

Web2.0



XML

준비

1. context.xml(Meta-inf)

<Context>

<Resource name="jdbc/OracleDB"

auth="Container"

type="javax.sql.DataSource"

username="scott"

password="tiger"

driverClassName="oracle.jdbc.driver.OracleDriver"

factory="org.apache.tomcat.dbcp.dbcp.BasicDataSourceFactory"

url="jdbc:oracle:thin:@127.0.0.1:1521:ORCL"

maxActive="500"

maxIdle="100"/>

</Context>

2. Lib

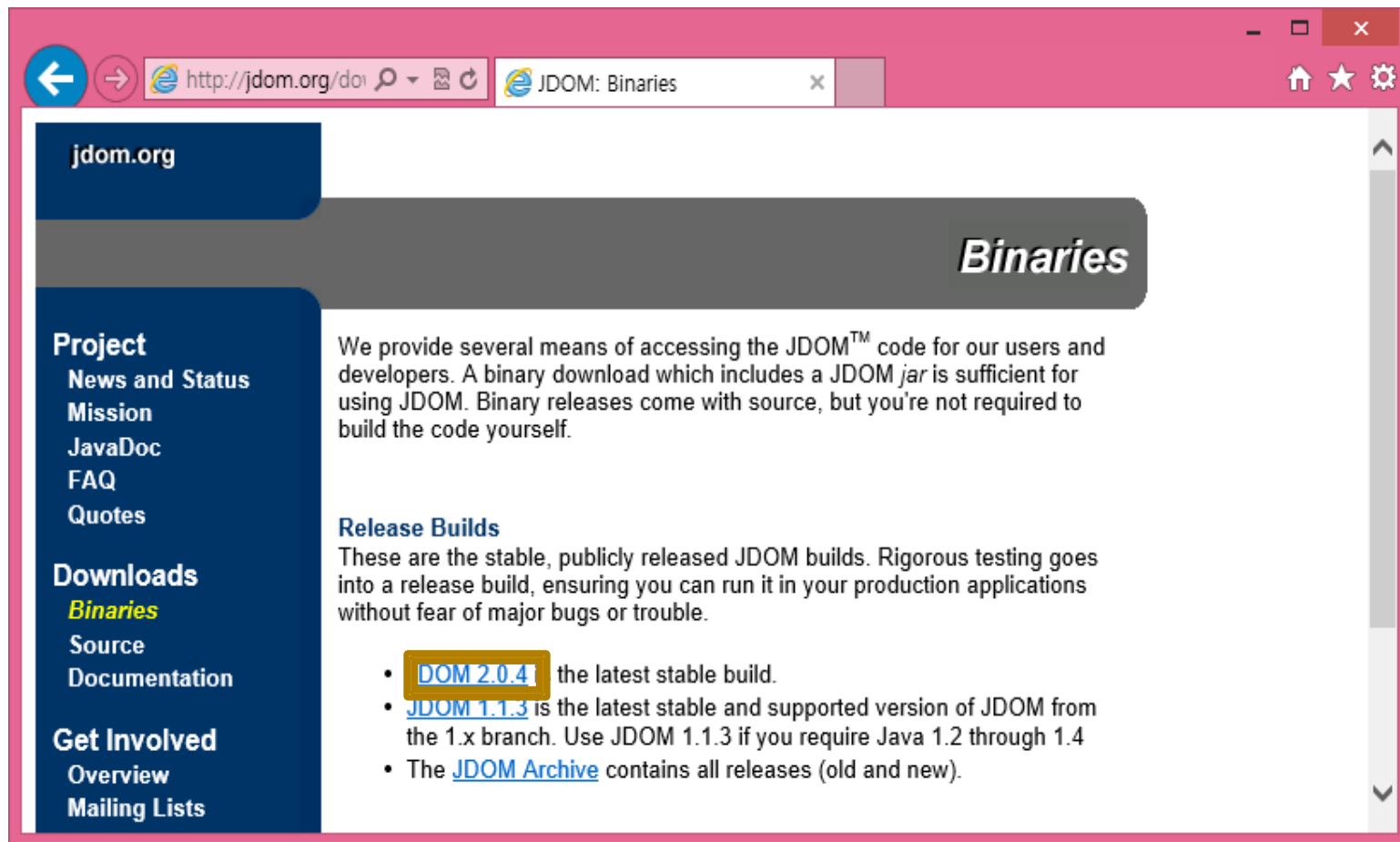
ojdbc14.jar

commons-collections, dbcp, pool

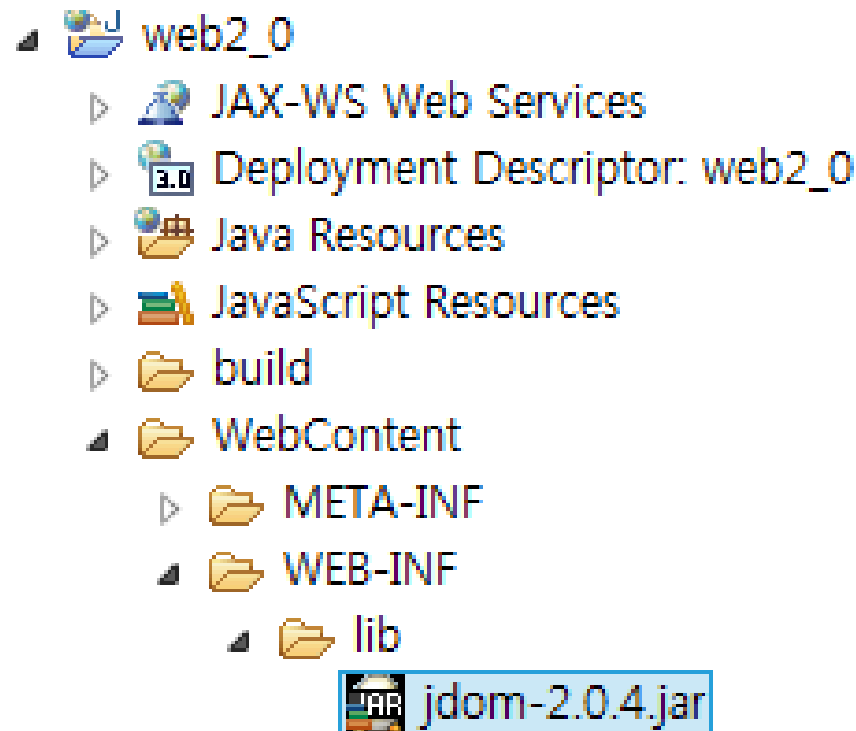
JDOM 라이브러리 다운

downloads 및의 Binaries

- <http://www.jdom.org/>



- 압축 풀어 나온 jar파일 WEB-INF/lib에 추가



Member.html

```
<html>
  <head>
  </head>
  <body>
    <h2>직원정보</h2>
    <form action="MemberInfoXml1">
      업무 : <input type="text" name="job"><p>
      <input type="submit" value="완료">
    </form>
  </body>
</html>
```

