CONTENTS

- 서블릿의 기초
- 서블릿 구현
- URL 매핑 규칙

■ 서블릿

- 서블릿은 JSP 표준이 나오기 전에 자바에서 웹 어플리케이션 개발을 위해 만들어진 표준.
- 서블릿 개발 과전은 JSP 비해 복잡함.
- MVC 패턴을 지원하는 프레임워크의 경우 기반 코드를 서블릿에서 개발하기 때문에 서블릿을 직접 구현하지 않더라도 서블릿 자체에 대해서 이해하는 것은 중요.

■ 서블릿 개발과정

1. 서블릿 규약에 따라 자바 코드를 작성 (HttpServlet 상속 후 메서드 구현)
2. 자바코드를 컴파일 해서 클래스파일을 생성한다.
   (이클립스 환경에서 개발할 경우 이 단계는 하지 않음.)
3. 클래스파일을 /WEB-INF/classes 디렉토리에 패키지에 알맞게 위치시킨다. (이클립스 환경에서 개발할 경우 이 단계는 하지 않음.)
4. web.xml 파일에 서블릿 클래스를 설정한다.
5. 톰캣 등의 웹컨테이너를 재생시킨다.
6. 웹 브라우저에서 확인한다.

■ 서블릿 개발과정

1. 서블릿 규약에 따라 자바 코드를 작성 (HttpServlet 상속 후 메서드 구현)
2. 자바코드를 컴파일 해서 클래스파일을 생성한다.
   (이클립스 환경에서 개발할 경우 이 단계는 하지 않음.)
3. 클래스파일을 /WEB-INF/classes 디렉토리에 패키지에 알맞게 위치시킨다. (이클립스 환경에서 개발할 경우 이 단계는 하지 않음.)
4. web.xml 파일에 서블릿 클래스를 설정한다.
5. 톰캣 등의 웹컨테이너를 재생시킨다.
6. 웹 브라우저에서 확인한다.

■ 서블릿 구현

NowServlet.java

```
package mvjsp.chap20;

import java.io.IOException;
import java.io.PrintWriter;
import java.util.Date;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class NowServlet extends HttpServlet { // HttpServlet 클래스를 상속받아 클래스 작성
// 처리하고자 하는 HTTP 방식(mothod)에 따라 알맞게 메서드를 오버라이딩 해주어야 함.
// doGet, doPost
        @Override
        protected void doGet(HttpServletRequest request, HttpServletResponse response)
                                                        throws ServletException, IOException {
// HttpServletReques 와 HttpServletResponse 파라미터는 JSP에서의 request와 response 와 같다.
                response.setContentType("text/html; charset=euc-kr");
```

■ 서블릿 구현

NowServlet.java

```
                PrintWriter out = response.getWriter();
                out.println("<html>");
                out.println("<head><title>현재시간</title></head>");
                out.println("<body>");
                out.println("현재 시간은");
                out.println(new Date());
                out.println("입니다.");
                out.println("</body></html>");
        }

}
```

