

3주차

3주차

[11] 19.07.29

Apach Tomcat

1. Context 설정
2. Tomcat에 등록
3. html 파일 작성
4. 접근하기

Servlet

1. servlet.jar 가져오기
 2. 서블릿 클래스 작성하기
 3. 서블릿 등록
 4. 서블릿 실행
 5. Writer
- 동작 흐름
- .jar 파일 사용 (DB 연동)
- DB 연동 - Browser에 띄우기

이클립스로 하기

1. Tomcat 설정
2. Context 만들기
3. Tomcat에 Context 등록
4. Context에 Servlet 등록
5. Run On Server
6. lib 사용하기

방명록 띄워주기

[12] 19.07.30

Form으로 통신하기

방명록 게시판 만들기

1. List 띄우기
2. 글 작성

StringToken

DAO Pattern

1. DAO 인터페이스 만들기
2. 가짜 DAO 만들기
3. 가짜 DAO를 활용하여 개발하기
4. 진짜 DAO로 통합하기

프로젝트 백업하기

Servlet은 공유자원이다.

Servlet LifeCycle

1. init()
 - getInitParameter
 - getRealPath
2. service()

[13] 19.07.31

null 컬럼

PreparedStatement

가변인자

가변인자를 활용하여 Insert

익명클래스

Request 방식

Cookie

Session

동작 과정

JSP

기본 문법

JSP 는 Servlet 으로 변환된다.

JSP 는 변환된 Servlet의 멤버를 사용할 수 있다.

JSP 로 구구단 구현

[14] 19.08.01

익명 채팅 프로그램 만들기

1. sql
2. 로그인 페이지
3. 유효한 값 입력 확인
4. 입력값 유효성 검증
5. DB 연동
6. 채팅방 입장
7. 채팅방에서 사용자 권한 판별
8. 채팅목록 불러오기
9. 채팅 삭제하기
10. 채팅 입력하기
11. banana view
12. orange view

Attribute

EL (Expression Language)

기본 문법

태그

역사

특징

JSTL

기본 문법

[11] 19.07.29

Apach Tomcat

클라이언트	서버
웹 브라우저	톰캣 (Web Application Server : WAS)

브라우저가 톰캣에 접속해서 서버의 파일을 다운로드 받아 보여준다.

톰캣서버 하나에는 여러 개의 Web Application이 설치 가능하다.

이것을 **Context** 라고 한다.

context 가 가져야 하는거 : **홈 디렉토리**, **접근경로**

폴더명	설명
StudyHome/	Web App 홈 디렉토리
StudyHome/WEB-INF	클라이언트가 접근 불가능한 폴더. 주로 설정파일 저장.
StudyHome/WEB-INF/classes	Java class 파일들
StudyHome/WEB-INF/lib	동작에 필요한 jar 파일들

1. Context 설정

```

1 cd /
2 C:> mkdir StudyHome
3 C:> cd StudyHome
4 C:\StudyHome> mkdir WEB-INF
5 C:\StudyHome> cd WEB-INF
6 C:\StudyHome\WEB-INF> mkdir classes
7 C:\StudyHome\WEB-INF> mkdir lib
8 C:\StudyHome\WEB-INF> copy C:\apache-tomcat-8.0.45\webapps\ROOT\WEB-
  INF\web.xml

```

[web.xml]

```

1 <?xml version="1.0" encoding="ISO-8859-1"?>
2
3 <web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
4   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
5   xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
6                       http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
7   version="3.1"
8   metadata-complete="true">
9
10   <!-- XML 에서의 주석 -->
11
12 </web-app>

```

2. Tomcat에 등록

[C:\apache-tomcat-8.0.45\conf\server.xml]

```

1  ...
2  <Host name="localhost"  appBase="webapps" unpackWARs="true"
   autoDeploy="true">
3
4  <!-- 추가하기 -->
5  <Context docBase="C:\StudyHome" <!-- 홈 디렉토리 -->
6           path="/study" <!-- 접근경로 -->
7           reloadable="true">
8  </Context>
9  ...

```

3. html 파일 작성

[StudyHome/test.html]

```


1  <html>
2      <body>
3          hello world?
4      </body>
5  </html>

```

4. 접근하기

접근 경로 : `서버IP:port번호/Context경로.../file명`

[참고] WEB-INF 폴더에는 접근할 수 없다.

 1564366517626

 1564368003236

 1564368013836

Servlet

1. servlet.jar 가져오기

```

1  C:\Users\sohi8\Documents\ACupOfJava\libs> copy C:\apache-tomcat-
   8.0.45\lib\servlet-api.jar

```

2. 서블릿 클래스 작성하기

```

1  package day11;
2
3  import javax.servlet.ServletException;
4  import javax.servlet.http.HttpServlet;
5  import javax.servlet.http.HttpServletRequest;

```

```

6  import javax.servlet.http.HttpServletResponse;
7  import java.io.IOException;
8
9  public class HelloWorld2 extends HttpServlet {
10     @Override
11     public void service(HttpServletRequest request, HttpServletResponse
response) throws IOException, ServletException {
12         System.out.println("HELLOWORLD2");
13     }
14 }

```

컴파일

```

1  javac -d C:\StudyHome\WEB-INF\classes -classpath ../../libs/servlet-api.jar
day11/HelloWorld2.java

```

3. 서블릿 등록

[StudyHome/WEB-INF/web.xml]

```

1  <web-app ...>
2  ...
3  <!-- 서블릿 등록 -->
4  <servlet>
5      <servlet-name>ssoso</servlet-name> <!-- 서블릿 이름. 맘대로 정함. -->
6      <servlet-class>day11.HelloWorld2</servlet-class> <!-- 서블릿 클래스 지정 -
->
7  </servlet>
8
9  <!-- 서블릿 매핑 -->
10 <servlet-mapping>
11     <servlet-name>ssoso</servlet-name> <!-- 매핑할 서블릿 이름 -->
12     <url-pattern>/study01</url-pattern> <!-- 매핑할 url -->
13 </servlet-mapping>
14 ...
15 </web-app>

```

4. 서블릿 실행

서버IP:port번호/Context경로/매핑된URI

localhost:8081/study/study01

 1564369851225

[톰캣 콘솔창]

 1564370046331

5. Writer

```
1 public class HelloWorld2 extends HttpServlet {
2     @Override
3     public void service(HttpServletRequest request, HttpServletResponse
response) throws IOException, ServletException {
4         System.out.println("HELLOWORLD2");
5
6         PrintWriter out = response.getWriter();
7         out.println("<html>");
8         out.println("    <body>");
9         out.println("        HelloWorld");
10        out.println("    </body>");
11        out.println("</html>");
12        out.close();
13    }
14 }
```

1564375520000

출력방법	보내지는곳
System.out.println("str");	서버
PrintWriter out = response.getWriter(); out.println("str");	브라우저

동작 흐름

1. brower가 /study/study01 을 호출
2. 톰캣의 server.xml -> /study에 맞는 Context 를 찾음
3. Context 의 web.xml -> /study01 이랑 매핑된 서블릿이 있는지 찾음 (servlet-name)
4. 매핑된 서블릿 이름(servlet-name) 을 보고, 그 서블릿 클래스(day11.HelloWorld2)를 찾음
5. 서블릿클래스의 service(req, res)를 호출
6. service(req, res)의 결과를 response.

1. WAS 쪽에 적절히 설정이 되어 있다고 가정한다.
2. 브라우저에서 /study/study01으로 요청이 들어온다.
3. /study 로 경로를 설정한 Context 있는가? > 있다
4. (3)에서 찾은 Context의 docBase 폴더 (홈 디렉토리) 아래의 /WEB-INF/web.xml을 찾음
5. (4)의 url-pattern 중 /study01 에 해당하는 패턴이 있는가? -> 있다
6. (5)에서 찾은 url-pattern은 어떤 servlet-name을 가지고 있는가? -> ssoso
7. (6)에서 찾은 servlet-name에 해당하는 servlet이 있는가? -> 있다. day11.HelloWorld2
8. day11.HelloWorld2를 찾아 -> 인스턴스 생성 -> 오버라이딩 하는 service(req, res) 함수를 호출한다.

.jar 파일 사용 (DB 연동)

Servlet에서 외부 라이브러리를 사용할 때, Context docBase/WEB-INF/lib 에 해당 .jar 파일을 넣어줘야 한다.