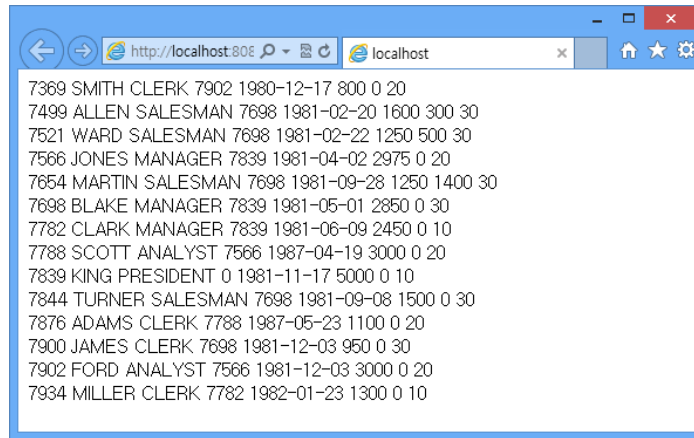
■ MemberInfoXml1.java

@WebServlet("/MemberInfoXml1")

```
public class MemberInfoXml1 extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        String job = request.getParameter("job");
        PrintWriter out = response.getWriter();
        Context init = null;          Connection conn = null;
        PreparedStatement pstmt = null; ResultSet rs = null;
        try { init = new InitialContext();
            DataSource ds = (DataSource) init.lookup("java:comp/env/jdbc/OracleDB");
            conn = ds.getConnection();      String sql = "select * from emp";
            if (job != null && job != "") { sql += " where job = ?";
                pstmt = conn.prepareStatement(sql); pstmt.setString(1, job);
            } else pstmt = conn.prepareStatement(sql);
            rs = pstmt.executeQuery();
            while(rs.next()) {
                String result = rs.getInt(1) + " "; result += rs.getString(2) + " ";
                result += rs.getString(3) + " ";   result += rs.getString(4) + " ";
                result += rs.getDate(5) + " ";    result += rs.getInt(6) + " ";
                result += rs.getInt(7) + " ";      result += rs.getInt(8) + " ";
                out.println(result);
            }
        }
    }
}
```

```
} catch (Exception e) {
    System.out.println(e.getMessage());
} finally {
    if (rs != null) { try {rs.close(); } catch(Exception e) {} }
    if (pstmt != null) { try {pstmt.close(); } catch(Exception e) {} }
    if (conn != null) { try {conn.close(); } catch(Exception e) {} }
    if (init != null) { try {init.close(); } catch(Exception e) {} }
}
}
protected void doPost(HttpServletRequest request,
    HttpServletResponse response) throws ServletException, IOException
{
}
}
```

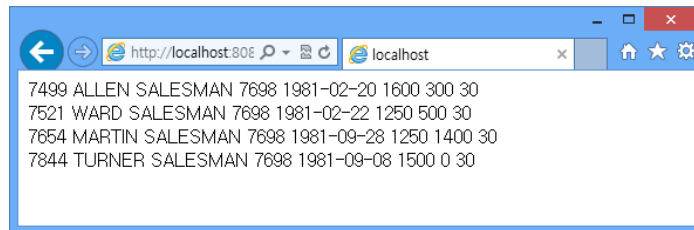
- <http://localhost:8181/ch18/MemberInfoXml1>



A screenshot of a web browser window displaying a list of employee records. The browser's address bar shows the URL <http://localhost:8080>. The page content lists 15 employees with their IDs, names, titles, hire dates, salaries, and commissions.

7369	SMITH	CLERK	7902	1980-12-17	800	0	20
7499	ALLEN	SALESMAN	7698	1981-02-20	1600	300	30
7521	WARD	SALESMAN	7698	1981-02-22	1250	500	30
7566	JONES	MANAGER	7839	1981-04-02	2975	0	20
7654	MARTIN	SALESMAN	7698	1981-09-28	1250	1400	30
7698	BLAKE	MANAGER	7839	1981-05-01	2850	0	30
7782	CLARK	MANAGER	7839	1981-06-09	2450	0	10
7788	SCOTT	ANALYST	7566	1987-04-19	3000	0	20
7839	KING	PRESIDENT	0	1981-11-17	5000	0	10
7844	TURNER	SALESMAN	7698	1981-09-08	1500	0	30
7876	ADAMS	CLERK	7788	1987-05-23	1100	0	20
7900	JAMES	CLERK	7698	1981-12-03	950	0	30
7902	FORD	ANALYST	7566	1981-12-03	3000	0	20
7934	MILLER	CLERK	7782	1982-01-23	1300	0	10

- http://localhost:8181/web2_o/MemberInfoXml1?job=SALESMAN



A screenshot of a web browser window displaying a filtered list of employee records for the job title 'SALESMAN'. The browser's address bar shows the URL <http://localhost:8080>. The page content lists 4 salesmen with their IDs, names, hire dates, salaries, and commissions.