■ 서블릿 구현 : web.xml 매핑

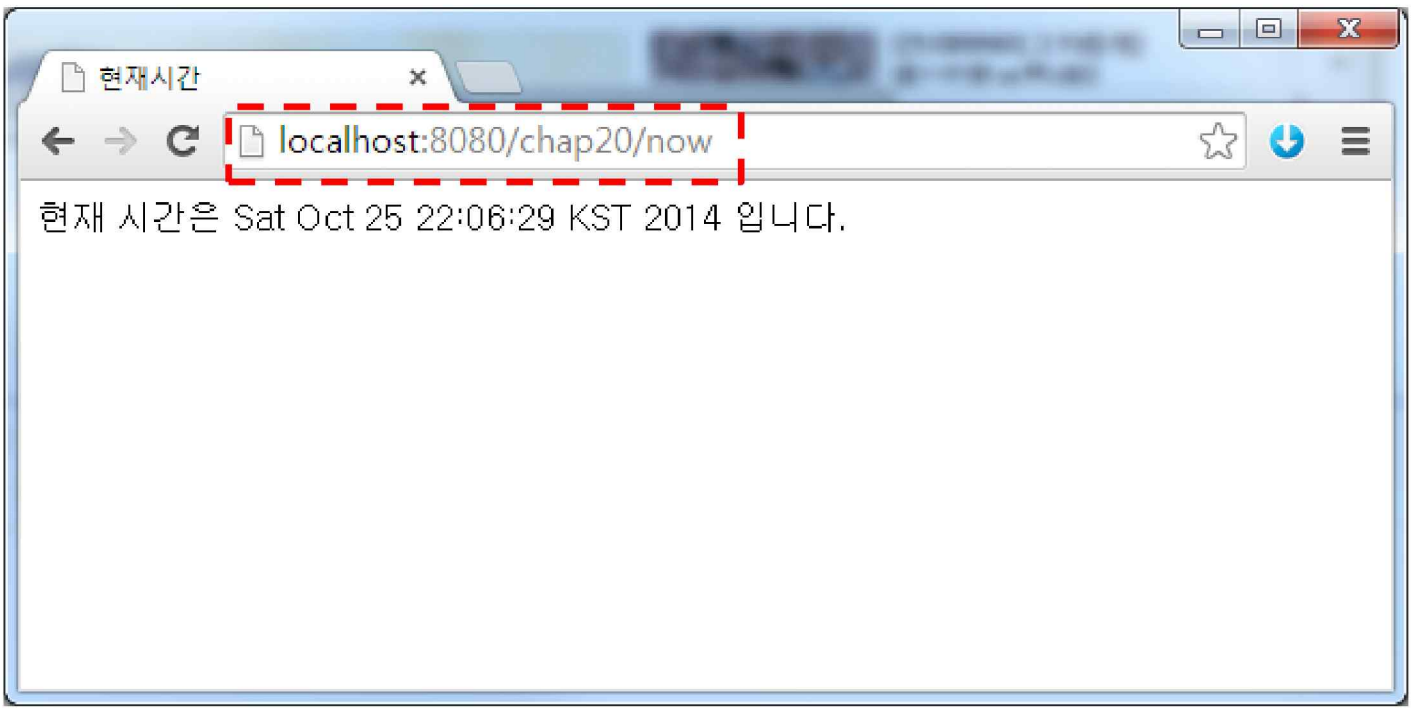
NowServlet.java

```
<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns="http://java.sun.com/xml/ns/javaee"
        xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
                http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
        id="chap20" version="3.0">
<!-- 서블릿으로 사용할 클래스 등록 -->
        <servlet>
                <servlet-name>now</servlet-name>
                <servlet-class>NowServlet</servlet-class>
        </servlet>
<!-- URL을 통해 서블릿을 호출할 수 있도록 서블릿과 URL과의 매핑 -->
        <servlet-mapping>
                <servlet-name>now</servlet-name>
                <url-pattern>/now</url-pattern>
        </servlet-mapping>

</web-app>
```

■ 서블릿 구현 : 결과


현재시간
localhost:8080/chap20/now
현재 시간은 Sat Oct 25 22:06:29 KST 2014 입니다.

■ 서블릿 구현 :이노테이션으로 매핑하기

HelloServlet.java

```
package mvjsp.chap20;

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet(urlPatterns = "/hello")
// 두개 이상의 URL 패턴을 처리 할 경우
//@WebServlet(urlPatterns = {"/hello", "/hello1"})
public class HelloServlet extends HttpServlet {

	@Override
	protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
```