[HelloWorld2.java]

```
1 public class HelloWorld2 extends HttpServlet {
2     @Override
3     public void service(HttpServletRequest request, HttpServletResponse
response) throws IOException {
4         Class.forName("oracle.jdbc.driver.OracleDriver");
5         Connection conn = DriverManager.getConnection(url, user, pw);
6
7         System.out.println(conn);
8
9         conn.close();
10    }
11 }
```

[컴파일]

```
1 javac -d C:\StudyHome\WEB-INF\classes -classpath ../../libs/servlet-api.jar
day11/HelloWorld2.java
```

[StudyHome/WEB-INF/lib]

1564377264281

[콘솔창]

1564377414021

DB 연동 - Browser에 띄우기


[HelloWorld2.java]

```
1 public class HelloWorld2 extends HttpServlet {
2     static String url = "jdbc:oracle:thin:@127.0.0.1:1521/XE";
3     static String user = "HR";
4     static String pw = "hr";
5
6     @Override
7     public void service(HttpServletRequest request, HttpServletResponse
response) throws IOException {
8         String theTime = "";
9     }
```

```

10         try {
11             Class.forName("oracle.jdbc.driver.OracleDriver");
12             Connection conn = DriverManager.getConnection(url, user, pw);
13             Statement stmt = conn.createStatement();
14
15             String sql = "SELECT SYSDATE FROM DUAL";
16
17             ResultSet rs = stmt.executeQuery(sql);
18             if ( rs.next() ) {
19                 theTime = rs.getString(1);
20             }
21             rs.close();
22
23             stmt.close();
24             conn.close();
25         } catch (Exception e) {
26             e.printStackTrace();
27         }
28
29         PrintWriter out = response.getWriter();
30         out.println("<html>");
31         out.println("    <body>");
32         out.println("HelloWorld time : " + theTime);
33         out.println("    </body>");
34         out.println("</html>");
35         out.close();
36     }
37 }

```

 1564377836778

이클립스로 하기

1. Tomcat 설정

Windows > Preferences > Server > Runtime Environment 에 tomcat 추가

 1564380893255

2. Context 만들기

File > New > Others > [Web > Dynamic Web Project] > NextNext... > Generate web.xml... 체크 > Finish

[study2/HelloWorld3.java]

```

1 package study2;
2
3 import java.io.IOException;
4
5 import javax.servlet.ServletException;

```



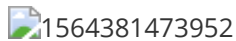
```

6  import javax.servlet.http.HttpServlet;
7  import javax.servlet.http.HttpServletRequest;
8  import javax.servlet.http.HttpServletResponse;
9
10 public class HelloWorld3 extends HttpServlet {
11
12     @Override
13     public void service(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
14         System.out.println("Hello world 3");
15     }
16
17 }

```

3. Tomcat에 Context 등록

하단 탭 Servers > Tomcat v8.0 ... 우클릭 > Add and Remove > 해당 Context Add



4. Context에 Servlet 등록

[WebContent/WEB-INF/web.xml]

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd" id="webApp_ID"
version="3.1">
3      <servlet>
4          <servlet-name>ssoso</servlet-name>
5          <servlet-class>study2.HelloWorld3</servlet-class>
6      </servlet>
7
8      <servlet-mapping>
9          <servlet-name>ssoso</servlet-name>
10         <url-pattern>/study2</url-pattern>
11     </servlet-mapping>
12 </web-app>

```


5. Run On Server

HelloWorld3.java 우클릭 > Run as > Run On Server > 돌릴 서버 선택



6. lib 사용하기

WebContent > WEB-INF > lib 에 넣기

1564383016885

[HelloWorld3.java]

```
1 public class HelloWorld3 extends HttpServlet {
2
3     @Override
4     public void service(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
5         String url = "jdbc:oracle:thin:@127.0.0.1/XE";
6         String user = "HR";
7         String pw = "hr";
8         String theTime = "";
9
10        try {
11            Class.forName("oracle.jdbc.driver.OracleDriver");
12            Connection conn = DriverManager.getConnection(url, user, pw);
13            Statement stmt = conn.createStatement();
14
15            String sql = "SELECT SYSDATE FROM DUAL";
16            ResultSet rs = stmt.executeQuery(sql);
17            if ( rs.next() ) theTime = rs.getString(1);
18            rs.close();
19
20            stmt.close();
21            conn.close();
22
23        } catch ( Exception e ) {
24            e.printStackTrace();
25        }
26
27        PrintWriter out = response.getWriter();
28        out.println("<html>");
29        out.println(" <body>");
30        out.println(" THIS TIME IS " + theTime);
31        out.println(" </body>");
32        out.println("</html>");
33        out.close();
34    }
35
36 }
```

1564383341452

방명록 띄워주기

[HelloWorld3.java]

```
1 public class HelloWorld3 extends HttpServlet {
2
3     @Override
```

```

4      public void service(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
5          String url = "jdbc:oracle:thin:@127.0.0.1/XE";
6          String user = "HR";
7          String pw = "hr";
8
9          ArrayList<BangMyungVO> ls = new ArrayList<BangMyungVO>();
10
11         try {
12             Class.forName("oracle.jdbc.driver.OracleDriver");
13             Connection conn = DriverManager.getConnection(url, user, pw);
14             Statement stmt = conn.createStatement();
15
16             String sql = "SELECT no, gul, the_time FROM bangmyung_t";
17             ResultSet rs = stmt.executeQuery(sql);
18
19             while ( rs.next() ) {
20                 BangMyungVO vo = new BangMyungVO();
21
22                 vo.setNo( rs.getInt("no") );
23                 vo.setGul( rs.getString("gul") );
24                 vo.setTheTime( rs.getString("the_time") );
25
26                 ls.add( vo );
27             }
28             rs.close();
29
30             stmt.close();
31             conn.close();
32
33         } catch ( Exception e ) {
34             e.printStackTrace();
35         }
36
37         // brower의 Charset 세팅
38         response.setContentType("text/html;charset=utf-8");
39         PrintWriter out = response.getWriter();
40         out.println("<html>");
41         out.println(" <body>");
42         for ( BangMyungVO vo : ls ) {
43             out.println( vo.toString() );
44         }
45         out.println(" </body>");
46         out.println("</html>");
47         out.close();
48     }
49
50 }

```

 1564385255521

Form으로 통신하기

Request	Response
요청과 관련된 함수	응답과 관련된 함수

[web.xml]

```
1  ...
2  <servlet>
3      <servlet-name>myFormTest</servlet-name>
4      <servlet-class>study2.FormTestServlet</servlet-class>
5  </servlet>
6
7  <servlet-mapping>
8      <servlet-name>myFormTest</servlet-name>
9      <url-pattern>/form_test2</url-pattern>
10 </servlet-mapping>
11 </web-app>
```

[FormTestServlet.java]

```
1  public class FormTestServlet extends HttpServlet {
2
3      @Override
4      public void service(HttpServletRequest request, HttpServletResponse
response) throws IOException, ServletException {
5          // request charset 세팅
6          request.setCharacterEncoding("utf-8");
7
8          // request.getParameter(name) : 해당 파라미터값 가져옴
9          String gul = request.getParameter( "gul" );
10         String love = request.getParameter( "love" );
11         String hate = request.getParameter( "hate" );
12         String method = request.getParameter( "method" );
13         String password = request.getParameter( "pwd" );
14         String content = request.getParameter( "content" );
15         String animal = request.getParameter( "animal" );
16
17         System.out.println("[gul] " + gul);
18         System.out.println("[Love & Hate] " + love + " & " + hate);
19         System.out.println("[method] " + method);
20         System.out.println("[password] " + password);
21         System.out.println("[content] " + content);
22         System.out.println("[animal] " + animal);
23         System.out.println("=====");
24
25         // 해당 url로 리다이렉트
26         response.sendRedirect("/study2/test_01.html");
```

```
27     }
28 }
```

[test_01.html]

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <meta charset="utf-8">
5  <title>Insert title here</title>
6  </head>
7  <body>
8      <form method="POST" action="form_test2">
9          <!-- param key : gu1 -->
10         Gu1 <input type="text" name="gu1" size="50" /> <br><br>
11
12         <!-- [checkbox] checkbox 를 체크하면 해당 value가, 아니면 null이 넘어간
13         다. -->
14         Love <input type="checkbox" name="love" value="apple" />
15         Hate<input type="checkbox" name="hate" value="banana" /> <br>
16
17         <!-- [radio] 같은 이름을 가진 radio 는 하나만 체크가 가능하다. -->
18         Plus <input type="radio" name="method" value="plus" />
19         Minus <input type="radio" name="method" value="minus"
20         checked="checked" /> <br>
21
22         <!-- [password] type text와 동일하지만, 값이 노출되지 않는다. -->
23         Password <input type="password" name="pwd" size="16" /> <br><br>
24
25         <!-- [textarea] 여러 줄을 입력할 때 사용된다. -->
26         <!-- 태그 안의 값이 default -->
27         <!-- rows: 가로줄, cols: 세로줄 -->
28         Content <br>
29         <textarea name="content" rows="7" cols="10"> HelloWorld </textarea>
30
31         <br>
32
33         <!-- [select] option 중 하나 선택 -->
34         Animal
35         <select name="animal">
36             <option value="dog">개</option>
37             <option value="cat">고양이</option>
38             <option value="mouse">쥐</option>
39             <option value="tiger">호랑이</option>
40             <option value="wolf">늑대</option>
41         </select> <br>
42
43         <input type="submit"/>
44     </form>
45 </body>
46 </html>
```