7499	ALLEN	SALESMAN	7698	1981-02-20	1600	300	30
7521	WARD	SALESMAN	7698	1981-02-22	1250	500	30
7654	MARTIN	SALESMAN	7698	1981-09-28	1250	1400	30
7844	TURNER	SALESMAN	7698	1981-09-08	1500	0	30

자바빈 작성 Employee.java

```
import java.util.Date;
public class Employee {
    private int empno; private String ename; private String job;
    private int mgr; private Date hiredate; private int sal;
    private int comm; private int 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 int getSal() {return sal;}
    public void setSal(int sal) {this.sal = sal;}
    public int getComm() {return comm;}
    public void setComm(int comm) {this.comm = comm;}
    public int getDeptno() {return deptno;}
    public void setDeptno(int deptno) {this.deptno = deptno;}
}
```

■ MemberInfoXml.java 수정

```
ArrayList<Employee> employees = new ArrayList<Employee>();  
while(rs.next()) {  
    Employee e = new Employee();  
    e.setEmpno(rs.getInt(1));  
    e.setName(rs.getString(2));  
    e.setJob(rs.getString(3));  
    e.setMgr(rs.getInt(4));  
    e.setHiredDate(rs.getDate(5));  
    e.setSal(rs.getInt(6));  
    e.setComm(rs.getInt(7));  
    e.setDeptno(rs.getInt(8));  
    employees.add(e);  
}
```

라이브러리 사용법

- DOM객체 생성

```
Element 변수명 = new Element("elementName"); // <elementName></elementName>생성
```

- 속성 설정

```
변수명.setAttribute("속성명", "값"); // <elementName 속성명="값"></elementName>
```

- 텍스트 입력

```
변수명.setText("글자");//<elementName>글자</elementName>
```

- 자식 객체 추가

```
변수명.addContent(Element객체);//<elementName><elementName2></elementName2></elementName>
```

- 출력

```
Document 변수명d = new Document(Element객체);  
XMLOutputter 변수명x = new XMLOutputter();  
Format 변수명f = Format.getPrettyFormat();  
변수명f.setEncoding("인코딩");  
변수명x.setFormat(변수명f);  
변수명x.output(변수명d, Writer 객체);
```

■ MemberInfoXml.java에 추가(while문 뒤)

```
Response.setContentType("text/xml;charset=utf-8"); // out 생성전에
Element root = new Element("employeeData");
for (Employee e:employees) {
    Element employee = new Element("employee");

    Element empno = new Element("empno");
    empno.setText(e.getEmpno() + "");
    employee.addContent(empno);

    Element ename = new Element("ename");
    ename.setText(e.getEname() + "");
    employee.addContent(ename);

    Element _job = new Element("job");
    _job.setText(e.getJob() + "");
    employee.addContent(_job);

    Element mgr = new Element("mgr");
    mgr.setText(e.getMgr() + "");
    employee.addContent(mgr);
}
```

```
Element hiredate = new Element("hiredate");
```

```
hiredate.setText(e.getHiredate() + "");
```

```
employee.addContent(hiredate);
```

```
Element sal = new Element("sal");
```

```
sal.setText(e.getSal() + "");
```

```
employee.addContent(sal);
```

```
Element comm = new Element("comm");
```

```
comm.setText(e.getComm() + "");
```

```
employee.addContent(comm);
```

```
Element deptno = new Element("deptno");
```

```
deptno.setText(e.getDeptno() + "");
```

```
employee.addContent(deptno);
```

```
root.addContent(employee);
```

```
}
```

```
Document d = new Document(root);
```

```
XMLOutputter xml = new XMLOutputter();
```

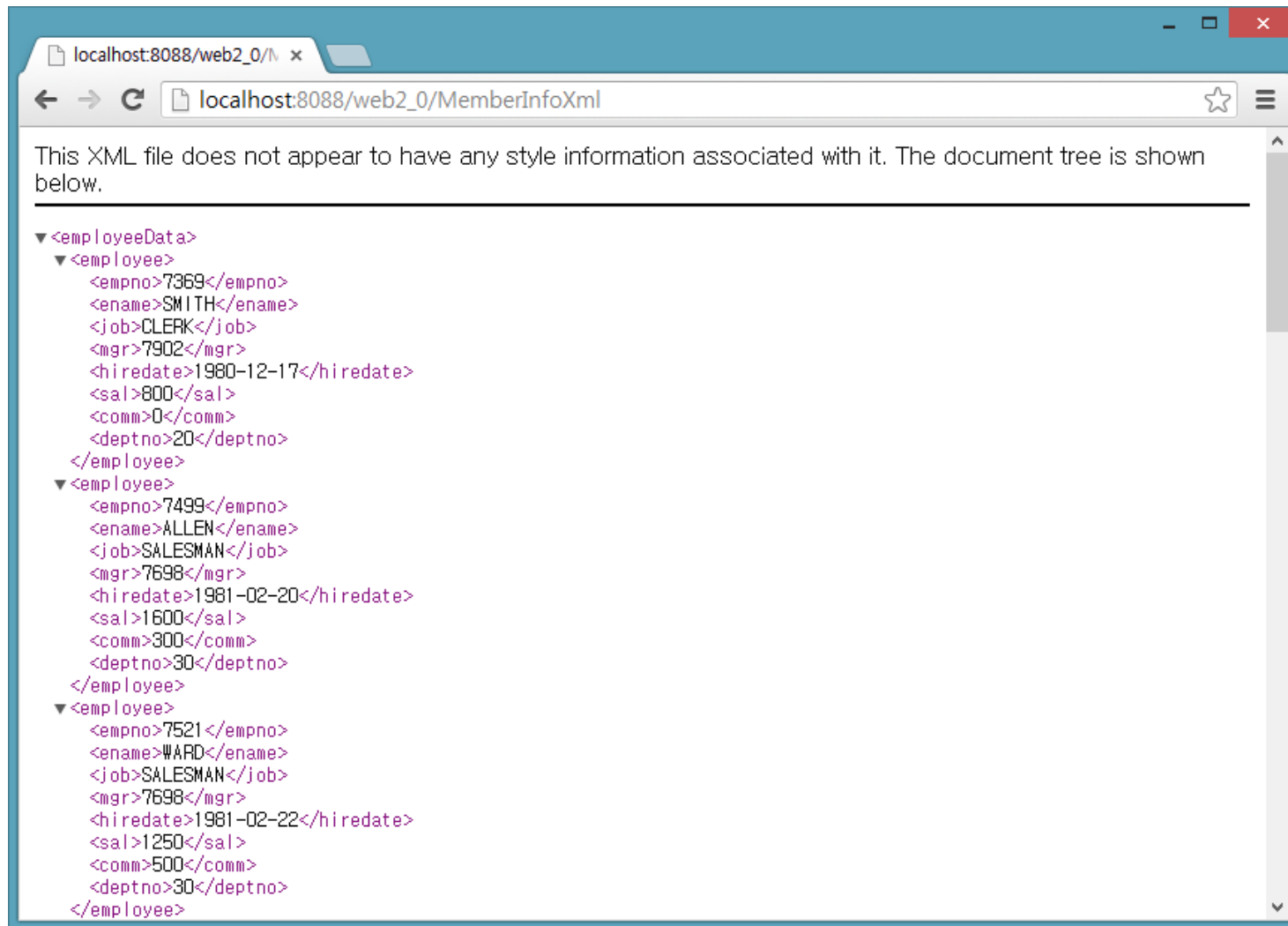
```
Format f = Format.getPrettyFormat();
```

```
// f.setEncoding("인코딩 설정");
```

```
xml.setFormat(f);
```

```
xml.output(d, out);
```

