

〈13주차 실습〉

웹 프로그래밍

Web Programming

프로젝트 생성 완료 후 해당 프로젝트 디렉토리 구조

appli 프로젝트 폴더 구조

appli/ 구조

appli			
이름	수정한 날짜	유형	크기
META-INF	2017-11-29 오전...	파일 폴더	
WEB-INF	2017-11-29 오후...	파일 폴더	
AnimalServlet	2017-11-28 오후...	Chrome HTML D...	1KB

appli/WEB-INF 구조

appli > WEB-INF			
이름	수정한 날짜	유형	크기
classes	2017-11-29 오후...	파일 폴더	
lib	2017-11-29 오전...	파일 폴더	

1. 프로젝트 파일 war 파일로 압축을 위해 해당 디렉토리로 이동
2. (WEB-INF 상위 디렉토리에서 압축) `jar cvf [압축명.war] *`

```
C:\Users\hong>cd C:\Users\hong\Desktop\appli
```

```
C:\Users\hong\Desktop\appli>jar cvf appli.war *
```

Manifest를 추가함

추가하는 중: AnimalServlet.html(입력 = 464) (출력 = 267)(42%를 감소함)

META-INF/ 항목을 무시하는 중

META-INF/MANIFEST.MF 항목을 무시하는 중

추가하는 중: WEB-INF/(입력 = 0) (출력 = 0)(0%를 저장함)

추가하는 중: WEB-INF/classes/(입력 = 0) (출력 = 0)(0%를 저장함)

추가하는 중: WEB-INF/classes/AnimalServlet.class(입력 = 1904) (출력 = 1046)

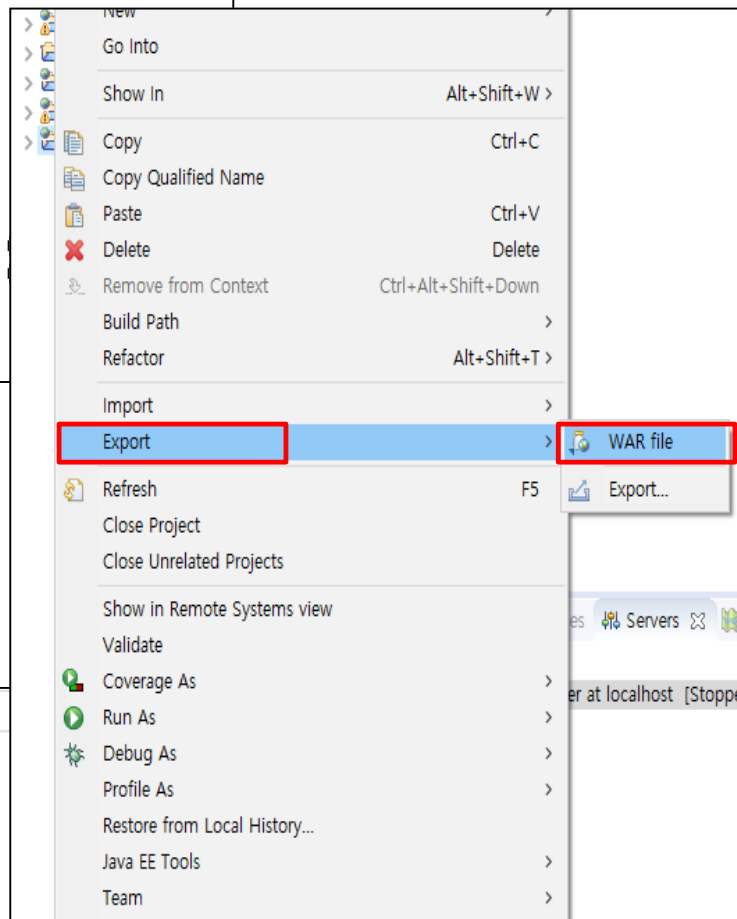
추가하는 중: WEB-INF/classes/ResultServlet.class(입력 = 1994) (출력 = 1086)

추가하는 중: WEB-INF/lib/(입력 = 0) (출력 = 0)(0%를 저장함)