방명록 게시판 만들기

1. List 띄우기

[BangMyungList.java]

```
1 // url-pattern : /bangmyung_list
2 public class BangMyungList extends HttpServlet {
3     @Override
4     public void service(HttpServletRequest request, HttpServletResponse
response) throws IOException, ServletException {
5         String url = "jdbc:oracle:thin:@127.0.0.1/XE";
6         String user = "HR";
7         String pw = "hr";
8
9         ArrayList<BangMyungVO> ls = new ArrayList<BangMyungVO>();
10
11         try {
12             Class.forName("oracle.jdbc.driver.OracleDriver");
13             Connection conn = DriverManager.getConnection(url, user, pw);
14             Statement stmt = conn.createStatement();
15
16             String sql = "SELECT no, gul, the_time FROM bangmyung_t ORDER
BY no desc";
17             ResultSet rs = stmt.executeQuery(sql);
18
19             while ( rs.next() ) {
20                 BangMyungVO vo = new BangMyungVO();
21
22                 vo.setNo( rs.getInt("no") );
23                 vo.setGul( rs.getString("gul") );
24                 vo.setTheTime( rs.getString("the_time") );
25
26                 ls.add( vo );
27             }
28             rs.close();
29
30             stmt.close();
31             conn.close();
32
33         } catch ( Exception e ) {
34             e.printStackTrace();
35         }
36
37         response.setContentType("text/html;charset=utf-8");
38         PrintWriter out = response.getWriter();
39         out.println("<html>");
40         out.println(" <body>asdf");
41         out.println(" <table border=\"1\">");
42         out.println("<tr><td><font face='impact'>NO</font></td><td><font
face='impact'>글</font></td><td><font face='impact'>시간</font></td></tr>");
43         for ( BangMyungVO vo : ls ) {
44             out.println(" <tr>");
45             out.println("<td bgColor='#aabbcc'>" + vo.getNo() + "</td>");
46             out.println("<td bgColor='#ccdde'>" + vo.getGul() + "</td>");
47             out.println("<td bgColor='#bbccdd'>" + vo.getTheTime() +
</td>");
```

```

48         out.println("    </tr>");
49     }
50     out.println(" </table>");
51     out.println(" </body>");
52     out.println("</html>");
53
54     out.close();
55 }
56 }

```

2. 글 작성

[BangMyungList.java]

```

1  out.println("<form method='POST' action='bangmyung_add2'>");
2  out.println("<input type='text' size='50' name='gul' />");
3  out.println("<input type='submit' />");
4  out.println("</form>");

```

[BangMyungAdd2.java]

```

1  // url-pattern : /bangmyung_add2
2  // DB에 방명록 insert
3  public class BangMyungAdd2 extends HttpServlet {
4      String url = "jdbc:oracle:thin:@127.0.0.1:1521/XE";
5      String user = "HR";
6      String pw = "hr";
7
8      @Override
9      public void service(HttpServletRequest request, HttpServletResponse
response) throws IOException, ServletException {
10         System.out.println("BangMyungAdd2");
11         response.setCharacterEncoding("utf-8");
12
13         try {
14             Class.forName("oracle.jdbc.driver.OracleDriver");
15             Connection conn = DriverManager.getConnection(url, user, pw);
16             Statement stmt = conn.createStatement();
17
18             request.setCharacterEncoding("utf-8");
19             String gul = request.getParameter( "gul" );
20             String sql = "INSERT INTO bangmyung_t VALUES (
seq_bangmyung.NEXTVAL, '" + gul + "', SYSDATE) ";
21             stmt.executeUpdate(sql);
22
23             stmt.close();
24             conn.close();
25         } catch (Exception e) {
26             e.printStackTrace();
27         }
28
29         response.sendRedirect("/study2/bangmyung_list");
30     }

```

StringToken

문장에서 공백을 기준으로 하나씩 잘라서 볼 때 사용한다.

String.split() 보다 가볍게 동작한다.

```

1 String l = "이것은 글을 쓴 내용에 해당하는 #해시태그 문장입니다.";
2
3 StringTokenizer st = new StringTokenizer(l);
4 while ( st.hasMoreTokens() ) { // token이 있으면
5     String tkn = st.nextToken(); // token 떼옴
6
7     // String 조작 메소드
8     System.out.println( tkn );
9     System.out.println( tkn.indexOf("해당") != -1 );
10    if ( tkn.charAt(0) == '#' ) {
11        System.out.println( tkn );
12    }
13
14    // 금칙어 조사는 보통 startswith() 을 사용
15    System.out.println( tkn.startswith("해당") );
16 }

```

Type	함수	설명
boolean	hasMoreTokens()	다음 토큰이 있는지 반환
String	nextToken()	다음 토큰으로 넘어가고, String으로 반환

DAO Pattern

데이터를 다루는 클래스를 독립시키는 패턴.

데이터베이스 관련 기능과 웹프로그래밍 관련 기능을 독립시킴으로써, 독립적으로 개발하고 나중에 끼워맞추는 일을 가능케함.

DAO 패턴을 활용하면 웹개발과 DB개발을 독립적으로 할 수 있다.

(가짜 DAO로 흐름을 잡아두고 나중에 연결)

1. DAO 인터페이스 만들기

[인터페이스]

방명록 게시판에 필요한 데이터 입출력 기능을 모아서 추상화 한 인터페이스


```

1 public interface BangMyungDAO {
2     public void add ( BangMyungVO vo ) throws Exception;
3     public List<BangMyungVO> findAll() throws Exception;
4 }

```

2. 가짜 DAO 만들기

[가짜 DAO]

추후 구현될 기능을 비슷하게 흉내낸 DAO.

가짜 데이터를 사용한다.

```

1 // 테스트용 가짜 DAO
2 // DB 없이 돌아감
3 public class BangMyungDAO_KaraImpl implements BangMyungDAO {
4
5     // 가짜 DB 역할
6     private static List<BangMyungVO> data = new ArrayList<BangMyungVO>();
7
8     @Override
9     public void add(BangMyungVO vo) throws Exception {
10         data.add(vo);
11     }
12
13     @Override
14     public List<BangMyungVO> findAll() throws Exception {
15         return data;
16     }
17 }

```

3. 가짜 DAO를 활용하여 개발하기

디비 담당자가 진짜 DAO를 개발하는 동안,

웹 담당자는 가짜 DAO를 활용하여 개발한다.

[Add 클래스]

```

1 public class BangMyungAdd2 extends HttpServlet {
2
3     @Override
4     public void service(HttpServletRequest request, HttpServletResponse
5 response) throws ServletException, IOException {
6         request.setCharacterEncoding("utf-8");
7
8         // 가짜 DAO
9         BangMyungDAO dao = new BangMyungDAO_KaraImpl();
10        BangMyungVO vo = new BangMyungVO();
11
12        // 가짜 데이터
13        vo.setNo(101);
14    }
15 }

```

```

13         vo.setGul( request.getParameter("gul") );
14         vo.setTheTime("1999-01-01 12:00:00");
15
16         try {
17             dao.add(vo);
18         } catch (Exception e) {
19             e.printStackTrace();
20         }
21     }
22
23     response.sendRedirect("/study2/list");
24 }

```

[List 클래스]

```

1 public class BangMyungList extends HttpServlet {
2
3     @Override
4     public void service(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
5
6         // 가짜 DAO 의 가짜 DB의 데이터를 가져옴
7         BangMyungDAO dao = new BangMyungDAO_KaraImpl();
8         List<BangMyungVO> ls = new ArrayList<BangMyungVO>();
9
10        try {
11            ls = dao.findAll();
12        } catch (Exception e) {
13            e.printStackTrace();
14        }
15
16        // html 작성 코드
17        ...
18    }
19
20 }

```

4. 진짜 DAO로 통합하기

[Add 클래스]

```

1 public class BangMyungAdd2 extends HttpServlet {
2
3     @Override
4     public void service(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
5         request.setCharacterEncoding("utf-8");
6
7         // DAO만 바꿔줌!
8         BangMyungDAO dao = new BangMyungDAO_OracleImpl();
9         BangMyungVO vo = new BangMyungVO();
10
11         ..
12     }

```

[List 클래스]

```

1 public class BangMyungList extends HttpServlet {
2
3     @Override
4     public void service(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
5
6         // DAO만 바꿔줌!
7         BangMyungDAO dao = new BangMyungDAO_OracleImpl();
8         List<BangMyungVO> ls = new ArrayList<BangMyungVO>();
9
10         ...
11     }
12
13 }

```

 1564466808196

프로젝트 백업하기

프로젝트명 우클릭 > Export > Export

 1564470637169

General > Archive File > Next

 1564470729885

원하는 프로젝트와 반출(?) 경로 선택하고 저장

나중에 import 가능

Servlet은 공유자원이다.

Servlet은 인스턴스 하나만 만들고 재사용한다.

즉, 다른 곳에서 여러번 접속하더라도 다 같은 인스턴스이다.