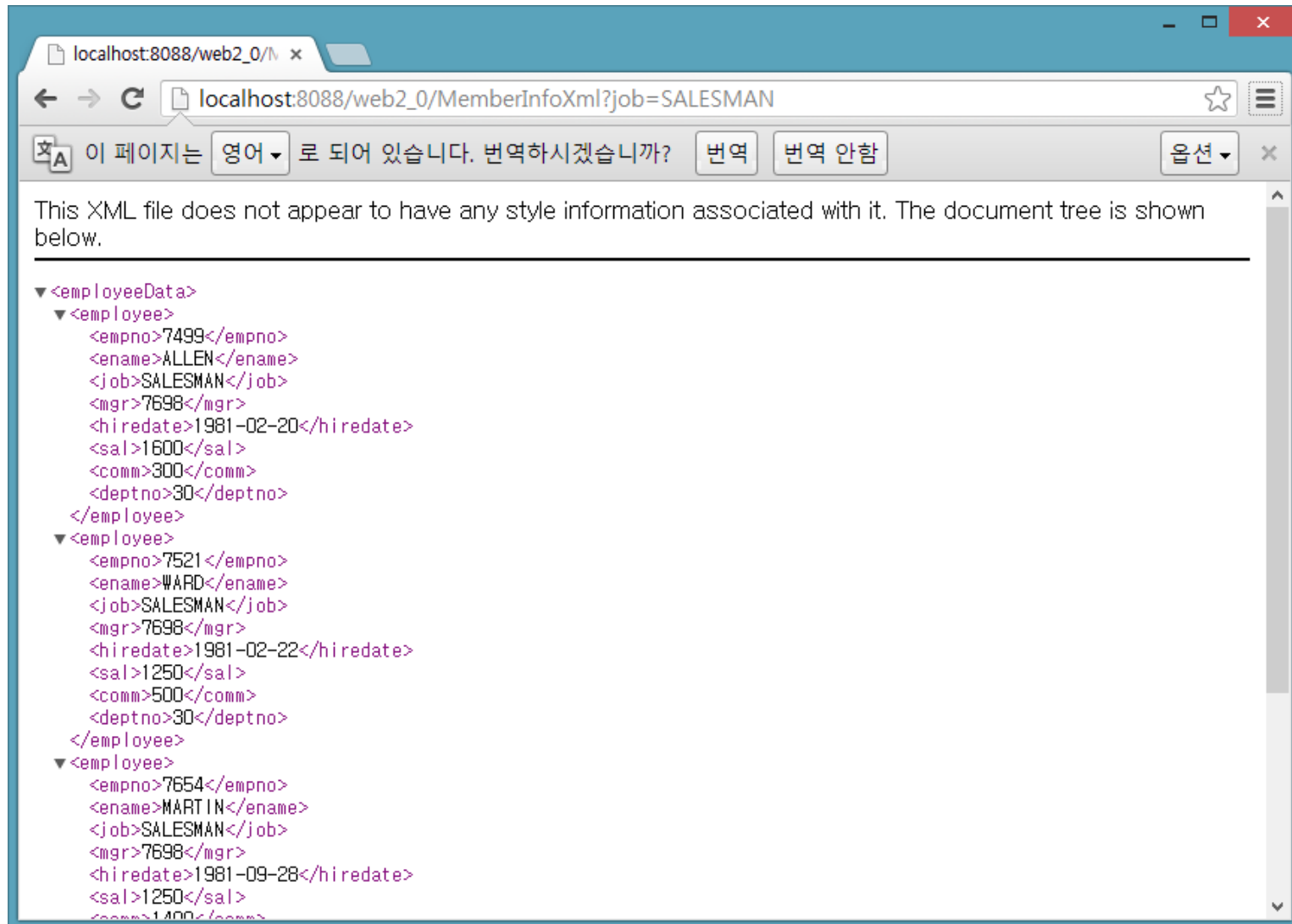
- <http://localhost:8181/ch18/MemberInfoXml>



The screenshot shows a web browser window with the address bar set to `localhost:8088/web2_0/MemberInfoXml`. The page content displays a message: "This XML file does not appear to have any style information associated with it. The document tree is shown below." Below this message, the XML document tree is rendered in a collapsed, tree-like structure. The root element is `<employeeData>`, which contains three `<employee>` elements. Each `<employee>` element contains a set of attributes: `empno`, `ename`, `job`, `mgr`, `hiredate`, `sal`, `comm`, and `deptno`.

```
<employeeData>
  <employee>
    <empno>7369</empno>
    <ename>SMITH</ename>
    <job>CLERK</job>
    <mgr>7902</mgr>
    <hiredate>1980-12-17</hiredate>
    <sal>800</sal>
    <comm>0</comm>
    <deptno>20</deptno>
  </employee>
  <employee>
    <empno>7499</empno>
    <ename>ALLEN</ename>
    <job>SALESMAN</job>
    <mgr>7698</mgr>
    <hiredate>1981-02-20</hiredate>
    <sal>1600</sal>
    <comm>300</comm>
    <deptno>30</deptno>
  </employee>
  <employee>
    <empno>7521</empno>
    <ename>WARD</ename>
    <job>SALESMAN</job>
    <mgr>7698</mgr>
    <hiredate>1981-02-22</hiredate>
    <sal>1250</sal>
    <comm>500</comm>
    <deptno>30</deptno>
  </employee>
</employeeData>
```

- <http://localhost:8181/ch18/MemberInfoXml?job=SALESMAN>



JSON



JSON-lib 라이브러리 다운

- <http://json-lib.sourceforge.net/>

JSON-lib

Last Published: 12/14/2010 Json.org | Maven 2

Json-lib

- Download
- Feature List
- Getting Started
- Advanced features
- Snippets
- Groovy integration
- Build instructions
- Who is using it?
- FAQ
- Changes
- Javadoc (jdk13)
- Javadoc (jdk15)

Welcome to Json-lib

JSON-lib is a java library for transforming beans, maps, collections, java arrays and XML to JSON and back again to beans and DynaBeans.
It is based on the work by Douglas Crockford in <http://www.json.org/java>.

