

17장 서블릿 기초

17.1 서블릿 기초

- 서블릿은 JSP 표준이 나오기 전에 만들어진 표준으로 자바로 웹 어플리케이션을 개발할 수 있도록 하기 위해 만들어졌다. 일반적인 서블릿의 개발 과정은 다음과 같다.
 1. 서블릿 규약에 따라 자바 코드를 작성한다.
 2. 자바 코드를 컴파일해서 클래스 파일을 생성한다.
 3. 클래스 파일을 /WEB-INF/classes 폴더에 패키지에 알맞게 위치시킨다.
 4. web.xml 파일에 서블릿 클래스를 설정한다.
 5. 톰캣 등의 컨테이너를 실행한다.
 6. 웹 브라우저에서 확인한다.

17.2 예제 프로젝트 생성

- 이클립스에서 File > Import > Existing Projects into Workspace 메뉴를 이용하여 프로젝트 (chap17)를 임포트한다.

17.2.1 서블릿 구현

[chap17/src/example/NowServlet.java]

```
01 package example;
02
03 import java.io.IOException;
04 import java.io.PrintWriter;
05 import java.util.Date;
06
07 import javax.servlet.ServletException;
08 import javax.servlet.http.HttpServlet;
09 import javax.servlet.http.HttpServletRequest;
10 import javax.servlet.http.HttpServletResponse;
11
12 public class NowServlet extends HttpServlet {
13
14     @Override
15     protected void doGet(HttpServletRequest request,
16         HttpServletResponse response) throws ServletException, IOException {
17         response.setContentType("text/html; charset=utf-8");
18
19         PrintWriter out = response.getWriter();
20         out.println("<html>");
21         out.println("<head><title>현재시간</title></head>");
22         out.println("<body>");
23         out.println("현재 시간은");
24         out.println(new Date());
25         out.println("입니다.");
26         out.println("</body></html>");
27     }
28
29 }
```

17.2.2 web.xml로 매핑하기

- 서블릿 클래스를 생성했다면 그다음으로 할 작업은 WEB-INF 폴더의 web.xml 파일에 서블릿 클래스를 등록하는 것이다.

[chap17/WebContent/WEB-INF/web.xml]

```
01 <?xml version="1.0" encoding="UTF-8"?>
02 <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
03         xmlns="http://xmlns.jcp.org/xml/ns/javaee"
04         xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
05                             http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
06         version="3.1">
07
08     <servlet>
09         <servlet-name>now</servlet-name>
10         <servlet-class>example.NowServlet</servlet-class>
11     </servlet>
12
13     <servlet-mapping>
14         <servlet-name>now</servlet-name>
15         <url-pattern>/now</url-pattern>
16     </servlet-mapping>
17
18     <servlet>
19         <servlet-name>DBCPInit2</servlet-name>
20         <servlet-class>jdbc.DBCPInit2</servlet-class>
21         <init-param>
22             <param-name>jdbcdriver</param-name>
23             <param-value>com.mysql.jdbc.Driver</param-value>
24         </init-param>
25         <init-param>
26             <param-name>jdbcUrl</param-name>
27             <param-value>
28                 jdbc:mysql://localhost:3306/chap14?characterEncoding=utf8
29             </param-value>
30         </init-param>
31         <init-param>
32             <param-name>dbUser</param-name>
33             <param-value>jspexam</param-value>
34         </init-param>
35         <init-param>
36             <param-name>dbPass</param-name>
37             <param-value>jspw</param-value>
38         </init-param>
39         <init-param>
40             <param-name>poolName</param-name>
41             <param-value>chap14</param-value>
42         </init-param>
43         <load-on-startup>1</load-on-startup>
44     </servlet>
45
46 </web-app>
```

17.2.3 애노테이션으로 매핑하기

- 서블릿 3.0 버전부터는 @WebServlet 애노테이션을 사용하면 web.xml 파일에 따로 등록하지 않아도 서블릿으로 등록된다.