[instance가 하나임을 확인하는 코드]

```
1 // 하나의 instance를 재사용함을 확인하는 코드
2 public class LifeServlet extends HttpServlet {
3
4     @Override
5     public void service(HttpServletRequest req, HttpServletResponse res)
6     throws ServletException, IOException {
7         /* 많은 사람들이 다른 곳에서 각기 접속하더라도, this.toString()의 결과는
8         동일하다.
9         * -> 동일한 인스턴스가 계속 재사용되고 있다.
10        * -> 한꺼번에 많이 들어와도, 메모리를 적게 소모한다.
11        */
12        System.out.println( this.toString() );
13    }
14 }
```

따라서, Servlet은 여러 Thread에게 공유된다. (공유자원)

```
1 // 다른 thread에 의해, 접속 중에 변수값이 변하는 코드
2 public class LifeServlet extends HttpServlet {
3     int i = 0;
4
5     @Override
6     public void service(...) ... {
7         ...
8         i = i+1;
9         for (int j = 0; j < 100; j++) {
10            /* 원래라면 0만 출력되어야 하지만,
11            * 동시에 여럿이 접속하면, 0 ... 1... 2... 점점 숫자가 커진다.
12            * -> 다른 스레드가 여길 통과하며 i = i+1;을 수행했다.
13            */
14            out.println( this.i );
15        }
16        ...
17    }
18 }
```

 1564473845364

- Servlet은 하나의 instance만 갖고 있다.
- 사용자는 Thread를 통해서 Tomcat server에 접속한다.
- 동시접속이 있을 경우, 여러 개의 Thread가 동시에 Servlet instance에 접근한다.
 - 즉, Servlet instance는 공유자원이다.

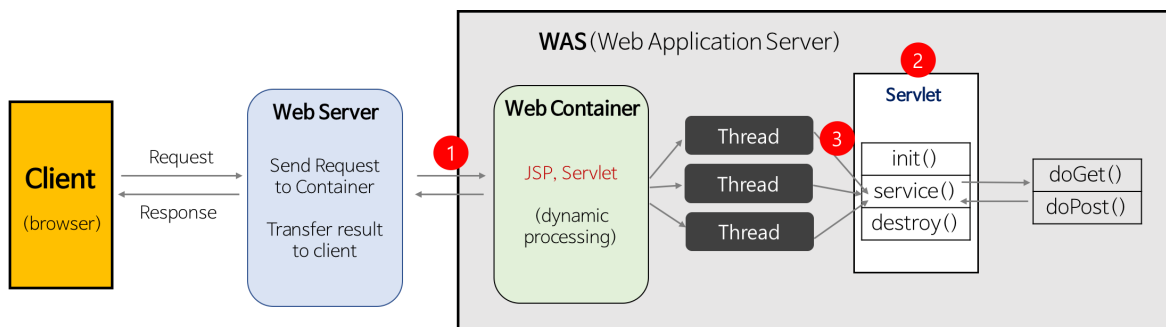
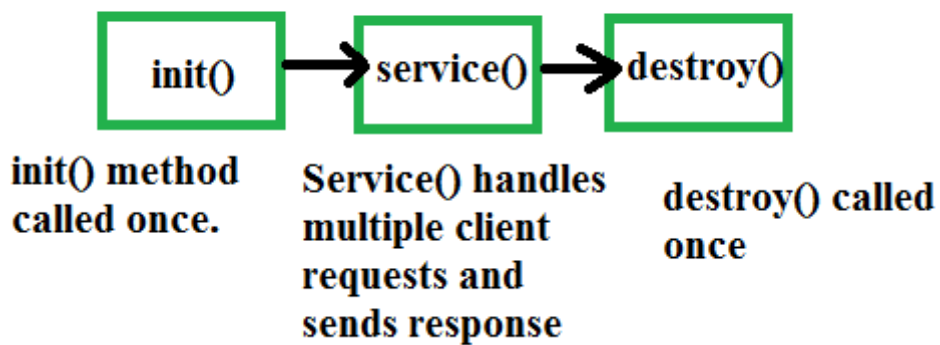
Servlet은 여러 Thread에게 공유되므로, 동기화 혹은 지역변수만 사용해야 한다.

```
1 // synchronized 를 사용하여 동기화 시킨 코드
2 ...
3 synchronized ( this ){
4     i = i+1;
5     for (int j = 0; j < 100; j++){
6         out.println( this.i );
7     }
8 }
9 ...
```

Servlet LifeCycle

Servlet의 instance는 LifeCycle을 거치며 재활용된다.

Life cycle methods of a Servlet



```
1 public class ConfigServlet extends HttpServlet {
2
3     @Override
4     public void init(ServletConfig config) throws ServletException {
5         system.out.println("init");
6     }
7
8     @Override
```

```

10     public void service(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
11         System.out.println("service");
12     }
13
14     @Override
15     public void destroy() {
16         System.out.println("destroy");
17     }
18
19 }

```

```

1  init
2  service
3  service
4  service
5  service
6  destroy

```

1. init()

init (config)

최초 요청시에만 호출.

instance 재할용을 위하여 내부적으로 적재함.

[web.xml]

```

1  ...
2  <servlet>
3      <servlet-name>configServlet</servlet-name>
4      <servlet-class>study3.ConfigServlet</servlet-class>
5      <init-param>
6          <param-name>apple</param-name>
7          <param-value>ILikeApple</param-value>
8      </init-param>
9  </servlet>
10 ...

```

[서블릿]

```

1  @Override
2  public void init(ServletConfig config) throws ServletException {
3      System.out.println("init");
4
5      // getInitParameter : web.xml에 설정된 내용을 읽어올 수 있는 방법을 제공한다.
6      String val = config.getInitParameter("apple");

```

```

7      System.out.println( val ); // ILikeApple
8
9      // ServletContext: 서블릿이 서블릿 컨테이너와 통신하기 위한 방법 제공
10     ServletContext application = config.getServletContext();
11     // WEB-INF 폴더의 절대경로값을 얻어온다.
12     String path = application.getRealPath("/WEB-INF/");
13     System.out.println( path ); // C:\Users\sohi8\workspace\...\study3\WEB-
    INF\
14 }

```

getInitParameter

web.xml 에 <init-param> 태그로 정의된 값을 읽어온다.

```

1 String val = config.getInitParameter("apple");
2 System.out.println( val ); // ILikeApple

```

getRealPath

해당 폴더의 절대경로값을 얻어온다.

```

1 ServletContext application = config.getServletContext();
2 String path = application.getRealPath("/WEB-INF/");
3 System.out.println( path ); // C:\Users\sohi8\workspace\...\study3\WEB-INF\

```

FileInputStream 으로 파일을 읽어오기 위해 절대경로를 가져오는 등의 쓰임새가 있다.

2. service()

service (req, res)

요청이 발생할 때 마다 호출.

요청에 대한 내용을 처리하고 응답해준다.

```

1 @Override
2 public void service(HttpServletRequest request, HttpServletResponse
    response) throws ServletException, IOException {
3     System.out.println("service");
4
5     // 컨텍스트 경로
6     String l = request.getContextPath();
7     System.out.println(l); // /study3
8
9     // request URI ( IP, port 제외 )
10    String m = request.getRequestURI();
11    System.out.println(m); // /study3/config
12

```

```

13 // 접속한 브라우저의 IP 주소값
14 String n = request.getRemoteAddr();
15 System.out.println(n); // 192.168.2.25
16
17 // request Header 값
18 String o = request.getHeader("User-Agent");
19 System.out.println(o); // ... Chrome/67.0.3396.87 Mobile Safari/537.36
20
21 // request Parameter 값
22 String use = request.getParameter("use");
23 /// 호출 url : /config/use=addCookie
24 System.out.println(use); // addCookie
25 }

```

[13] 19.07.31

null 컬럼

```

1 CREATE TABLE temp20t (
2     data char(3) null
3 );

```

```

1 ...
2 // null 컬럼이 하나 일 때는 이렇게 가능 하지만,
3 /// 여러개면 if문이 답이 없다; => PreparedStatement 사용!
4 String sql = ( data != null ) ?
5     "INSERT INTO temp20t VALUES ('"+ data +"')" :
6     "INSERT INTO temp20t VALUES (null)";
7 stmt.executeUpdate(sql);
8 ...

```

PreparedStatement

일반적인 Statement와 달리, 지정한 SQL query만 수행할 수 있는 Statement.

Query문에 들어갈 값을 ? 로 지정한 후, 추후에 set 한다.