The following tables summarizes the types conversion between java and javascript:



JSON		Java
string	<=>	java.lang.String, java.lang.Character, char
number	<=>	java.lang.Number, byte, short, int, long, float, double
true false	<=>	java.lang.Boolean, boolean
null	<=>	null
function	<=>	net.sf.json.JSONFunction
array	<=>	net.sf.json.JSONArray (object, string, number, boolean, function)
object	<=>	net.sf.json.JSONObject


The function type from javascript is not part of the JSON format "officially" (please refer to

← → http://sourceforge.net/projects/Json-lib - Browse Files at S... x

sourceforge Search Browse Blog Help Jobs

SOLUTION CENTERS Smarter Commerce Go Parallel Smarter IT Newsletters

 화면을 켜는 순간,
현실과의 경계가 사라집니다  최고 11만원
할인 행사중! New 델 S 시리즈 모니터
지금 만나보세요 >


 Sign up for
the **Slashdot**
Daily News!

Json-lib
aalmiray

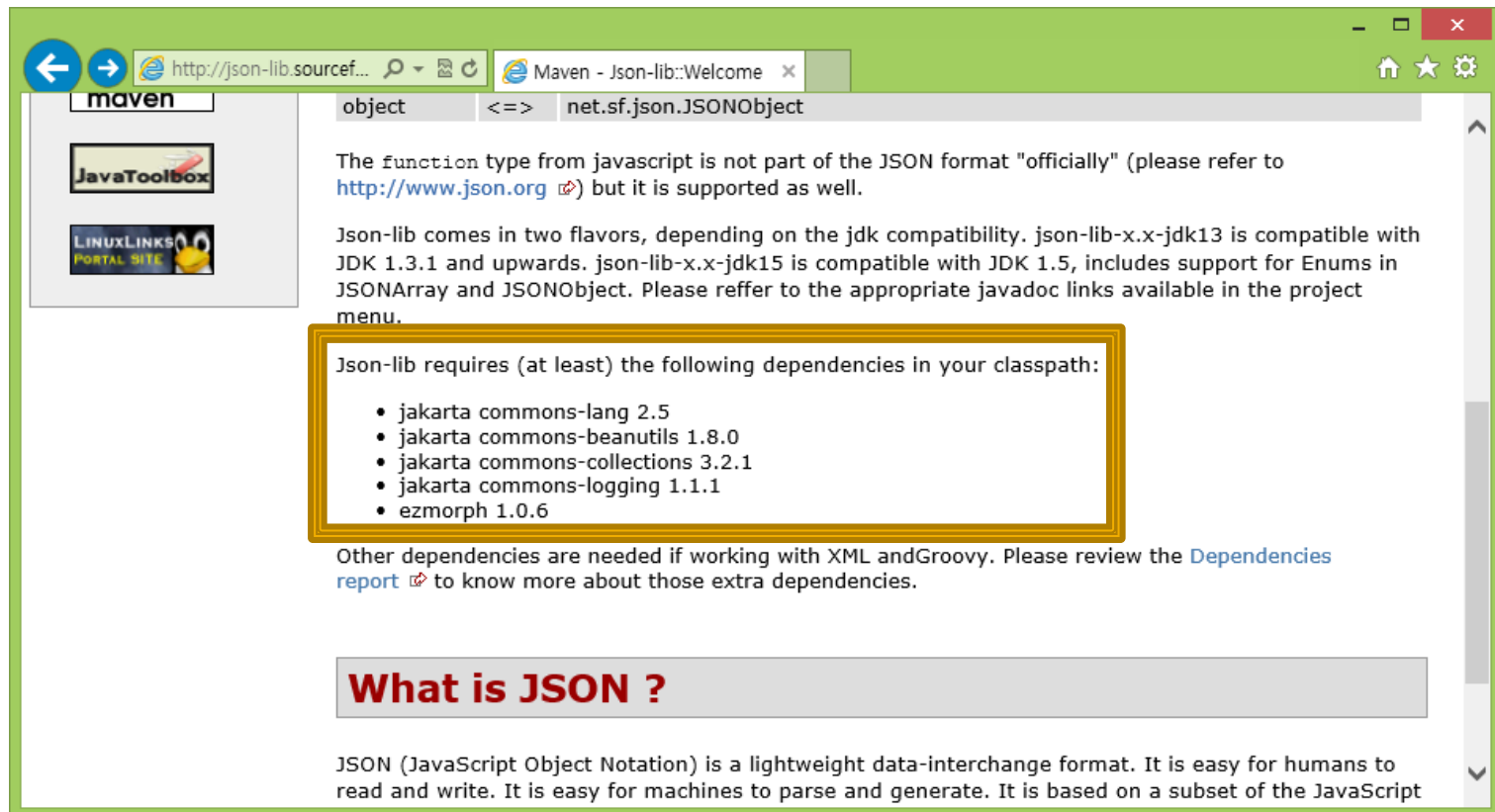
Summary Files Reviews Support Develop Tracker Mailing Lists Forums

Looking for the latest version? [Download json-lib-2.4-jdk15.jar \(159.1 kB\)](#)

Home

Name ▾	Modified ▾	Size ▾	Downloads ▾
			

■ 전부 다 다운로드



The screenshot shows a web browser window with the URL `http://json-lib.sourceforge.net/`. The page title is "Maven - Json-lib::Welcome". The breadcrumb trail is "object <=> net.sf.json.JSONObject".

The main content area contains the following text:

The function type from javascript is not part of the JSON format "officially" (please refer to <http://www.json.org>) but it is supported as well.