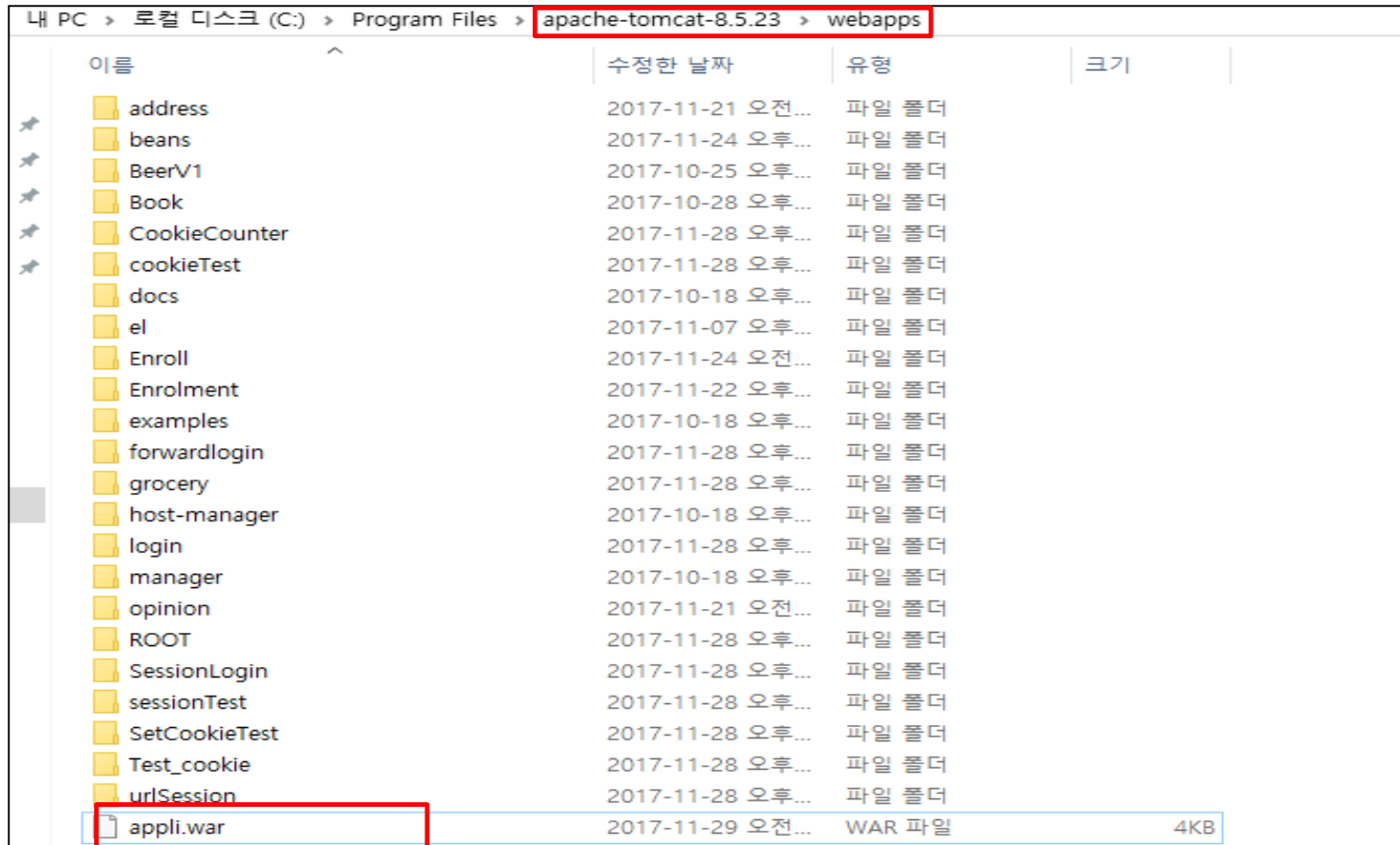
```
C:\Users\hong\Desktop\appli>
```

압축 파일 생성

appli				
이름	수정한 날짜	유형	크기	
META-INF	2017-11-29 오전...	파일 폴더		
WEB-INF	2017-11-29 오후...	파일 폴더		
AnimalServlet	2017-11-28 오후...	Chrome HTML D...		
appli.war	2017-11-29 오후...	WAR 파일	4KB	



3. Webapps 디렉토리에 배포파일(war 파일) 복사하여 배포



4. 접속 경로 설정을 위해 Tomcat/conf/server.xml 수정

내 PC > 로컬 디스크 (C:) > Program Files > apache-tomcat-8.5.23 > conf			
이름	수정한 날짜	유형	크기
Catalina	2017-10-18 오후...	파일 폴더	
catalina.policy	2017-09-28 오전...	POLICY 파일	14KB
catalina.properties	2017-09-28 오전...	PROPERTIES 파일	8KB
context	2017-09-28 오전...	XML 문서	2KB
jaspic-providers	2017-09-28 오전...	XML 문서	2KB
jaspic-providers.xsd	2017-09-28 오전...	XML 스키마 파일	3KB
logging.properties	2017-09-28 오전...	PROPERTIES 파일	4KB
server	2017-11-28 오후...	XML 문서	8KB
tomcat-users	2017-09-28 오전...	XML 문서	3KB
tomcat-users.xsd	2017-09-28 오전...	XML 스키마 파일	3KB
web	2017-09-28 오전...	XML 문서	170KB

5. server.xml 의 Host 정보 아래에 다음 추가

6. contextPath: 서버이름(hostname) 바로 다음에 오는 경로.
여기선 path를 "/"로 설정했으므로 contextPath는 없다고 보면 됨



```
server - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

<Realm className="org.apache.catalina.realm.LockOutRealm">
  <!-- This Realm uses the UserDatabase configured in the global JNDI
        resources under the key "UserDatabase". Any edits
        that are performed against this UserDatabase are immediately
        available for use by the Realm. -->
  <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
        resourceName="UserDatabase"/>
</Realm>

<Host name="localhost" appBase="webapps"
      unpackWARs="true" autoDeploy="true">
  <Context path="/" docBase="appli" reloadable="true" />
  <!-- SingleSignOn valve, share authentication between web applications
        Documentation at: /docs/config/valve.html -->
  <!--
  <Valve className="org.apache.catalina.authenticator.SingleSignOn" />
  -->

  <!-- Access log processes all example.
        Documentation at: /docs/config/valve.html
        Note: The pattern used is equivalent to using pattern="common" -->
  <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
        prefix="localhost_access_log" suffix=".txt"
        pattern="%h %l %u %t &quot;%r&quot; %s %b" />

</Host>
</Engine>
</Service>
</Server>
```