Statement	PreparedStatement
어떤 SQL 쿼리문이든 수행 가능.	생성자에 넣은 쿼리만 수행 가능.
	쿼리문에 넣을 값을 ?로 지정하여, 추후에 넣을 수 있음.
	단, ? 영역의 순서는 1부터 시작한다.
execute 시에 query문 넣어야함.	execute 시에 query문 넣을 필요 x


```

1 // ? 로 지정한 영역은 추후 set 한다.
2 String sql = "INSERT INTO temp20t VALUES ( ? )";
3 // 생성자 매개변수로 SQL query문 넣음
4 PreparedStatement stmt = conn.prepareStatement(sql);
5
6 String data = null;
7 // query의 ? 영역을 set 한다. 순서는 1부터 시작.
8 stmt.setString( 1, data ); // null 이든 아니든 신경쓰지 않고 넣을 수 있음.
9 // query 문 넣지 않고 execute 수행.
10 stmt.executeUpdate();

```

가변인자

매개변수의 개수가 정해지지 않을 때 사용하며, 배열의 형태로 매개변수를 받는다.

```

1 public static void test(Object... args){ // Object형 가변인자
2     System.out.println(args.length);
3 }
4
5 public static void main(String[] args) {
6     test("apple");
7     test();
8     test("apple", "banana");
9     test(100, 200, 300); // int -> Integer (AutoBoxing)
10 }

```

가변인자를 활용하여 Insert

```

1 public class Test113 {
2
3     public static int update( String sql, Object... args) throws Exception
4     {
5         int rc = 0;
6
7         Class.forName("oracle.jdbc.driver.OracleDriver");
8         Connection conn = DriverManager.getConnection(
9             "jdbc:oracle:thin:@127.0.0.1:1521/XE", "HR", "hr");
10
11         PreparedStatement stmt = conn.prepareStatement(sql);
12
13         for (int i = 0; i < args.length; i++) {
14             if ( args[i] == null ) {
15                 System.out.println("null");
16                 stmt.setObject(i+1, null);
17             } else if ( args[i] instanceof Integer ) {
18                 int r = ( (Integer)args[i] ).intValue();
19                 System.out.println("integer : " + r);
20                 stmt.setInt(i+1, r);
21             } else if ( args[i] instanceof Double ) {
22                 double r = ( (Double)args[i] ).doubleValue();
23

```

```

24         System.out.println("double : " + r);
25         stmt.setDouble(i+1, r);
26
27     } else if ( args[i] instanceof String ) {
28         System.out.println("string : " + (String)args[i] );
29         stmt.setString(i+1, (String) args[i]);
30
31     }
32 }
33
34 stmt.executeUpdate();
35
36 stmt.close();
37 conn.close();
38
39 return rc;
40 }
41
42 public static void main(String[] args) throws Exception {
43     String sql = "INSERT INTO temp20t VALUES ( ? )";
44
45     update(sql, 55);
46 }
47
48 }

```

익명클래스

이름이 없는 클래스. (조상은 있다.)

재사용 불가. 클래스 선언 및 인스턴스 생성만이 역할.

`final` 로컬변수 만 익명클래스 내부에서 사용할 수 있다.

```

1  interface Itemp {
2      public void print();
3  }
4
5  public class Test114 {
6      public static void main(String[] args) {
7          // final 로컬변수
8          final int i = 100;
9
10         // interface Itemp 를 구현하는 익명클래스
11         Itemp t = new Itemp() {
12             @Override
13             public void print() {
14                 System.out.println("익명클래스" + i);
15             }
16         };
17
18         t.print(); // 익명클래스 100
19     }
20 }

```

Request 방식

GET	POST
신속, 경량, 데이터 노출	느림, 무거움, 보안
example.com/mypage?key=value	
request.getParameter(key) 로 추출	

Cookie

- 쿠키는 클라이언트 로컬에 저장되는 키와 값이 들어있는 작은 데이터 파일이다.
- 쿠키에는 이름, 값, 만료날짜(쿠키 저장기간), 경로 정보가 들어있다.
- 쿠키는 일정시간동안 데이터를 저장할 수 있다. (로그인 상태 유지에 활용)
- 쿠키는 클라이언트의 상태 정보를 로컬에 저장했다가 참조한다.
- 자동로그인, 팝업에서 "오늘 더 이상 이 창을 보지 않음" 체크, 쇼핑물의 장바구니



함수	설명
request.getCookies()	request의 쿠키들을 받음
response.addCookie(key, value)	response 에게 Set-Cookie

[쿠키 서블릿]

```
1 // url-pattern : /study3/state?use=addCookie
2 @Override
3 public void service(...) throws ServletException, IOException {
4     String use = request.getParameter("use");
5
6     if ( use.equals("addCookie") ) {
7         System.out.println("add Cookie.");
8     }
9 }
```

```

8
9      Cookie ck = new Cookie("name", "apple");
10     response.addCookie( ck ); // addCookie
11
12     } else if ( use.equals("readCookie") ) {
13
14         Cookie[] cks = request.getCookies(); // getCookies
15
16         System.out.println("cookie len : " + cks.length);
17         if (cks != null) {
18             for (Cookie ck: cks) {
19                 System.out.println( ck.getName() + ", " + ck.getValue() );
20             }
21         } else {
22             System.out.println("cookie is null");
23         }
24     }
25 }

```

Session

- 일정 시간동안 같은 브라우저로 부터 들어오는 일련의 요구를 하나의 상태로 보고 그 상태를 유지 하는 기술
- 즉, 웹 브라우저를 통해 웹 서버에 접속한 이후로 브라우저를 종료할 때 까지 유지되는 상태
- 클라이언트가 Request를 보내면, 해당 서버의 엔진이 클라이언트에게 유일한 ID를 부여하는 데 이것이 세션ID다.
- 브라우저마다 개별적인 기억공간이다.

동작 과정

Server 에는 Client 에게 개별 저장공간과 Session Id (JSESSIONID)를 발급하고, 이를 쿠키로 보낸다.

Client는 Server에 요청할 때 마다, 쿠키를 통해 Session Id 를 전송한다.

 1564550170399

1. 최초 getSession() 호출 시에는 고유번호 없이 요청이 들어온다.
 1. 기억장소 생성 : Map<String, Object> 형태
 2. 고유번호 (Session Id) 생성
 3. JSESSIONID 라는 키로 Session Id를 쿠키에 저장
2. 이후 getSession() 호출
 1. Session Id 를 받아서 해당 기억장소를 찾음

[세션 서블릿]

```

1 // url-pattern : /study3/session
2 public class SessionServlet extends HttpServlet {
3     @Override
4     public void service(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
5         HttpSession session = request.getSession();
6         System.out.println( session.isNew() ); // 처음만 true, 이후에는 false
7
8         session.setAttribute("apple", "MYObject!!");
9         Object value = session.getAttribute("apple");
10        System.out.println( value.toString() );
11    }
12 }

```

[/study3/session 접속 후 /study3/state?use=readCookie]

```

1 true // session.isNew()
2 MYObject!! // value.toString()
3 cookie len : 1
4 JSESSIONID, F0B09427F5EFEF7CB61D2719EF643A7D // ck.getName(), ck.getValue()

```

발급 받은 session id 를 client가 request 시 cookie 로 보냄

	Cookie	Session
Where?	브라우저	서버
저장 형태	파일	Map <String, Object>
	외부조작 가능	외부조작 어렵
다중 WAS 환경	문제x	세션관리에 대한 고민 필요

초반엔 세션을 사용했으나, 요즘엔 쿠키를 많이 쓰는 경향이 있다.

JSP

- 서블릿 이후 ASP의 등장으로 위기를 느낀 자바쪽에서 ASP 비슷하게 만든것

1. jsp 파일을 요청하면 톰캣은 이것을 *.java 파일로 변환한다. (서블릿 코드)
2. 이것을 컴파일하고, 인스턴스를 만들어, 적재하고, 인스턴스를 재활용한다.

기본 문법

- `<%! i %>` : 멤버변수 (i 는 변수)
- `<% ... %>` : 로컬변수
- `<%= ... %>` : 화면 출력

```

1 <%@ page language="java" contentType="text/html; charset=utf-8"
2   pageEncoding="EUC-KR"%>
3 <%! int i = 0; %> <%-- 멤버변수 --%>
4 <% int j = 0; %> <%-- 로컬변수 --%>
5 <!DOCTYPE html>
6 <html>
7 <body>
8     <%= i++ %>
9     <%= j++ %>
10 </body>
11 </html>
12 <%-- JSP 주석 --%>
13 <%-- http://localhost:8081/study3/Test115.jsp --%>

```

JSP 는 Servlet 으로 변환된다.