Json-lib comes in two flavors, depending on the jdk compatibility. json-lib-x.x-jdk13 is compatible with JDK 1.3.1 and upwards. json-lib-x.x-jdk15 is compatible with JDK 1.5, includes support for Enums in JSONArray and JSONObject. Please refer to the appropriate javadoc links available in the project menu.

Json-lib requires (at least) the following dependencies in your classpath:

- jakarta commons-lang 2.5
- jakarta commons-beanutils 1.8.0
- jakarta commons-collections 3.2.1
- jakarta commons-logging 1.1.1
- ezmorph 1.0.6

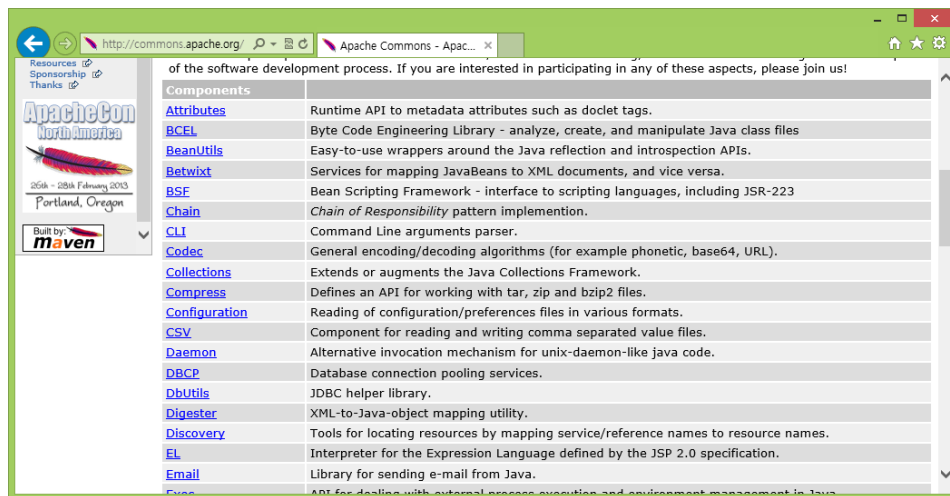
Other dependencies are needed if working with XML and Groovy. Please review the [Dependencies report](#) to know more about those extra dependencies.

What is JSON ?

JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. It is based on a subset of the JavaScript

jakarta commons-???

- <http://commons.apache.org/>



- commons-lang-은 2.x버전 사용

Commons Lang 2.6 (Requires Java 1.2 or later)

Binaries

[commons-lang-2.6-bin.tar.gz](#)

[md5](#)

[pgp](#)

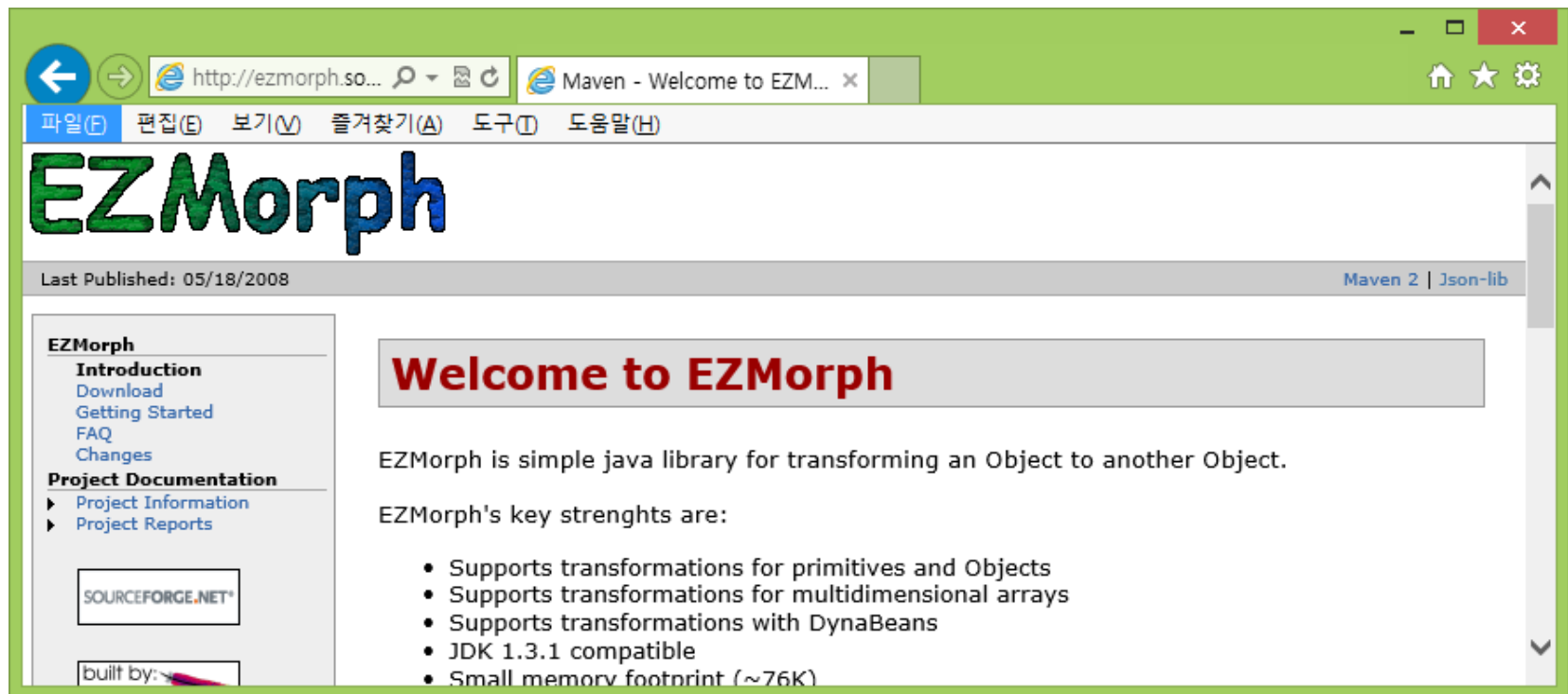
[commons-lang-2.6-bin.zip](#)

[md5](#)

[pgp](#)