- 7. 톰캣 실행
- 8. 배포 완료

내 PC > 로컬 디스크 (C:) > Program Files > apache-tomcat-8.5.23 > webapps

이름	수정한 날짜	유형	크기
address	2017-11-21 오전...	파일 폴더	
appli	2017-11-29 오후...	파일 폴더	
beans	2017-11-24 오후...	파일 폴더	
BeerV1	2017-10-25 오후...	파일 폴더	
Book	2017-10-28 오후...	파일 폴더	
CookieCounter	2017-11-28 오후...	파일 폴더	
cookieTest	2017-11-28 오후...	파일 폴더	
docs	2017-10-18 오후...	파일 폴더	
el	2017-11-07 오후...	파일 폴더	
Enroll	2017-11-24 오전...	파일 폴더	
Enrolment	2017-11-22 오후...	파일 폴더	
examples	2017-10-18 오후...	파일 폴더	
forwardlogin	2017-11-28 오후...	파일 폴더	
grocery	2017-11-28 오후...	파일 폴더	
host-manager	2017-10-18 오후...	파일 폴더	
login	2017-11-28 오후...	파일 폴더	
manager	2017-10-18 오후...	파일 폴더	
opinion	2017-11-21 오전...	파일 폴더	
ROOT	2017-11-28 오후...	파일 폴더	
SessionLogin	2017-11-28 오후...	파일 폴더	
sessionTest	2017-11-28 오후...	파일 폴더	
SetCookieTest	2017-11-28 오후...	파일 폴더	
Test_cookie	2017-11-28 오후...	파일 폴더	
urlSession	2017-11-28 오후...	파일 폴더	
appli.war	2017-11-29 오전...	WAR 파일	4KB

- Dump 파일 생성하기

- 특정 데이터베이스만

- Mysql -u [user] -p databasename > [경로]파일명.sql

```
C:\Users\User>mysql dump -u dbtest -p dump_test > C:\Users\User\Desktop\db_dump.sql  
Enter password: *****
```

- 특정 테이블만

- Mysql -u [user] -p databasename tablename > [경로]파일명.sql

```
C:\Users\User>mysql dump -u dbtest -p dump_test dept > C:\Users\User\Desktop\tbl_dump.sql  
Enter password: *****
```

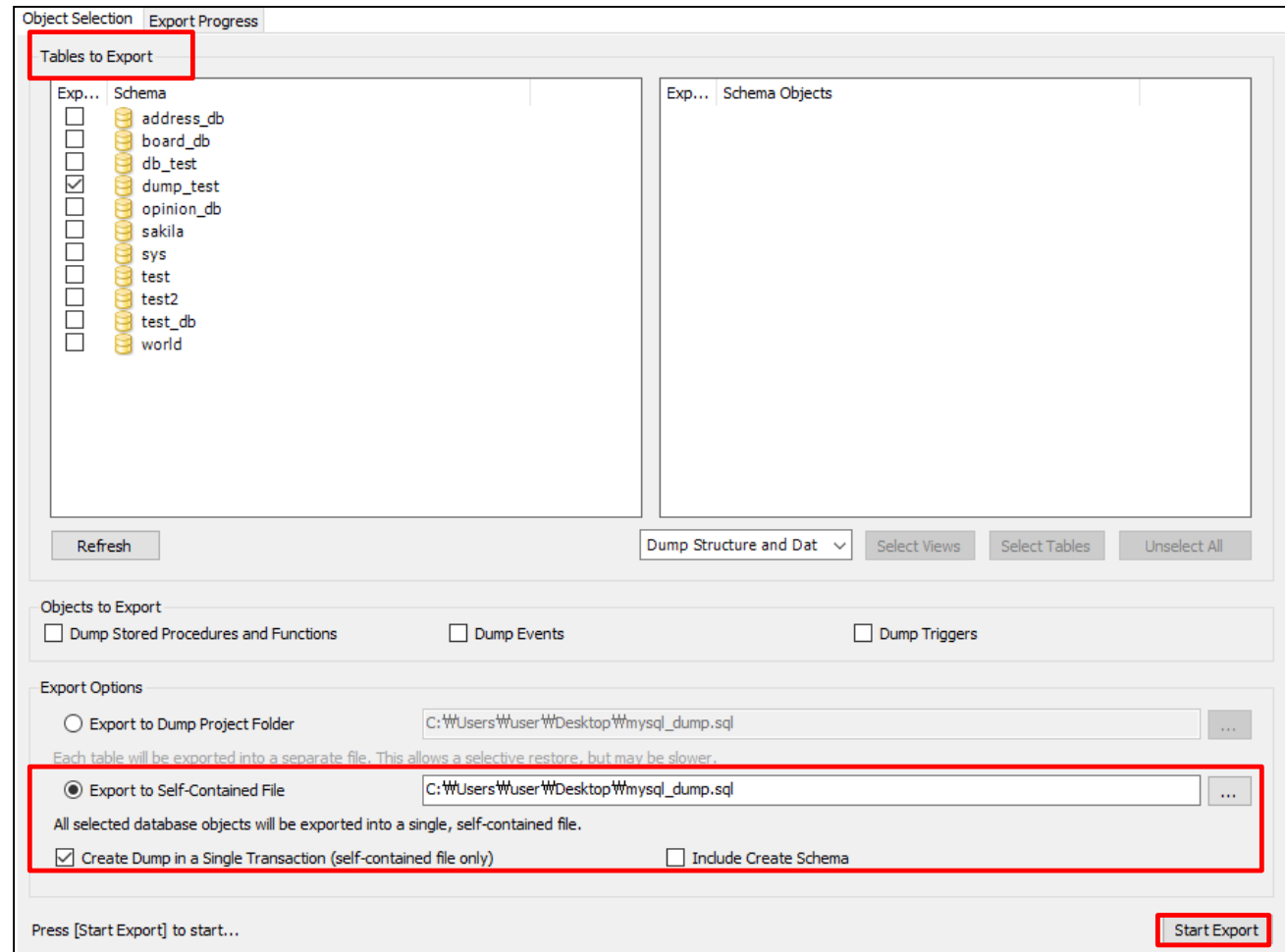
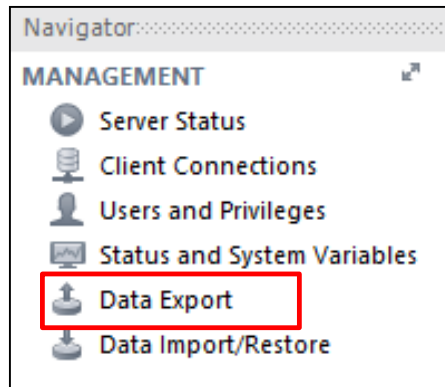
- 특정 데이터만