■ 서블릿 구현 :이노테이션으로 매핑하기

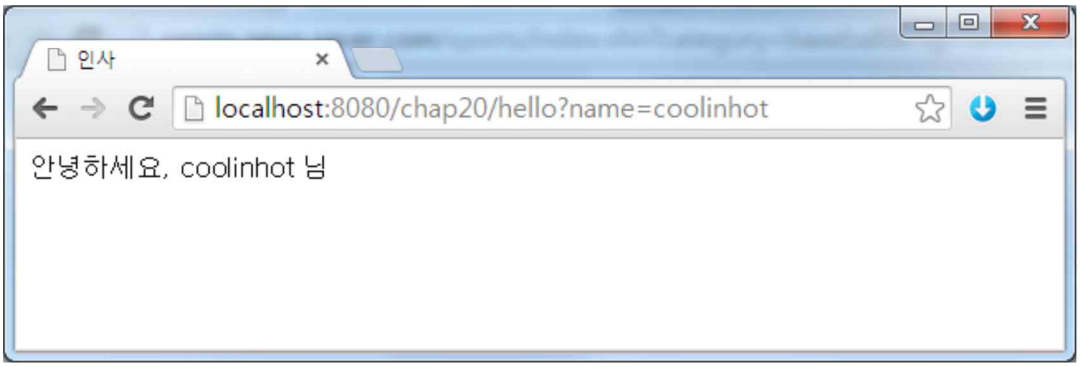
HelloServlet.java

```
                request.setCharacterEncoding("euc-kr");
                response.setContentType("text/html; charset=euc-kr");

                PrintWriter out = response.getWriter();
                out.println("<html>");
                out.println("<head><title>인사</title></head>");
                out.println("<body>");
                out.println("안녕하세요, ");
                out.println(request.getParameter("name"));
                out.println("님");
                out.println("</body></html>");
        }

}
```

■ 서블릿 구현 :이노테이션으로 매핑하기 결과



■ 서블릿 구현 : HTTP 각 방식별 구현 메서드

```
/**
 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
 */
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
// TODO Auto-generated method stub
}

/**
 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
 */
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
// TODO Auto-generated method stub
}
```

■ 서블릿 구현 : HTTP 각 방식별 구현 메서드

```
/**
 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
 */
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
// TODO Auto-generated method stub
}

/**
 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
 */
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
// TODO Auto-generated method stub
}
```

■ 서블릿 구현 : 서블릿의 초기화 파라미터, 초기화, 로딩

- 서블릿이 사용이 웹 URL을 통한 실행이 아닌
- <init-param>태그로 지정한 값은 초기화 파라미터라고 불리는 설정값을 지정할 때 사용.
  - <param-name> : 파라미터의 이름
  - <param-value> : 초기화 파라미터의 값
- 서블릿 클래스는 getInitParameter()메서드를 이용해서 초기화 파라미터 값을 얻어온다.
- init() 메서드 오버라이딩.

■ 서블릿 구현 : 서블릿의 초기화 파라미터, 초기화, 로딩

- 서블릿이 사용이 웹 URL을 통한 실행이 아닌
- <init-param>태그로 지정한 값은 초기화 파라미터라고 불리는 설정값을 지정할 때 사용.
  - <param-name> : 파라미터의 이름
  - <param-value> : 초기화 파라미터의 값
- 서블릿 클래스는 getInitParameter()메서드를 이용해서 초기화 파라미터 값을 얻어온다.
- init() 메서드 오버라이딩.
- 이노테이션에서 설정

```
@WebServlet(urlPattern = "/hello", loadOnStartup =1)

@WebServlet(urlPattern = "/hello",
   initParam = {
      @WebInitParam(name="greeting", value="Hello"),
      @WebInitParam(name="title", value="제목"),
   })
```

■ 서블릿 구현 : URL 패턴 매핑 규칙

- '/'로 시작하고 '/*'로 끝나는 url-pattern은 경로 매핑을 위해서 사용
- '*.'로 시작하는 url-pattern은 확장자에 대한 매핑을 할 때 사용.
- 오직 '/'만 포함하는 경우 어플리케이션의 기본 서블릿으로 매핑