIREDATE=?, SAL = ?, COMM=?, DEPTNO=? WHERE EMPNO=?";

String sql2 = "UPDATE EMP SET ENAME=?, JOB=?, MGR=null, HIREDATE=?, SAL = ?, COMM=?, DEPTNO=? WHERE EMPNO=?";

//update emp set ename='정지훈', job='연예인', mgr=7937, hiredate='2017/06/03', sal = 7000, comm=10, deptno=10 where empno=7941;;

Connection con=**null**;

PreparedStatement pstmt=**null**;

**try** {

con = setConn();

con.setAutoCommit(**false**);

**if**(emp.getMgr()!=0){

pstmt = con.prepareStatement(sql1);

pstmt.setString(1, emp.getEname());

pstmt.setString(2, emp.getJob());

pstmt.setInt(3, emp.getMgr());

pstmt.setDate(4, emp.getHiredate());

pstmt.setDouble(5, emp.getSal());

pstmt.setDouble(6, emp.getComm());

pstmt.setInt(7, emp.getDeptno());

pstmt.setInt(8, emp.getEmpno());

}**else**{

pstmt = con.prepareStatement(sql2);

pstmt.setString(1, emp.getEname());

pstmt.setString(2, emp.getJob());

pstmt.setDate(3, emp.getHiredate());

pstmt.setDouble(4, emp.getSal());

pstmt.setDouble(5, emp.getComm());

pstmt.setInt(6, emp.getDeptno());

pstmt.setInt(7, emp.getEmpno());

}

**int** result = pstmt.executeUpdate();

**if**(result>0){

System.***out***.println("사원수정성공");

con.commit();

}**else**{

System.***out***.println("입력수정실패");

con.rollback();

}

} **catch** (ClassNotFoundException e) {

e.printStackTrace();

} **catch** (SQLException e) {

e.printStackTrace();

}**finally**{

**try** {

**if**(pstmt!=**null**) pstmt.close();

**if**(con!=**null**) con.close();

} **catch** (SQLException e) {

e.printStackTrace();

}

}

}

**public** Emp getEmp(**int** empno){

Emp emp = **new** Emp();

String sql = "select \* from emp where empno=?";

Connection con=**null**;

PreparedStatement pstmt=**null**;

ResultSet rs=**null**;

**try** {

con = setConn();

pstmt = con.prepareStatement(sql);

pstmt.setInt(1, empno);

rs = pstmt.executeQuery();

**if**(rs.next()){

emp.setEmpno(empno);

emp.setEname(rs.getString("ename"));

emp.setJob(rs.getString("job"));

emp.setMgr(rs.getInt("mgr"));

emp.setHiredate(rs.getDate("hiredate"));

emp.setSal(rs.getDouble("sal"));

emp.setComm(rs.getDouble("comm"));

emp.setDeptno(rs.getInt("deptno"));

}

} **catch** (ClassNotFoundException e) {

e.printStackTrace();

} **catch** (SQLException e) {

e.printStackTrace();

}**finally**{

**try** {

**if**(rs != **null**) rs.close();

**if**(pstmt!=**null**) pstmt.close();

**if**(con!=**null**) con.close();

} **catch** (SQLException e) {

e.printStackTrace();

}

}

**return** emp;

}

**public** Emp getEmp(**int** empno, String ename){

Emp emp = **null**;

String sql = "select \* from emp where empno=? and ename=?";

Connection con=**null**;

PreparedStatement pstmt=**null**;

ResultSet rs=**null**;

**try** {

con = setConn();

pstmt = con.prepareStatement(sql);

pstmt.setInt(1, empno);

pstmt.setString(2, ename);

rs = pstmt.executeQuery();

**if**(rs.next()){

emp = **new** Emp();

emp.setEmpno(empno);

emp.setEname(rs.getString("ename"));

emp.setJob(rs.getString("job"));

emp.setMgr(rs.getInt("mgr"));

emp.setHiredate(rs.getDate("hiredate"));

emp.setSal(rs.getDouble("sal"));

emp.setComm(rs.getDouble("comm"));

emp.setDeptno(rs.getInt("deptno"));

}

} **catch** (ClassNotFoundException e) {

e.printStackTrace();