- mysql -u [user] -p databasename tablename -w'[조건]' > [경로]파일명.sql

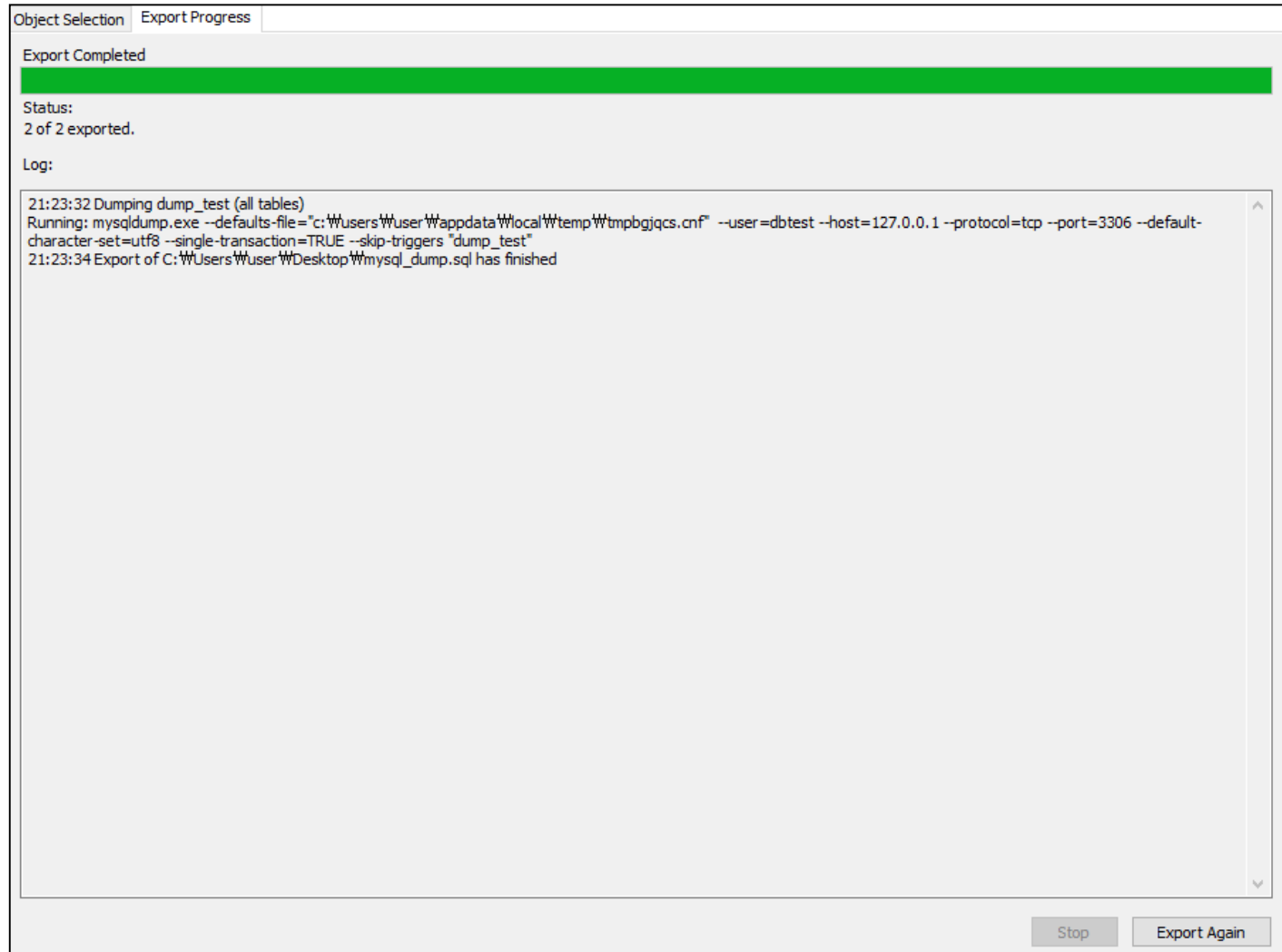
Database dump (2/4)

8/29

- Workbench에서 Dump 파일 생성하기
 - Data Export → Table to Export, 경로 설정 → Start Export



- Workbench에서 Dump 파일 생성하기
 - 생성 완료



- Dump 파일 사용하기

- `mysql -u [user] -p databasename < [경로]파일명.sql`

```
C:\Users\User>mysql -u dbtest -p test < C:\Users\User\Desktop\db_dump.sql
Enter password: *****
```

결과

```
mysql> use test;
Database changed
mysql> show tables;
+-----+
| Tables_in_test |
+-----+
| dept            |
| shop            |
+-----+
2 rows in set (0.00 sec)

mysql> desc shop;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| SHOP_NO        | int(11)       | NO   | PRI | NULL    |       |
| SHOP_NAME      | varchar(30)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> select * from shop;
+-----+-----+
| SHOP_NO | SHOP_NAME |
+-----+-----+
| 100     | 동대문상점1 |
| 200     | 동대문상점2 |
| 300     | 동대문상점3 |
| 400     | 동대문상점4 |
+-----+-----+
4 rows in set (0.00 sec)
```

아래의 코드를 이용하여 JSP 파일을 작성하시오.

