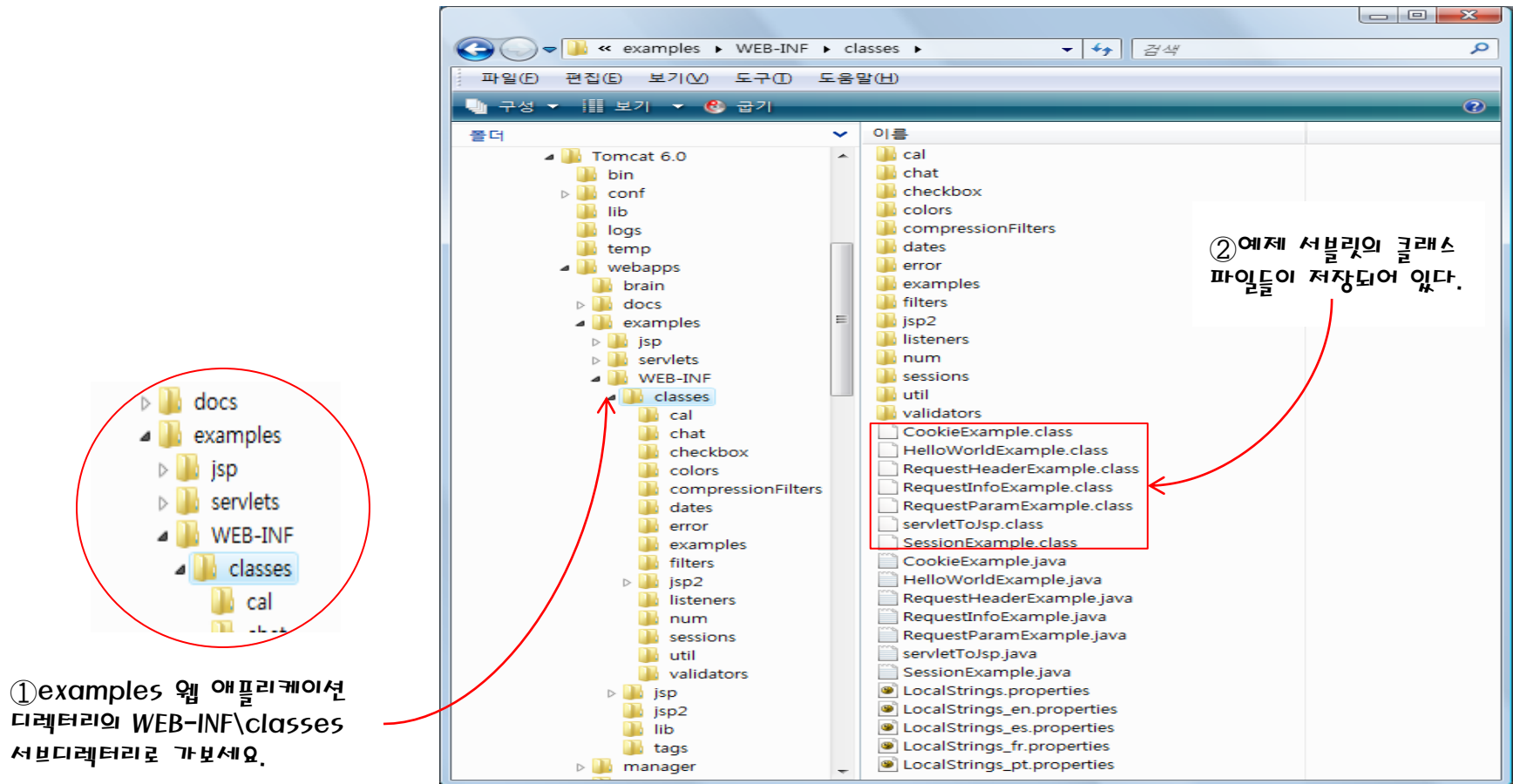


2. 서블릿 클래스의 작성, 컴파일, 설치, 등록

❖ 서블릿 클래스 설치하기

- JSP 페이지와 달리 서블릿 클래스는 소스 코드를 설치할 필요가 없고, 컴파일 결과물인 클래스 파일만 설치하면 된다.