[chap17/src/example/HelloServlet.java]

```

01 package example;
02
03 import java.io.IOException;
04 import java.io.PrintWriter;
05
06 import javax.servlet.ServletException;
07 import javax.servlet.annotation.WebServlet;
08 import javax.servlet.http.HttpServlet;
09 import javax.servlet.http.HttpServletRequest;
10 import javax.servlet.http.HttpServletResponse;
11
12 @WebServlet(urlPatterns = "/hello")
13 public class HelloServlet extends HttpServlet {
14
15     @Override
16     protected void doGet(HttpServletRequest request,
17                          HttpServletResponse response) throws ServletException, IOException {
18         request.setCharacterEncoding("utf-8");
19         response.setContentType("text/html; charset=utf-8");
20
21         PrintWriter out = response.getWriter();
22         out.println("<html>");
23         out.println("<head><title>인사</title></head>");
24         out.println("<body>");
25         out.println("안녕하세요, ");
26         out.println(request.getParameter("name"));
27         out.println("님");
28         out.println("</body></html>");
29     }
30
31 }

```

17.2.4 HTTP 각 방식별 구현 메서드

- GET 방식은 doGet() 메서드를 이용해서 처리하고, POST 방식의 경우 doPost() 메서드를 이용해서 처리하도록 정의하고 있다.

```

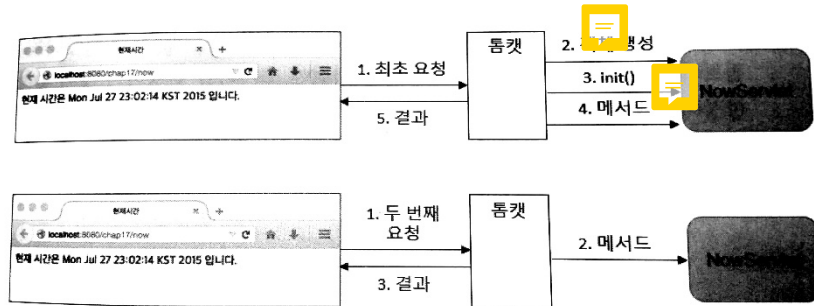
@Override
protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException,
IOException {
    ... // GET 방식에 대한 처리
}

@Override
protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException,
IOException {
    ... // Post 방식에 대한 처리
}

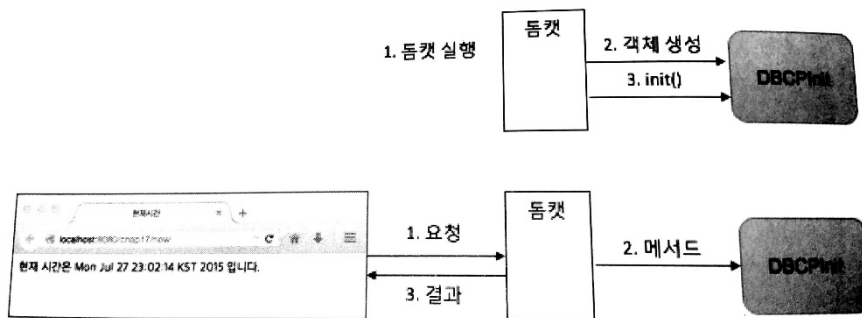
```

17.2.5 서블릿 로딩과 초기화

- 최초 요청이 발생하면 서블릿 객체를 생성



- `<load-on-startup>` 태그를 사용하면 서블릿을 미리 초기화할 수 있다. 태그의 값은 로딩 순서를 의미한다.



- `@WebServlet` 태그를 사용하는 경우에는 `loadOnStartup` 속성을 이용해서 로딩 값을 지정한다.

```
// 서블릿
public class DBCPInit2 extends HttpServlet {
    @Override
    public void init() throws ServletException {
        loadJDBCdriver();
        initConnectionPool();
    }
    ...
}

// web.xml
<servlet>
    <servlet-name>DBCPInit2</servlet-name>
    <servlet-class>jdbc.DBCPInit2</servlet-class>
    <load-on-startup>1</load-on-startup>
</servlet>

// @WebServlet 태그
@WebServlet(urlPatterns="/hello", loadOnStartup=1)
public class InitServlet extends HttpServlet {
    ...
}
```

17.2.6 초기화 파라미터

- 서블릿은 `web.xml`의 `<init-param>` 태그를 이용해서 서블릿을 초기화할 때 필요한 값을 전달하는 방법을 제공하고 있다.