} **catch** (SQLException e) {

e.printStackTrace();

}**finally**{

**try** {

**if**(rs != **null**) rs.close();

**if**(pstmt!=**null**) pstmt.close();

**if**(con!=**null**) con.close();

} **catch** (SQLException e) {

e.printStackTrace();

}

}

**return** emp;

}

**public** **int** deleteEmp(**int** empno){

**int** result = ***FAIL***;

Connection con=**null**;

PreparedStatement pstmt=**null**;

ResultSet rs=**null**;

String sql = "delete from emp where empno=?";

**try** {

con = setConn();

con.setAutoCommit(**false**);

pstmt = con.prepareStatement(sql);

pstmt.setInt(1, empno);

result = pstmt.executeUpdate();

**if**(result>***FAIL***){

System.***out***.println(empno+"삭제");

con.commit();

}**else**{

System.***out***.println(empno+"삭제 실패");

con.rollback();

}

} **catch** (ClassNotFoundException e) {

e.printStackTrace();

} **catch** (SQLException e) {

e.printStackTrace();

}**finally**{

**try** {

**if**(rs != **null**) rs.close();

**if**(pstmt!=**null**) pstmt.close();

**if**(con!=**null**) con.close();

} **catch** (SQLException e) {

e.printStackTrace();

}

}

**return** result;

}

}

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=UTF-8"*>

<title>Insert title here</title>

<style type=*"text/css"*>

*#wrap* { width:*700px*; margin: *0 auto*; background-color: *white*; text-align: *center*;}

</style>

<script src=*"http://code.jquery.com/jquery-1.10.2.js"*></script>

<script>

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

$("input[type=text]").keyup(**function**(){

**var** enameVal = $("input[name=ename]").val();

**var** jobVal = $("input[name=job]").val();

**var** sendData = "ename="+enameVal+"&job="+jobVal;

$.ajax({

type:'post',

url:"a09\_empList.jsp",

dataType:"html",

data:sendData,

success:**function**(empList){

$("#show").html(empList);

}

});

});

$("input[type=button]").click(**function**() {

**var** enameVal = $("input[name=ename]").val();

**var** jobVal = $("input[name=job]").val();

**var** sendData = "ename="+enameVal+"&job="+jobVal;

$.ajax({

type:'post',

url:"a09\_empList.jsp",

dataType:"html",

data:sendData,

success:**function**(empList){

$("#show").html(empList);

}

});

});

});

</script>

</head>

<body>

<div id=*"wrap"*>

이름 <input type=*"text"* name=*"ename"*/>

직책 <input type=*"text"* name=*"job"*/>

<input type=*"button"* value=*"검색"* />

<br><br>

<div id=*"show"*></div>

</div>

</body>

</html>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Insert title here</title>

<style></style>

<script src=*"http://code.jquery.com/jquery-3.2.1.js"*></script>

<script>

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

$('button').click(**function**() {

**var** mId = $('#mId').val();

$.get('idCon.do?mId='+mId, **function**(data, status) {

$('.idcon\_result').html(data);

});

});

$('#mId').keyup(**function**(){

**var** mId = $('#mId').val();

$.get('idCon.do?mId='+mId, **function**(data, status) {

$('.idcon\_result').html(data);

});

});

$('input[name="pw1"]').keyup(**function**(){

**if**(frmJoin.pw.value != frmJoin.pw1.value){

$('.pw\_result').html("<font color='red'>비밀번호가 동일하지 않습니다.</font>");

} **else** {

$('.pw\_result').html("비밀번호가 일치합니다.");

}

});

});

</script>

</head>

<body>

<form action=*""* name=*"frmJoin"*>

<h1>ID중복체크</h1>

<p>ID <input type=*"text"* name=*"mId"* id=*"mId"*> <input type=*"button"* value=*"중복체크"*></p>

<span class=*"idcon\_result"*></span>

<p>비번 <input type=*"password"* name=*"pw"*></p>

<p>비번 <input type=*"password"* name=*"pw1"*></p>

<span class=*"pw\_result"*></span>

</form>

</body>

</html>

회원 가입 시 반드시 사용가능한 ID, 비밀번호일치, 가입가능한 메일. 3가지 조건이 모두 만족일 시에만 가입하기(submit) 가능