[JSP서블릿]

```

1 public class JSPServlet extends HttpServlet {
2
3     private ServletContext application = null;
4     private ServletConfig config = null;
5
6     // -----
7     // <%! ... %> 은 여기로 온다.
8     int i = 0; // <%! int i = 0; %>
9     // -----
10
11     @Override
12     public void init(ServletConfig config) throws ServletException {
13         this.config = config;
14         this.application = config.getServletContext();
15     }
16
17     @Override
18     public void service(HttpServletRequest request, HttpServletResponse
19 response) throws ServletException, IOException {
20         HttpSession session = request.getSession();
21
22         response.setContentType("text/html; charset=utf-8");
23         PrintWriter out = response.getWriter();
24
25         // -----
26         /* <% ... %> 은 여기로 온다.
27          * <%= ... %> 은 out.println( ... ) 으로 변환된다.
28          */
29         int j = 0; // <% int j = 0; %>
30         out.println(i++); // <%= i++ %>
31         out.println(j++); // <%= j++ %>
32         // -----
33
34         out.flush();
35         out.close();
36     }
37 }

```

JSP 는 변환된 Servlet의 멤버를 사용할 수 있다.

```
1 <%@ page language="java" contentType="text/html; charset=utf-8"
2   pageEncoding="EUC-KR"%>
3 <!DOCTYPE html>
4 <%! // [멤버 공간]
5
6   // 불가능
7   // out.println( config.toString() );
8
9   int i = 0; // 가능
10  // i = i + 1; // 불가능
11
12  // 가능
13  public void print() { }
14
15 %>
16 <html>
17 <body>
18 <% // [로컬 공간]
19   // 가능 ( out, config 등은 JSP가 변환된 JSPServlet에 존재한다. )
20   out.println( config.toString() );
21   String l = request.getParameter("abcd");
22   String m = config.getInitParameter("initparam");
23 %>
24 </body>
25 </html>
```

JSP 로 구구단 구현

```
1 <%@ page language="java" contentType="text/html; charset=utf-8"
2   pageEncoding="EUC-KR"%>
3 <!DOCTYPE html>
4 <%!
5   String getColor() {
6       char[] chs = "0123456789abcdef".toCharArray();
7       StringBuffer sb = new StringBuffer();
8
9       for (int i = 0; i < 6; i++) {
10           int idx = (int) (Math.random() * 16);
11           sb.append(chs[idx]);
12       }
13
14       return sb.toString();
15   }
16 %>
17 <html>
18 <body>
19 <table>
20   <% for (int i = 1; i <= 9; i++) { %>
21   <tr>
22     <% for (int j = 1; j <= 9; j++) { %>
```

```

23         <td bgColor='#<%= getColor() %>'> <%= i %> * <%= j %> = <%= i*j
    %> </td>
24     <% } %>
25 </tr>
26 <% } %>
27 </table>
28 </body>
29 </html>

```

1564555877872

[14] 19.08.01

회원가입 없고, 권한은 있는 익명 채팅 프로그램 만들기.

익명 채팅 프로그램 만들기

1. sql

[talk.sql.txt]

```

1  DROP TABLE talk_room_t;
2
3  CREATE TABLE talk_room_t(
4      room_no number(5),
5      apple char(4),
6      banana char(4),
7      orange char(4)
8  );
9
10 INSERT INTO talk_room_t values(0, '1234', '2345', '3456');
11 COMMIT;
12
13 CREATE TABLE talk_t(
14     talk_no number(6),
15     room_no number(5),
16     content varchar2(128)
17 );
18
19 CREATE SEQUENCE seq_talk;
20
21 INSERT INTO talk_t VALUES ( seq_talk.NEXTVAL, 0, 'hello my name is zito' );
22 COMMIT;

```

2. 로그인 페이지

[talk_login.jsp]

```

1  <!-- talk_login.jsp -->

```



```

2 <%@ page language="java" contentType="text/html; charset=utf-8"
3   pageEncoding="EUC-KR"%>
4 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
   "http://www.w3.org/TR/html4/loose.dtd">
5 <html>
6 <head>
7 <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
8 <title>talk_login.jsp</title>
9 </head>
10 <body>
11
12 <form method="POST" action="talk_login2.jsp">
13     roomNo : <input type="text" name="roomNo" size="4"/> <br>
14     Password : <input type="password" name="pwd" size="8"/> <br>
15     <input type="submit"/>
16 </form>
17
18 </body>
19 </html>

```

3. 유효한 값 입력 확인

[talk_login2.jsp]

```

1 <%-- talk_login2.jsp --%>
2 <%@page import="utils.Util" %>
3 <%@page pageEncoding="EUC-KR" %>
4
5 <%
6   String ctxPath = request.getContextPath();
7
8   int roomNo = Util.parseInt( request.getParameter("roomNo") );
9   String pwd = request.getParameter("pwd");
10
11   if ( roomNo == -1 || pwd == null || pwd.equals("") ) {
12       response.sendRedirect( ctxPath + "/talk_login.jsp");
13       return; // redirect 되었으니 의미 x
14   }
15
16 %>

```

[utils/Util.java]

```

1 package utils;
2
3 public class Util {
4     public static int parseInt( String str ) {
5         try {
6             int r = Integer.parseInt(str);
7             return r;
8         } catch ( Exception e ) {
9             return -1;
10        }
11    }
12 }

```

4. 입력값 유효성 검증

[talk_login2.jsp]

```

1 <% // 입력값 유효성 검증
2 String ctxPath = request.getContextPath();
3
4 int roomNo = Util.parseInt( request.getParameter("roomNo") );
5 String pwd = request.getParameter("pwd");
6
7 if ( roomNo == -1 || pwd == null || pwd.equals("") ) {
8     response.sendRedirect( ctxPath + "/talk_login.jsp");
9     return; // redirect 되었으니 의미 x
10 }
11
12 %>

```

5. DB 연동

[vo/TalkRoomVO.java]

```

1 package vo;
2
3 public class TalkRoomVO {
4     private Integer roomNo = null;
5     private String apple = null;
6     private String banana = null;
7     private String orange = null;
8
9     public Integer getRoomNo() {
10         return roomNo;
11     }
12     public void setRoomNo(Integer roomNo) {
13         this.roomNo = roomNo;
14     }
15     public String getApple() {
16         return apple;
17     }
18 }

```

```

18     public void setApple(String apple) {
19         this.apple = apple;
20     }
21     public String getBanana() {
22         return banana;
23     }
24     public void setBanana(String banana) {
25         this.banana = banana;
26     }
27     public String getOrange() {
28         return orange;
29     }
30     public void setOrange(String orange) {
31         this.orange = orange;
32     }
33 }

```

[talk_login2.jsp]

```

1 // 유효성 검증
2
3 <% // DB 연동
4 Exception err = null;
5 TalkRoomVO vo = null;
6
7 try {
8     Class.forName("oracle.jdbc.driver.OracleDriver");
9     Connection conn = DriverManager.getConnection(
10         "jdbc:oracle:thin:@127.0.0.1:1521/XE",
11         "HR", "hr" );
12
13     String sql = "SELECT * FROM talk_room_t WHERE room_no=?";
14     PreparedStatement stmt = conn.prepareStatement(sql);
15     stmt.setInt(1, roomNo);
16
17     ResultSet rs = stmt.executeQuery();
18     if ( rs.next() ) {
19         vo = new TalkRoomVO();
20
21         vo.setRoomNo( rs.getInt("room_no") );
22         vo.setApple( rs.getString("apple") );
23         vo.setBanana( rs.getString("banana") );
24         vo.setOrange( rs.getString("orange") );
25
26         System.out.println(vo.toString());
27     }
28
29     rs.close();
30     stmt.close();
31     conn.close();
32 } catch (Exception e) {
33     err = e;
34     e.printStackTrace();
35 }
36 %>

```

6. 채팅방 입장

[talk_login2.jsp]

```
1  <%
2
3  if ( vo == null ){ // 해당 방이 없음
4      response.sendRedirect( ctxPath + "/talk_login.jsp" );
5  } else if ( vo.getApple().equals( pwd ) ) { // 해당 방의 apple pwd 입력했을 때
6      Cookie ck = new Cookie("level", "apple");
7      response.addCookie(ck);
8
9      Cookie ck2 = new Cookie("roomNo", String.valueOf(roomNo) );
10     response.addCookie(ck2);
11
12     response.sendRedirect( ctxPath + "/talk_view_apple.jsp" );
13 } else if ( vo.getBanana().equals( pwd ) ) { // 해당 방의 banana pwd 입력했을 때
14     Cookie ck = new Cookie("level", "banana");
15     response.addCookie(ck);
16
17     Cookie ck2 = new Cookie("roomNo", String.valueOf(roomNo) );
18     response.addCookie(ck2);
19
20     response.sendRedirect( ctxPath + "/talk_view_banana.jsp" );
21 } else if ( vo.getOrange().equals( pwd ) ) { // 해당 방의 orange pwd 입력했을 때
22     Cookie ck = new Cookie("level", "orange");
23     response.addCookie(ck);
24
25     Cookie ck2 = new Cookie("roomNo", String.valueOf(roomNo) );
26     response.addCookie(ck2);
27
28     response.sendRedirect( ctxPath + "/talk_view_orange.jsp" );
29 } else { // 알맞은 pwd 가 아님
30     response.sendRedirect( ctxPath + "/talk_login.jsp" );
31 }
32 %>
```

7. 채팅방에서 사용자 권한 판별

[talk_view_apple.jsp]

```
1  <%
2  String ctxPath = request.getContextPath();
3
4  Cookie[] cks = request.getCookies();
5  if ( cks == null ) {
6      response.sendRedirect( ctxPath + "talk_login.jsp" );
7      return;
8  }
9
10 int roomNo = -1;
```

```

11 String level = null;
12
13 for (Cookie ck : cks) {
14     if ( ck.getName().equals("roomNo") ) {
15         roomNo = Integer.parseInt( ck.getValue() );
16     } else if ( ck.getName().equals("level") ) {
17         level = ck.getValue();
18     }
19 }
20
21 if ( level == null || !level.equals("apple") ) {
22     response.sendRedirect( ctxPath + "talk_login.jsp" );
23     return;
24 }
25 %>