(1) DBCPInit2.java

[chap17/src/jdbc/DBCPInit2.java]

```
01 package jdbc;
02
03 import java.sql.DriverManager;
04
05 import javax.servlet.ServletException;
06 import javax.servlet.http.HttpServlet;
07
08 import org.apache.commons.dbcp2.ConnectionFactory;
09 import org.apache.commons.dbcp2.DriverManagerConnectionFactory;
10 import org.apache.commons.dbcp2.PoolableConnection;
11 import org.apache.commons.dbcp2.PoolableConnectionFactory;
12 import org.apache.commons.dbcp2.PoolingDriver;
13 import org.apache.commons.pool2.impl.GenericObjectPool;
14 import org.apache.commons.pool2.impl.GenericObjectPoolConfig;
15
16 public class DBCPInit2 extends HttpServlet {
17
18     @Override
19     public void init() throws ServletException {
20         loadJDBCDriver();
21         initConnectionPool();
22     }
23
24
25     private void loadJDBCDriver() {
26         String driverClass = getInitParameter("jdbcdriver");
27         try {
28             Class.forName(driverClass);
29         } catch (ClassNotFoundException ex) {
30             throw new RuntimeException("fail to load JDBC Driver", ex);
31         }
32     }
33
34     private void initConnectionPool() {
35         try {
36             String jdbcUrl = getInitParameter("jdbcUrl");
37             String username = getInitParameter("dbUser");
38             String pw = getInitParameter("dbPass");
39
40             ConnectionFactory connFactory =
41                 new DriverManagerConnectionFactory(jdbcUrl, username,
42 pw);
43
44             PoolableConnectionFactory poolableConnFactory =
45                 new PoolableConnectionFactory(connFactory, null);
46             poolableConnFactory.setValidationQuery("select 1");
47
48             GenericObjectPoolConfig poolConfig = new GenericObjectPoolConfig();
49             poolConfig.setTimeBetweenEvictionRunsMillis(1000L * 60L * 5L);
50             poolConfig.setTestWhileIdle(true);
51             poolConfig.setMinIdle(4);
52             poolConfig.setMaxTotal(50);
53
54             GenericObjectPool<PoolableConnection> connectionPool =
55                 new GenericObjectPool<>(poolableConnFactory,
56 poolConfig);
57             poolableConnFactory.setPool(connectionPool);
58
59             Class.forName("org.apache.commons.dbcp2.PoolingDriver");
60             PoolingDriver driver =
61                 (PoolingDriver)
```

```

62 DriverManager.getDriver("jdbc:apache:commons:dbcp:");
63         String poolName = getInitParameter("poolName");
64         driver.registerPool(poolName, connectionPool);
65     } catch (Exception e) {
66         throw new RuntimeException(e);
67     }
68 }
69 }

```

(2) web.xml

[chap17/WebContent/WEB-INF/web.xml]

```

01 <?xml version="1.0" encoding="UTF-8"?>
02 <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
03     xmlns="http://xmlns.jcp.org/xml/ns/javaee"
04     xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
05         http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
06     version="3.1">
07
08     <servlet>
09         <servlet-name>now</servlet-name>
10         <servlet-class>example.NowServlet</servlet-class>
11     </servlet>
12
13     <servlet-mapping>
14         <servlet-name>now</servlet-name>
15         <url-pattern>/now</url-pattern>
16     </servlet-mapping>
17
18     <servlet>
19         <servlet-name>DBCPInit2</servlet-name>
20         <servlet-class>jdbc.DBCPInit2</servlet-class>
21         <init-param>
22             <param-name>jdbcdriver</param-name>
23             <param-value>com.mysql.jdbc.Driver</param-value>
24         </init-param>
25         <init-param>
26             <param-name>jdbcUrl</param-name>
27             <param-value>
28                 jdbc:mysql://localhost:3306/chap14?characterEncoding=utf8
29             </param-value>
30         </init-param>
31         <init-param>
32             <param-name>dbUser</param-name>
33             <param-value>jspexam</param-value>
34         </init-param>
35         <init-param>
36             <param-name>dbPass</param-name>
37             <param-value>jspw</param-value>
38         </init-param>
39         <init-param>
40             <param-name>poolName</param-name>
41             <param-value>chap14</param-value>
42         </init-param>
43         <load-on-startup>1</load-on-startup>
44     </servlet>
45
46 </web-app>

```