StoreCookies.jsp

```
<%@ page language="java" contentType="text/html; charset=utf-8"
pageEncoding="utf-8"%>
<%
    response.addCookie(new Cookie("NAME", "John"));
    response.addCookie(new Cookie("GENDER", "Male"));
    response.addCookie(new Cookie("AGE", "20"));
%>
<html>
<head>
<title>Insert title here</title>
</head>
<body>
    쿠키 저장 완료<br>
</body>
</html>
```

쿠키 생성과 동시에 쿠키값 지정

아래의 코드를 이용하여 JSP 파일을 작성하시오.

ReadCookies.jsp

```
<%@ page language="java" contentType="text/html; charset=utf-8" pageEncoding="utf-8"%>
<% Cookie[] cookies = request.getCookies(); %>
<html>
<head>
</head>
<body>
    이름 : <%= getCookieValue(cookies, "NAME") %> <BR>
    성별 : <%= getCookieValue(cookies, "GENDER") %> <BR>
    나이 : <%= getCookieValue(cookies, "AGE") %>
</body>
</html>
<%!
    private String getCookieValue(Cookie[] cookies, String name){
        if(cookies == null)
            return null;
        for(Cookie cookie : cookies){
            if(cookie.getName().equals(name))
                return cookie.getValue();
        }
        return null;
    }
%>
```

해당 쿠키에서 값을 가져옴

쿠키값을 가져옴

아래의 코드를 이용하여 JSP 파일을 작성하시오.

ModifyCookies.jsp

```
<%@ page language="java" contentType="text/html; charset=utf-8" pageEncoding="utf-8"%>
<%
    response.addCookie(new Cookie("AGE", "16"));
%>
<html>
<head>
</head>
<body>
    수정 완료
</body>
</html>
```

기존 쿠키값 변경

아래의 코드를 이용하여 JSP 파일을 작성하시오.

DeleteCookies.jsp

```
<%@ page language="java" contentType="text/html; charset=utf-8" pageEncoding="utf-8"%>
<%
    Cookie cookie = new Cookie("GENDER", "");
    cookie.setMaxAge(0);
    response.addCookie(cookie);
%>
<html>
<head>
</head>
<body>
    쿠키 삭제 완료<BR>
</body>
</html>
```

특정 쿠키만 삭제

쿠키 유효시간 만료 시킴

응답 헤더에 추가

쿠키 예제 (5/5)

15/29

실행 결과

StoreCookies.jsp

localhost:8080/cookies_test/StoreCookies.jsp

쿠키 저장 완료

ReadCookies.jsp

localhost:8080/cookies_test/ReadCookies.jsp

이름 : John
성별 : Male
나이 : 20

ModifyCookies.jsp

localhost:8080/cookie_test/ModifyCookies.jsp

수정 완료

ReadCookies.jsp

localhost:8080/cookies_test/ReadCookies.jsp

이름 : John
성별 : Male
나이 : 16

DeleteCookies.jsp

localhost:8080/cookies_test/DeleteCookies.jsp

쿠키 삭제 완료

ReadCookies.jsp

localhost:8080/cookies_test/ReadCookies.jsp

이름 : John
성별 : null
나이 : 16

아래의 코드를 이용하여 html 파일을 작성하시오.

sessionLogin.html

```
<html>
<body>
<h2>Session생성 Test</h2>
<form action="sessionCreate.jsp" method="post">
id : <input type="textField" name="id">
<input type="submit" value="확인">
</form>
</body>
</html>
```

아래의 코드를 이용하여 jsp 파일을 작성하시오.

sessionCreate.jsp

```
<%@page contentType="text/html; charset=utf-8" session="false" %>
<html>
<body>
<h2>Session 생성 Test</h2>
<%
String id=request.getParameter("id");
if(id.equals("")){
response.sendRedirect("sessionLogin.html");
return;
}
HttpSession session=request.getSession(true);
if(session.isNew()){
session.setAttribute("id",id);
out.print("새로운 세션이 생성되었습니다. <br>");
out.print("세션 ID: " + session.getId());
}
else{
%>
이미 세션이 존재합니다.<br>
세션ID :<%=session.getId() %>
<%}%>
</body>
</html>
```

세션에 대한 접근

세션 속성값 지정

아래의 코드를 이용하여 jsp 파일을 작성하시오.

sessionRead.jsp

```
<%@page contentType="text/html; charset=utf-8" session="false"%>
<html><body>
<%
HttpSession session = request.getSession(false);
String s = (String)session.getAttribute("id");
if(s==null){
out.print("설정된 세션이 없습니다.<br>");
}else{
out.print("Attribute id 값 : "+s+"<br>");
}
%>
isNew():<%=session.isNew()%><br>
세션ID:<%=session.getId() %><br>
세션생성시간:<%=new java.util.Date(session.getCreationTime()).toString()
%><br>
세션마지막접속시간:<%=new
java.util.Date(session.getLastAccessedTime()).toString() %>
세션Active시간:<%=session.getMaxInactiveInterval() %>sec<br>
<a href="sessionDelete.jsp">session 삭제하기</a>
</body></html>
```