```

8. 채팅목록 불러오기

[talk_view_apple.jsp]

```

1  <% // 채팅목록 받아오기
2  List<TalkVO> r1 = new ArrayList<TalkVO>();
3  Exception err = null;
4
5  try {
6      Class.forName("oracle.jdbc.driver.OracleDriver");
7      Connection conn = DriverManager.getConnection(
8          "jdbc:oracle:thin:@127.0.0.1/XE", "HR", "hr" );
9
10     String sql = "SELECT * FROM talk_t WHERE room_no=? ORDER BY talk_no";
11     PreparedStatement stmt = conn.prepareStatement(sql);
12     stmt.setInt(1, roomNo);
13
14     ResultSet rs = stmt.executeQuery();
15     while( rs.next() ) {
16         TalkVO vo = new TalkVO();
17         vo.setTalkNo( rs.getInt("talk_no") );
18         vo.setRoomNo( rs.getInt("room_no") );
19         vo.setContent( rs.getString("content") );
20
21         r1.add( vo );
22     }
23
24     rs.close();
25     stmt.close();
26     conn.close();
27 } catch (Exception e) { err = e; }
28
29 if ( err != null ) {
30     response.sendRedirect( ctxPath + "/error.jsp" );
31 }
32 %>

```

9. 채팅 삭제하기

[talk_view_apple.jsp]

```
1 <% for (TalkVO vo: r1) { %>
2 <tr>
3     <td align="center"><%= vo.getTalkNo() %></td>
4     <td><%= vo.getContent() %></td>
5     <td align="center"><a href="talk_delete2.jsp?talk_no=<%= vo.getTalkNo()
%>">삭제</a></td>
6 </tr>
7 <% } %>
```

[talk_delete2.jsp] 삭제 권한 검사

```
1 <% // 삭제 권한 검사
2 String ctxPath = request.getContextPath();
3
4 //삭제 권한자: 해당 room 의 apple 권한자
5 int talkNo = Util.parseInt( request.getParameter("talk_no") );
6 int roomNo = -1;
7 String level = null;
8
9 Cookie[] cks = request.getCookies();
10 if ( cks == null ) {
11     response.sendRedirect(ctxPath + "/talk_login.jsp");
12 }
13
14 for (Cookie ck: cks) {
15     if ( ck.getName().equals("roomNo") ) roomNo = Util.parseInt(
16         ck.getValue() );
17     else if ( ck.getName().equals("level") ) level = ck.getValue();
18 }
19 if ( roomNo == -1 || level == null || !level.equals("apple") ) {
20     response.sendRedirect(ctxPath + "/talk_login.jsp");
21 }
22
23 %>
```

[talk_delete2.jsp] 삭제 수행

```
1 <% // DB 연결
2 Exception err = null;
3
4 try {
5     Class.forName("oracle.jdbc.driver.OracleDriver");
6     Connection conn = DriverManager.getConnection(
7         "jdbc:oracle:thin:@127.0.0.1/XE", "HR", "hr" );
8
9     String sql = "DELETE FROM talk_t WHERE room_no=? AND talk_no=?";
10    PreparedStatement stmt = conn.prepareStatement(sql);
11    stmt.setInt(1, roomNo);
```

```

12     stmt.setInt(2, talkNo);
13
14     stmt.executeUpdate();
15
16     stmt.close();
17     conn.close();
18 } catch ( Exception e ) {
19     err = e;
20 }
21
22 if (err != null) {
23     response.sendRedirect(ctxPath + "/error.jsp");
24 } else {
25     response.sendRedirect(ctxPath + "/talk_view_apple.jsp");
26 }
27 %>

```

10. 채팅 입력하기

[talk_view_apple.jsp]

```

1 <form method="POST" action="talk_add2.jsp">
2     <textarea rows="3" cols="60" name="content"></textarea>
3     <input type="submit" value="입력"/>
4 </form>

```

[utils/Util.java]

```

1 public class Util {
2     // 한글 인코딩 처리
3     public static String encodeUTF8( String str ) {
4         if (str == null || str == "") {
5             return str;
6         }
7
8         try {
9             byte[] bs = str.getBytes("8859_1");
10            str = new String( bs, "UTF-8" );
11        } catch ( Exception e ) {}
12
13        return str;
14    }
15 }

```

[talk_add2.jsp]

```

1 // 권한 판별 ...
2
3
4 <% // DB 연결

```

```

5      Exception err = null;
6      String content = Util.encodeUTF8(request.getParameter("content"));
7
8      try {
9          Class.forName("oracle.jdbc.driver.OracleDriver");
10         Connection conn =
11         DriverManager.getConnection("jdbc:oracle:thin:@127.0.0.1/XE", "HR", "hr");
12
13         String sql = "INSERT INTO talk_t VALUES (seq_talk.NEXTVAL, ?, ?)";
14         PreparedStatement stmt = conn.prepareStatement(sql);
15         stmt.setInt(1, roomNo);
16         stmt.setString(2, content);
17         stmt.executeUpdate();
18
19         stmt.close();
20         conn.close();
21     } catch (Exception e) {
22         err = e;
23     }
24
25     // 권한 별로 다른 redirect
26     if (err != null) {
27         response.sendRedirect(ctxPath + "/error.jsp");
28     } else if ( level.equals("apple") ) {
29         response.sendRedirect(ctxPath + "/talk_view_apple.jsp");
30     } else if ( level.equals("banana") ) {
31         response.sendRedirect(ctxPath + "/talk_view_banana.jsp");
32     } else {
33         response.sendRedirect(ctxPath + "/talk_login.jsp");
34     }
35 }
36 %>

```

11. banana view

apple 에서 권한검사랑 삭제버튼만 수정

[talk_view_banana.jsp]

```

1  <%@ page import="vo.TalkVO" %>
2  <%@ page import="java.util.*" %>
3  <%@ page import="java.sql.*" %>
4  <%@ page import="utils.Util" %>
5  <%@ page language="java" contentType="text/html; charset=utf-8"
6      pageEncoding="EUC-KR"%>
7  <% // 사용자 권한 판별
8      String ctxPath = request.getContextPath();
9
10     Cookie[] cks = request.getCookies();
11     if ( cks == null ) {
12         response.sendRedirect( ctxPath + "/talk_login.jsp" );
13         return;
14     }
15
16     int roomNo = -1;
17     String level = null;

```



```

18
19 for (Cookie ck : cks) {
20     if ( ck.getName().equals("roomNo") ) {
21         roomNo = Util.parseInt( ck.getValue() );
22     } else if ( ck.getName().equals("level") ) {
23         level = ck.getValue();
24     }
25 }
26
27 if ( level == null || !level.equals("banana") ) {
28     response.sendRedirect( ctxPath + "/talk_login.jsp" );
29     return;
30 }
31 %>
32
33 <% // 채팅목록 받아오기
34 List<TalkVO> r1 = new ArrayList<TalkVO>();
35 Exception err = null;
36
37 try {
38     Class.forName("oracle.jdbc.driver.OracleDriver");
39     Connection conn = DriverManager.getConnection(
40         "jdbc:oracle:thin:@127.0.0.1/XE", "HR", "hr" );
41
42     String sql = "SELECT * FROM talk_t WHERE room_no=? ORDER BY talk_no";
43     PreparedStatement stmt = conn.prepareStatement(sql);
44     stmt.setInt(1, roomNo);
45
46     ResultSet rs = stmt.executeQuery();
47     while( rs.next() ) {
48         TalkVO vo = new TalkVO();
49         vo.setTalkNo( rs.getInt("talk_no") );
50         vo.setRoomNo( rs.getInt("room_no") );
51         vo.setContent( rs.getString("content") );
52
53         r1.add( vo );
54     }
55
56     rs.close();
57     stmt.close();
58     conn.close();
59 } catch (Exception e) { err = e; }
60
61 if ( err != null ) {
62     response.sendRedirect( ctxPath + "/error.jsp" );
63 }
64 %>
65 <!DOCTYPE html>
66 <html>
67 <body>
68 <h1><%= roomNo %>번 채팅방 (level : <%= level %>)</h1>
69 <table border="1">
70     <tr>
71         <td width="80" align="center">번호</td>
72         <td width="420" align="center">내용</td>
73     </tr>
74     <% for (TalkVO vo: r1) { %>
75         <tr>

```

```

76         <td align="center"><%= vo.getTalkNo() %></td>
77         <td><%= vo.getContent() %></td>
78     </tr>
79     <% } %>
80 </table>
81 <br>
82 <form method="POST" action="talk_add2.jsp">
83     <textarea rows="3" cols="60" name="content"></textarea>
84     <input type="submit" value="입력"/>
85 </form>
86
87 </body>
88 </html>