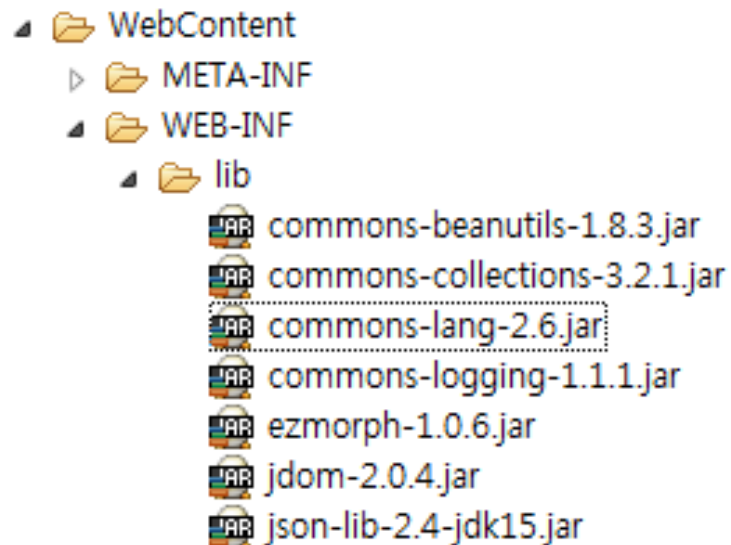
ezmorph

- <http://ezmorph.sourceforge.net/>



.jar추가

- 다 압축 풀어서 jar파일을 WEB-INF/lib에 추가



라이브리리 사용법

- 자바빈의 멤버 변수는 모두 String타입으로
- 자바빈 객체를 만들어 List에 넣음
- ```
JSONArray 변수명 = JSONArray.fromObject(List객체);
out.println(변수명);
```

# 자바빈 작성

- Employee.java의 멤버 변수들만 String타입으로 바꾼 Employee2.java 작성

```
public class Employee2 {
 private String empno;
 private String ename;
 private String job;
 private String mgr;
 private String hiredate;
 private String sal;
 private String comm;
 private String deptno;

 public String getEmpno() {return empno;}
 public void setEmpno(String empno) {this.empno = empno;}

 ...
}
```



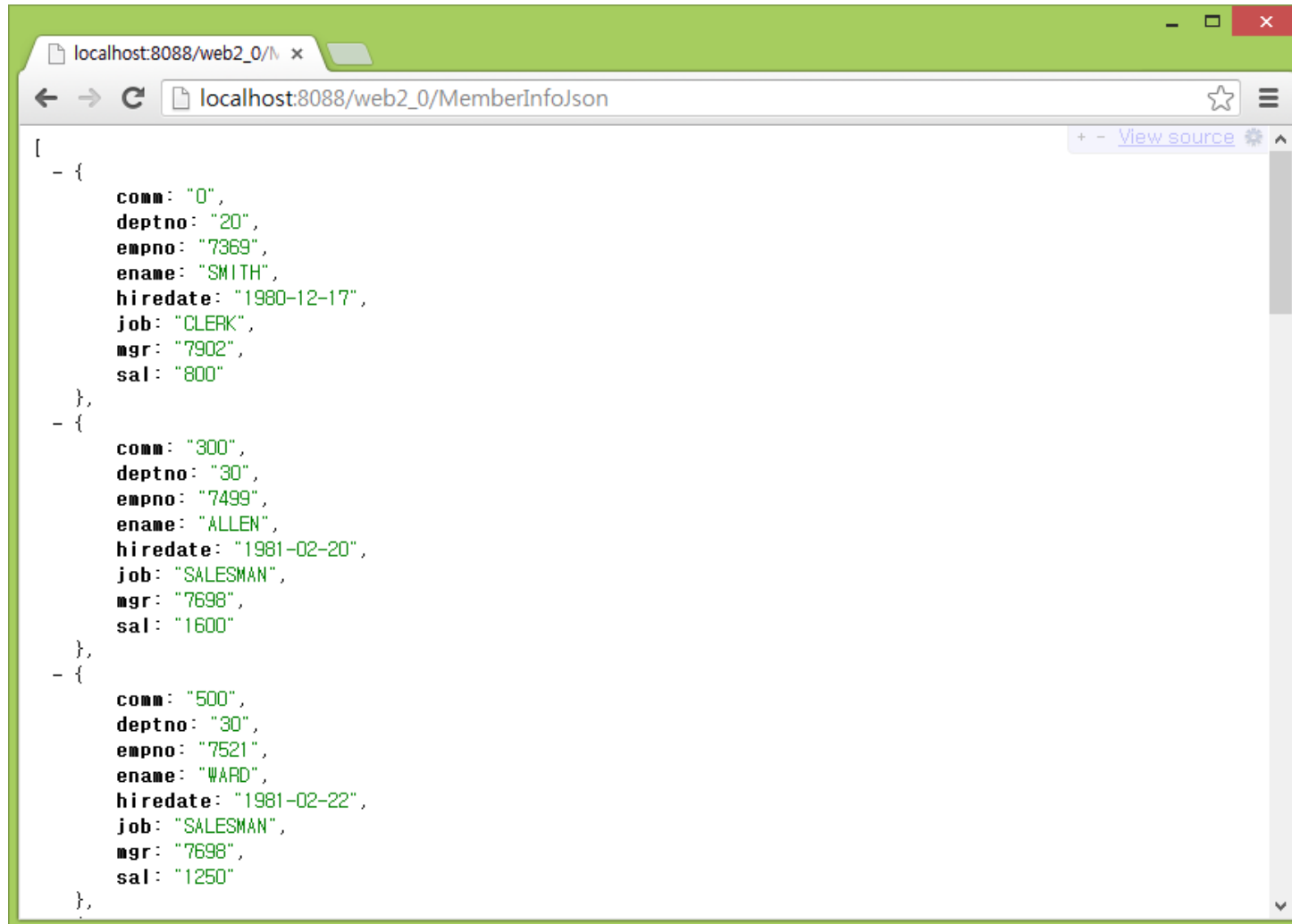
# 서블릿 작성

## ■ MemberInfoJson.java 작성

```
ArrayList<Employee2> employees = new ArrayList<Employee2>();
while(rs.next()) {
 Employee2 e = new Employee2();
 e.setEmpno(rs.getInt(1) + "");
 e.setEname(rs.getString(2));
 e.setJob(rs.getString(3));
 e.setMgr(rs.getInt(4) + "");
 e.setHiredate(rs.getDate(5) + "");
 e.setSal(rs.getInt(6) + "");
 e.setComm(rs.getInt(7) + "");
 e.setDeptno(rs.getInt(8) + "");
 employees.add(e);
}
JSONArray json = JSONArray.fromObject(employees);
out.println(json);
```