세션 속성값 조회

세션 생성 시간 조회

세션 최근 접속 시간 조회

세션 액티브 시간 조회

아래의 코드를 이용하여 jsp 파일을 작성하시오.

sessionDelete.jsp

```
<%@page contentType="text/html; charset=utf-8" session="false"%>
<% HttpSession session = request.getSession(false);
if(session != null){
    session.invalidate(); 세션 종료
}
%>
<h3> 세션이 삭제 되었습니다. </h3>
<a href = "sessionLogin.html">로그인 페이지로</a>
```

실행 결과

sessionLogin.html

← → ↻ 🏠 ⓘ localhost:8080/sessionTest/sessionLogin.html

Session생성 Test

id :

sessionCreate.jsp

← → ↻ 🏠 ⓘ localhost:8080/sessionTest/sessionCreate.jsp

Session생성 Test

새로운 세션이 생성되었습니다.
세션 ID: 77B9657B11F14F9ABE8B4CB380BD563D

sessionLogin.html

← → ↻ 🏠 ⓘ localhost:8080/sessionTest/sessionLogin.html

Session생성 Test

id :

sessionCreate.jsp

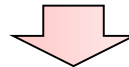
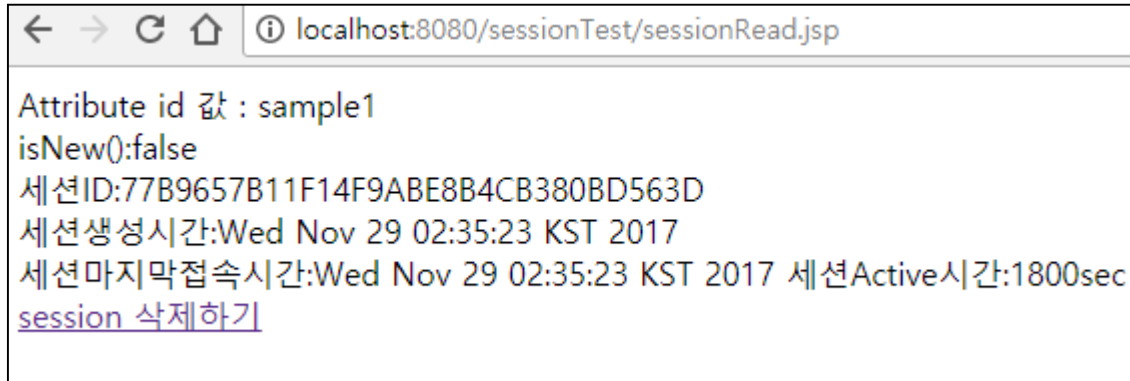
← → ↻ 🏠 ⓘ localhost:8080/sessionTest/sessionCreate.jsp

Session생성 Test

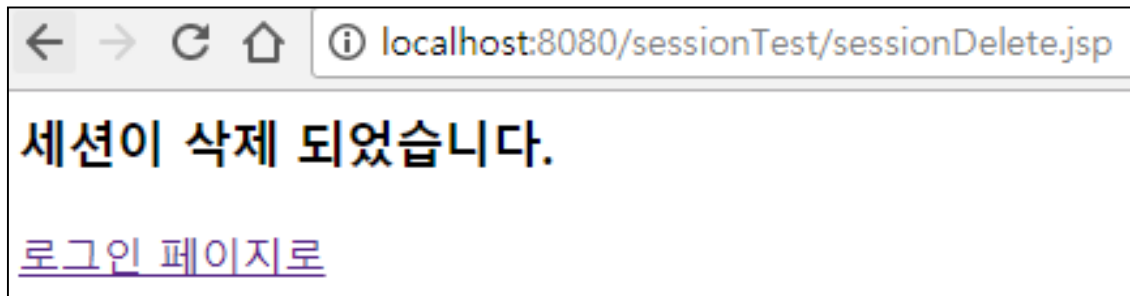
이미 세션이 존재합니다.
세션 ID: 77B9657B11F14F9ABE8B4CB380BD563D

실행 결과

sessionRead.jsp



sessionDelete.jsp



AnimalServlet.html

```
<HTML>
  <HEAD>
    <META http-equiv= "Content-Type " content=
"text/html; charset=utf-8 ">
    <TITLE>세션 예제</TITLE>
  </HEAD>
  <BODY>
    <H3>좋아하는 음식은?</H3>
    <FORM ACTION=animal>
      <INPUT TYPE=TEXTFIELD NAME=FOOD>
      <INPUT TYPE=SUBMIT VALUE= ' 확인' >
    </FORM>
  </BODY>
</HTML>
```

AnimalServlet.java

```
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import java.io.*;

@WebServlet(name = "animal", urlPatterns = { "/animal" })
public class AnimalServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    public AnimalServlet() {
        super();
        // TODO Auto-generated constructor stub
    }

    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws IOException, ServletException {
        String food = request.getParameter("FOOD");
        HttpSession session = request.getSession();
        session.setAttribute("FOOD", food);
        response.setContentType("text/html;charset=utf-8");
        PrintWriter out = response.getWriter();
        out.println("<HTML>");
        out.println("<HEAD><TITLE>세션 예제</TITLE></HEAD>");
        out.println("<BODY>");
        out.println("<H3> 좋아하는 동물은? </H3>");
        out.println("<FORM ACTION=result>");
        out.println("<INPUT TYPE=TEXTFIELD NAME=ANIMAL>");
        out.println("<INPUT TYPE=SUBMIT VALUE='확인'>");
        out.println("</FORM>");
        out.println("</BODY>");
        out.println("</HTML>");
    }
}
```