```

12. orange view

banana view 에서 권한검사랑 글쓰기 기능 변경

[talk_view_orange.jsp]

```

1  <%@ page import="vo.TalkVO" %>
2  <%@ page import="java.util.*" %>
3  <%@ page import="java.sql.*" %>
4  <%@ page import="utils.Util" %>
5  <%@ page language="java" contentType="text/html; charset=utf-8"
6      pageEncoding="EUC-KR"%>
7  <% // 사용자 권한 판별
8  String ctxPath = request.getContextPath();
9
10 Cookie[] cks = request.getCookies();
11 if ( cks == null ) {
12     response.sendRedirect( ctxPath + "/talk_login.jsp" );
13     return;
14 }
15
16 int roomNo = -1;
17 String level = null;
18
19 for (Cookie ck : cks) {
20     if ( ck.getName().equals("roomNo") ) {
21         roomNo = Util.parseInt( ck.getValue() );
22     } else if ( ck.getName().equals("level") ) {
23         level = ck.getValue();
24     }
25 }
26
27 if ( level == null || !level.equals("orange") ) {
28     response.sendRedirect( ctxPath + "/talk_login.jsp" );
29     return;
30 }
31 %>
32
33 <% // 채팅목록 받아오기
34 List<TalkVO> r1 = new ArrayList<TalkVO>();
35 Exception err = null;

```

```

36
37 try {
38     Class.forName("oracle.jdbc.driver.OracleDriver");
39     Connection conn = DriverManager.getConnection(
40         "jdbc:oracle:thin:@127.0.0.1/XE", "HR", "hr" );
41
42     String sql = "SELECT * FROM talk_t WHERE room_no=? ORDER BY talk_no";
43     PreparedStatement stmt = conn.prepareStatement(sql);
44     stmt.setInt(1, roomNo);
45
46     ResultSet rs = stmt.executeQuery();
47     while( rs.next() ) {
48         TalkVO vo = new TalkVO();
49         vo.setTalkNo( rs.getInt("talk_no") );
50         vo.setRoomNo( rs.getInt("room_no") );
51         vo.setContent( rs.getString("content") );
52
53         rl.add( vo );
54     }
55
56     rs.close();
57     stmt.close();
58     conn.close();
59 } catch (Exception e) { err = e; }
60
61 if ( err != null ) {
62     response.sendRedirect( ctxPath + "/error.jsp" );
63 }
64 %>
65 <!DOCTYPE html>
66 <html>
67 <body>
68 <h1><%= roomNo %>번 채팅방 (level : <%= level %>)</h1>
69 <table border="1">
70     <tr>
71         <td width="80" align="center">번호</td>
72         <td width="420" align="center">내용</td>
73     </tr>
74     <% for (TalkVO vo: rl) { %>
75         <tr>
76             <td align="center"><%= vo.getTalkNo() %></td>
77             <td><%= vo.getContent() %></td>
78         </tr>
79     <% } %>
80 </table>
81
82 </body>
83 </html>

```

Attribute

JSP 페이지 사이에서 정보를 주고 받거나 공유하기 위해 사용함.

각각의 기본객체가 존재하는 동안 사용할 수 있다.

기본객체 `Session`, `Request`, `Application`, `PageContext` 는

- `void setAttribute(String, Object)`
- `Object getAttribute(String)`
- `void removeAttribute(String)` 을 지원한다.

EL (Expression Language)

간단한 연산을 출력할 수 있는 JSP 언어. (빠른 디자인에 유용)

웹서버에서 동작 -> 웹브라우저에서 출력

`isELIgnored="true"` -> EL을 단순 문자열로 인식

기본 문법

- 사용법: `${ i }` (i 는 이름)
 - 각종 연산자 사용 가능

```
1  [산술연산]
2  ${ 100 + 100 } <!-- 200 --%>
3
4  [비교연산]
5  ${ 50 > 100 } <!-- false --%>
6  ${ 50 gt 100 } <!-- false --%> gt : >
7  ${ 5 == 100 } <!-- false --%>
8
9  [논리연산]
10 ${ true && false } <!-- false --%>
11 ${ not(true) } <!-- false --%>
12
13 [삼항연산]
14 ${ (100>50)?'apple':'banana' } <!-- apple --%>
```

`setAttribute` 된 값은 EL로 가져올 수 있다.

```
1  <%
2  request.setAttribute("apple", 100); // String key, Object value
3  request.setAttribute("arr", new int[]{1, 2, 3, 4});
4  %>
5
6  ${ apple }
7  ${ apple > 10 }
8  ${ arr[0] }
```

java 의 static 함수를 EL 에서 사용할 수 있다.

-> 이렇게 하면 복잡한 코드도 구현할 수는 있으나, 그냥 JSTL 이랑 같이 쓰자

태그

역사

- SGML 에서 태그를 처음으로 도입
- 미 국방성에 문서를 전산화 할 수 있는 언어를 개발했다.
- MathML, MusicML, postscript 등 파생된 서브셋이 엄청 많다.
- postscript가 발전해서 pdf로 발전
- 파생된 것들 중 가장 성공한 것이 XML, HTML 이다.

특징

1. 최 외각의 태그는 단 하나만 존재해야 한다.
 - `<html> ... </html>`
2. 태그는 중첩해서 선언하지 않는다.
 - `<banana> <apple> </apple> </banana>` : OK
 - `<banana> <apple> </banana> </apple>` : NO
3. 속성은 시작태그에 선언한다.
 - `<body bgColor="red"></body>`
4. 속성은 `key="value"` 혹은 `key='value'` 로 선언되어야 한다.
단, `"` 혹은 `'` 간의 중복은 불가능하다.
5. 하나의 태그에서 속성은 두 번 선언하지 않는다.
 - `<body bgColor="red" bgColor="green"></body>` : NO
6. 시작태그가 있으면 반드시 끝 태그가 있다.
 - `<apple></apple>` 과 같이 내용이 없는 경우, `<apple/>` 로 쓴다.

JSTL

태그 기반 언어로, EL과 연동하여 EL의 기능을 확장하였다.

기본 문법

분류	형태 (닫는 태그 생략)
for 문	<code><prefix:forEach var="i" begin="0" end="4"></code>
forEach문	<code><prefix:forEach varStatue="idx" var="vo" items="\${ 1st }"></code>
print	<code><prefix:out value="\${ i }"/></code>
조건문 영역	<code><prefix:choose></code>
if 문	<code><prefix:when test="\${ 조건 }"></code>
else 문	<code><prefix:otherwise></code>

```

1 <%@ page language="java" contentType="text/html; charset=utf-8"
2   pageEncoding="EUC-KR" isELIgnored="false"%>
3 <!-- tag 라이브러리를 가져오고, prefix를 원하는대로 지정 --%>
4 <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="sso" %>
5
6 <%
7 List<String> r1 = new ArrayList<String>();
8 r1.add("one");
9 r1.add("two");
10 r1.add("three");
11 r1.add("four");
12 r1.add("five");
13
14 request.setAttribute("r1", r1);
15 %>
16
17 <!DOCTYPE html>
18 <html>
19 <body>
20   <!-- for 문 --%>
21   <sso:forEach var="i" begin="0" end="4">
22     <!-- 해당 name 의 값 출력 --%>
23     <sso:out value="${ i }"/>
24
25     <!-- 조건문 --%>
26     <sso:choose>
27       <!-- if 문 --%>
28       <sso:when test="${ (i%3) == 0 }">APPLE</sso:when>
29       <sso:when test="${ (i%3) == 1 }">BANANA</sso:when>
30       <!-- else 문 --%>
31       <sso:otherwise>ORANGE</sso:otherwise>
32     </sso:choose>
33   </sso:forEach>
34
35   <sso:forEach var="vo" items="${ r1 }">
36     ${ vo }
37   </sso:forEach>
38 </body>
39 </html>

```

[방명리스트]

```

1 <%@ page language="java" contentType="text/html; charset=utf-8"
2   pageEncoding="utf-8" import="java.util.List, bangmyung.*" %>
3 <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="sso" %>
4 <%
5 ...
6 // 3. 흐름 만들기
7 if ( r1 == null || err != null ) {
8   response.setCharacterEncoding("utf-8");
9   response.sendRedirect("/study3/error.jsp");
10 } else {
11   request.setAttribute("lst", r1);

```

```
12 %>
13 <!DOCTYPE>
14 <html>
15 <body>
16 <table border='1'>
17     <tr>
18         <td>No</td>
19         <td>글</td>
20         <td>작성시간</td>
21     </tr>
22
23     <sso:forEach varStatus="vs" var="vo" items="${ lst }">
24         <tr bgColor="${ (vs.count % 2 != 0)?'#aabbcc':'#bbccdd' }">
25             <td>${ vs.count }</td>
26             <td>${ vo.no }</td>
27             <td>${ vo.gul }</td>
28             <td>${ vo.theTime }</td>
29         </tr>
30     </sso:forEach>
31 </table>
32 <form method="POST" action="bangmyung_add.jsp">
33     <input type="text" name="gul" size="50"/>
34     <input type="submit"/>
35 </form>
36 <% } %>
37 </body>
38 </html>
```