# 결과

- <http://localhost:8181/ch18/MemberInfoJson>

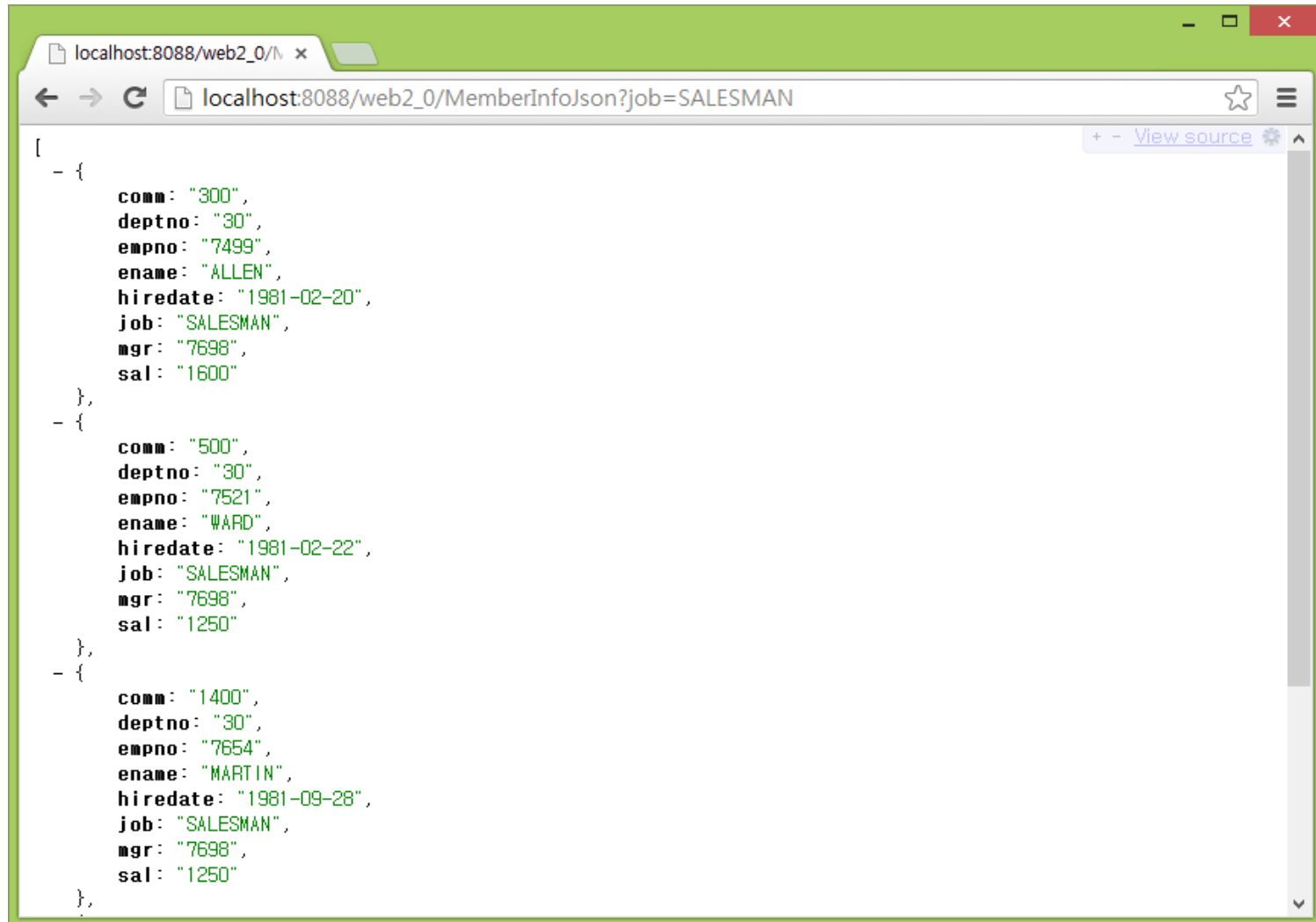


A screenshot of a web browser window with a green title bar. The address bar shows the URL `localhost:8088/web2_0/MemberInfoJson`. The main content area displays a JSON array of three employee objects. The first object is for SMITH, the second for ALLEN, and the third for WARD. Each object contains fields for commission, department number, employee number, name, hire date, job title, manager, and salary.

```
[
 - {
 comm: "0",
 deptno: "20",
 empno: "7369",
 ename: "SMITH",
 hiredate: "1980-12-17",
 job: "CLERK",
 mgr: "7902",
 sal: "800"
 },
 - {
 comm: "300",
 deptno: "30",
 empno: "7499",
 ename: "ALLEN",
 hiredate: "1981-02-20",
 job: "SALESMAN",
 mgr: "7698",
 sal: "1600"
 },
 - {
 comm: "500",
 deptno: "30",
 empno: "7521",
 ename: "WARD",
 hiredate: "1981-02-22",
 job: "SALESMAN",
 mgr: "7698",
 sal: "1250"
 },
]
```

# 결과

- <http://localhost:8181/ch18/MemberInfoJson?job=SALESMAN>



A screenshot of a web browser window with a green title bar. The address bar shows the URL `localhost:8088/web2_0/MemberInfoJson?job=SALESMAN`. The page content displays a JSON array of three employee records. The browser interface includes a back button, a refresh button, and a 'View source' link. The JSON data is as follows:

```
[
 - {
 comm: "300",
 deptno: "30",
 empno: "7499",
 ename: "ALLEN",
 hiredate: "1981-02-20",
 job: "SALESMAN",
 mgr: "7698",
 sal: "1600"
 },
 - {
 comm: "500",
 deptno: "30",
 empno: "7521",
 ename: "WARD",
 hiredate: "1981-02-22",
 job: "SALESMAN",
 mgr: "7698",
 sal: "1250"
 },
 - {
 comm: "1400",
 deptno: "30",
 empno: "7654",
 ename: "MARTIN",
 hiredate: "1981-09-28",
 job: "SALESMAN",
 mgr: "7698",
 sal: "1250"
 },
]
```