ResultServlet.java

```
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import java.io.*;

@WebServlet(name = "result", urlPatterns = { "/result" })
public class ResultServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    public ResultServlet() {
        super();
    }

    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws IOException, ServletException {
        HttpSession session = request.getSession();
        String food = (String) session.getAttribute("FOOD");
        String animal = request.getParameter("ANIMAL");
        session.invalidate();
        response.setContentType("text/html;charset=utf-8");
        PrintWriter out = response.getWriter();
        out.println("<HEAD><TITLE>세션 예제</TITLE></HEAD>");
        out.println("<BODY>");
        out.println("<H3>음식, 동물 조사 결과 결과</H3>");
        out.printf("당신은 음식은 %s, 동물은 %s을 좋아합니다. ", food, animal);
        out.println("</BODY>");
        out.println("</HTML>");
    }
}
```

실행 결과

AnimalServlet.html

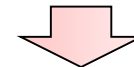
← → ↻ ⬆ ⓘ localhost:8080/ttt/AnimalServlet.html

좋아하는 음식은?

AnimalServlet.java

← → ↻ ⬆ ⓘ localhost:8080/ttt/animal?FOOD=밥

좋아하는 동물은?



ResultServlet.java

← → ↻ ⬆ ⓘ localhost:8080/ttt/result?ANIMAL=소

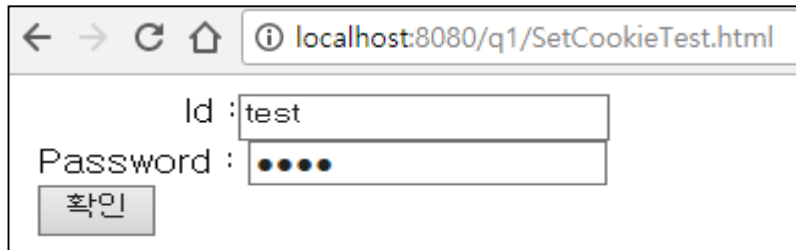
음식, 동물 조사 결과 결과

당신은 음식은 밥, 동물은 소를 좋아합니다.

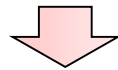
Q1> 다양한 쿠키 메소드를 사용하여 아래와 같은 화면을 가지는 파일들을 작성하시오.

실행 결과

SetCookisTest.html

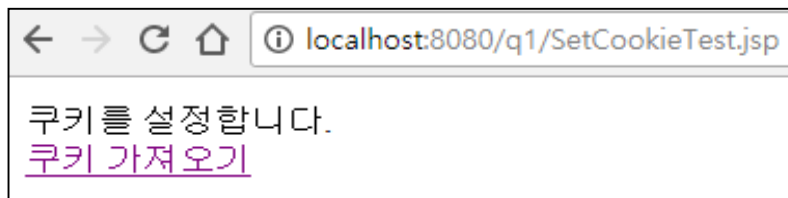


A web browser window showing a form at localhost:8080/q1/SetCookieTest.html. The form has two input fields: 'Id' with the value 'test' and 'Password' with four dots. Below the fields is a button labeled '확인' (Confirm).



쿠키가 없을 경우

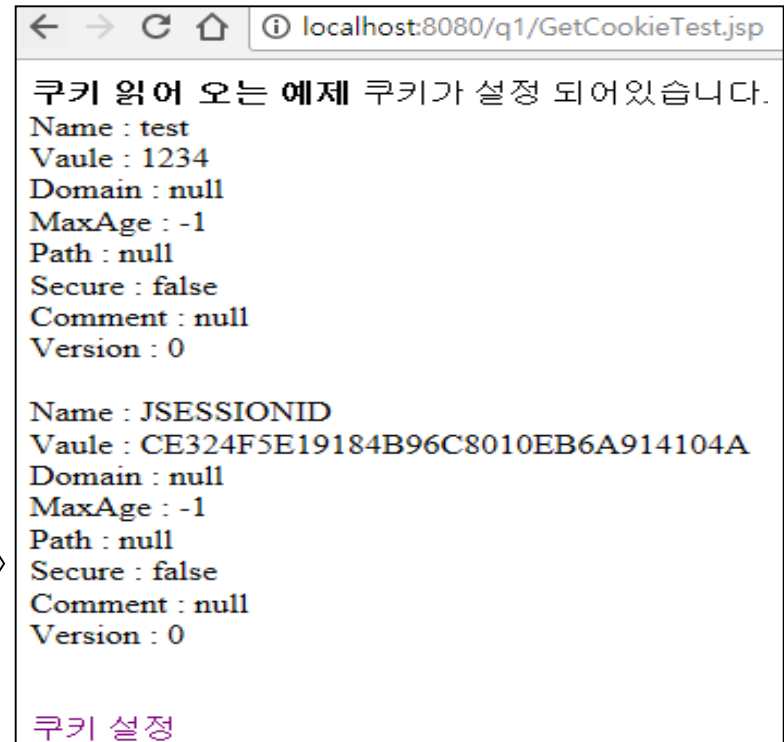
SetCookisTest.jsp



A web browser window showing the text '쿠키를 설정합니다.' (Set cookies) and a link '쿠키 가져오기' (Get cookies) in purple. The address bar shows localhost:8080/q1/SetCookieTest.jsp.



GetCookisTest.jsp



A web browser window showing the text '쿠키 읽어 오는 예제 쿠키가 설정 되어있습니다.' (Example of reading cookies, cookies are set). Below this is a list of cookie details for 'test' and 'JSESSIONID'. At the bottom, there is a link '쿠키 설정' (Set cookies) in purple. The address bar shows localhost:8080/q1/GetCookieTest.jsp.

쿠키 읽어 오는 예제 쿠키가 설정 되어있습니다.

Name : test
Vaule : 1234
Domain : null
MaxAge : -1
Path : null
Secure : false
Comment : null
Version : 0

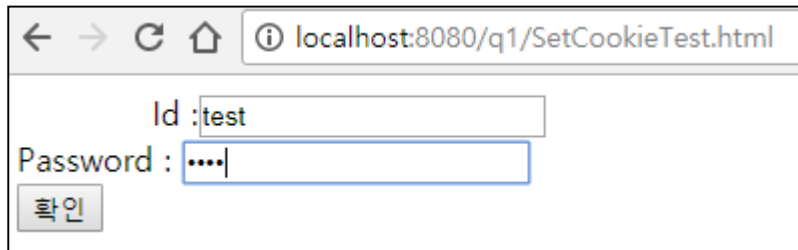
Name : JSESSIONID
Vaule : CE324F5E19184B96C8010EB6A914104A
Domain : null
MaxAge : -1
Path : null
Secure : false
Comment : null
Version : 0

쿠키 설정

Q1> 다양한 쿠키 메소드를 사용하여 아래와 같은 화면을 가지는 파일들을 작성하시오.

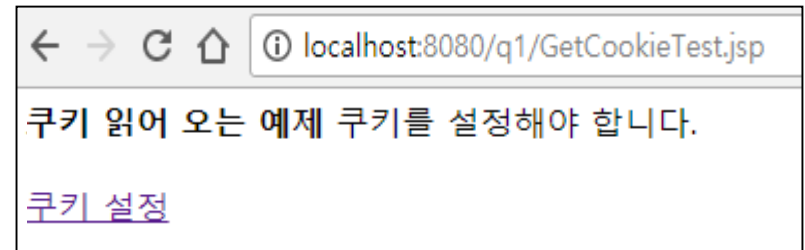
실행 결과

SetCookisTest.html



A web browser window showing the URL `localhost:8080/q1/SetCookieTest.html`. The page contains a form with two input fields: "Id :test" and "Password :". Below the password field is a button labeled "확인" (Confirm).

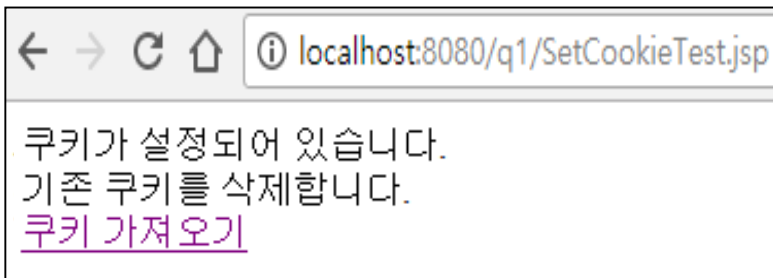
GetCookisTest.jsp



A web browser window showing the URL `localhost:8080/q1/GetCookieTest.jsp`. The page displays the text "쿠키 읽어 오는 예제 쿠키를 설정해야 합니다." (Example of reading cookies, you must set cookies). Below this text is a purple link labeled "쿠키 설정" (Set Cookie).

↓ 쿠키가 있을 경우

SetCookisTest.jsp



A web browser window showing the URL `localhost:8080/q1/SetCookieTest.jsp`. The page displays the text "쿠키가 설정되어 있습니다." (Cookies are set) and "기존 쿠키를 삭제합니다." (Delete existing cookies). Below this text is a purple link labeled "쿠키 가져오기" (Get Cookie).

Q2> URL Rewriting을 활용하여 아래와 같은 화면을 가지는 파일들을 작성하시오. (response.encodeURL())

urlSession.html

← → ↻ 🏠 ⓘ localhost:8080/urlSession/urlSession.html ☆ ⓘ T ⋮

Session생성 Test

id :

urlSession.jsp

← → ↻ 🏠 ⓘ localhost:8080/urlSession/urlSession.jsp ☆ ⓘ T ⋮

Session URL Rewriting Test

세션ID:149A326C06FA7F9DC29A83F9C9226EA2
ID : sample1

[Test1](#)
[Test2](#)

Test1 클릭

Test2 클릭

← → ↻ 🏠 ⓘ localhost:8080/urlSession/urlSessionResult.jsp?sessionId=149A326C06FA7F9DC29A83F9C9226EA2 ☆ ⓘ T ⋮

Session URL Rewriting Result

세션ID:149A326C06FA7F9DC29A83F9C9226EA2
ID : sample1

urlSessionResult.jsp

← → ↻ 🏠 ⓘ localhost:8080/urlSession/urlSessionResult.jsp ☆ ⓘ T ⋮

Session URL Rewriting Result

세션ID:149A326C06FA7F9DC29A83F9C9226EA2
ID : sample1

urlSessionResult.jsp

- 제출 방식 : E-Class를 통하여 제출
- 제출 내용 : 프로젝트 파일을 압축한 war파일
- 제출 형식 : ZIP File
- 제목 형식 : 학번_이름_주차
 - ex) 학번_홍길동_13주차.zip
- 제출 기한 : 11월 27일(토) 23시 59분까지
 - 연장 제출 불허