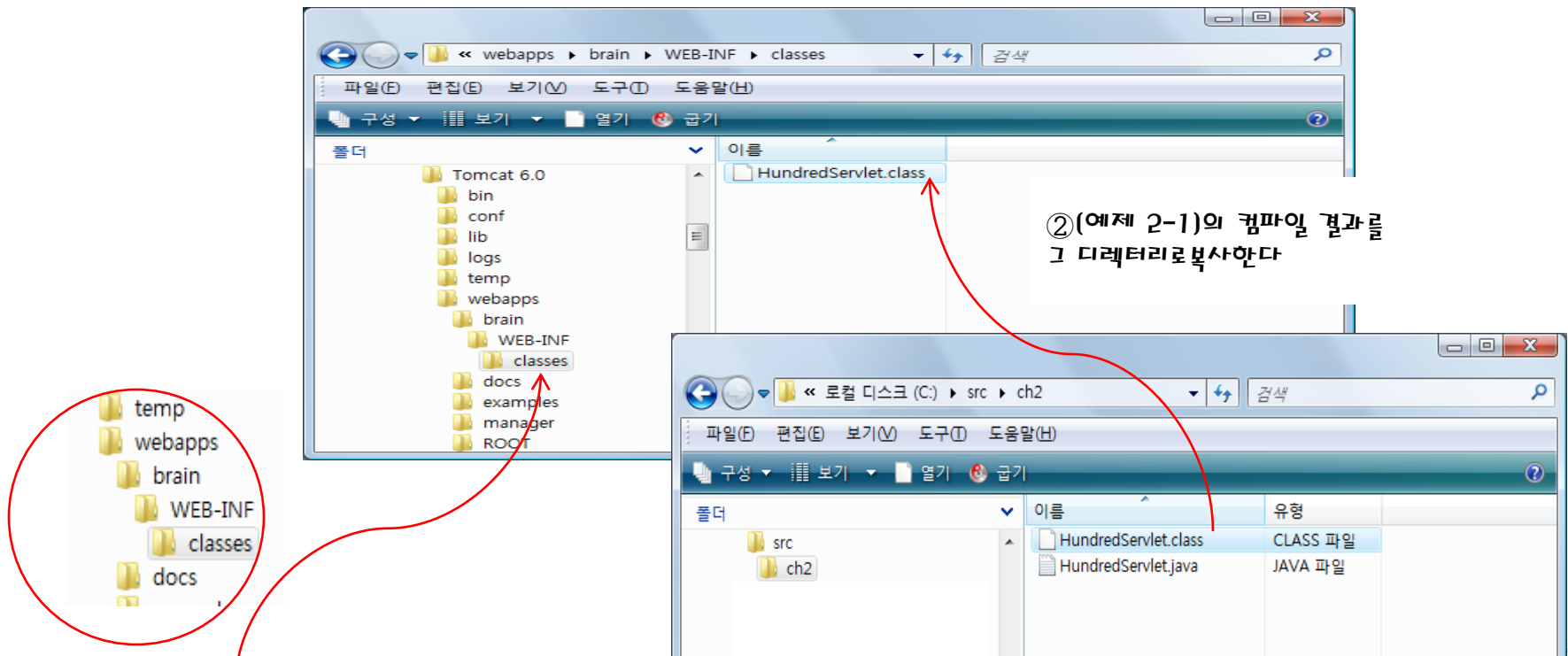


[그림 2-11] 예제 서블릿의 클래스 파일들이 저장되어 있는 디렉터리

2. 서블릿 클래스의 작성, 컴파일, 설치, 등록

❖ 서블릿 클래스 설치하기

- 톰캣의 webapps\brain 디렉터리로 가서 WEB-INF라는 서브디렉터리를 만들고, 그 아래에 classes라는 서브디렉터리를 만든 후, 컴파일 결과물인 HundredServlet.class 파일을 저장한다.



2. 서블릿 클래스의 작성, 컴파일, 설치, 등록

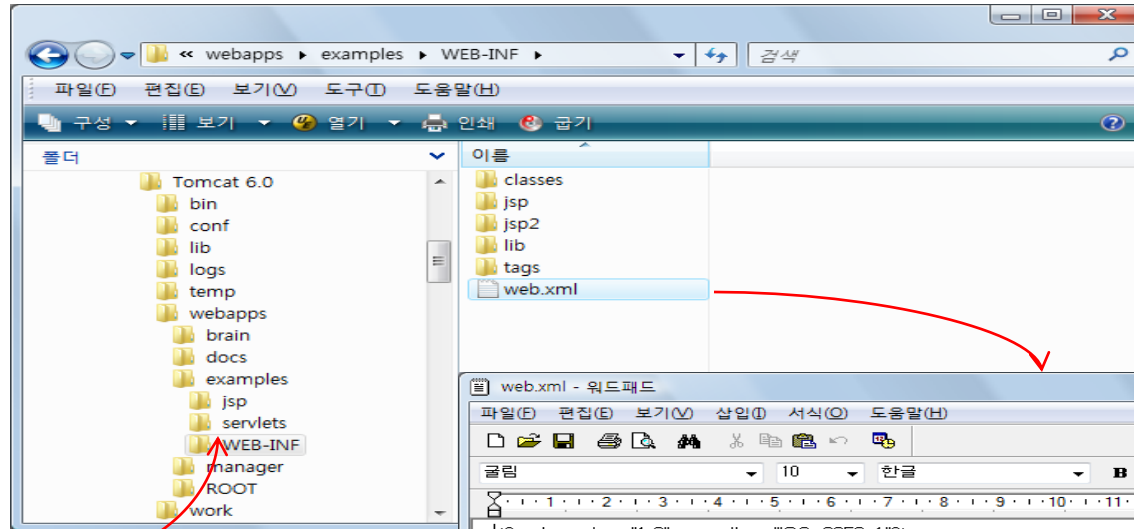
❖ 서블릿 클래스 등록하기

- 서블릿 클래스는 JSP 페이지와 달리 설치뿐만 아니라 등록 과정도 필요하다.
- 웹 애플리케이션의 디플로이먼트 디스크립터 파일에 등록해야 한다.
- 웹 애플리케이션의 디플로이먼트 디스크립터 파일란 웹 애플리케이션 디렉터리의 WEB_INF 서브디렉터리 아래 있는 web.xml이라는 이름의 파일을 말한다.
예: 톰캣의 webapps\examples\ WEB_INF 디렉터리에 있는 web.xml 파일
- XML 파일이고, 텍스트 에디터를 이용해서 열어 볼 수 있다.
- 주의: 톰캣에 있는 web.xml 파일 중에는 UNIX 포맷으로 만들어진 것도 있는데, 이런 파일은 메모장으로 열면 줄 바꿈 표시가 제대로 되지 않을 수 있다. 그럴 때는 메모장 대신 워드패드처럼 기능이 풍부한 텍스트 에디터를 이용해서 여는 것이 좋다.



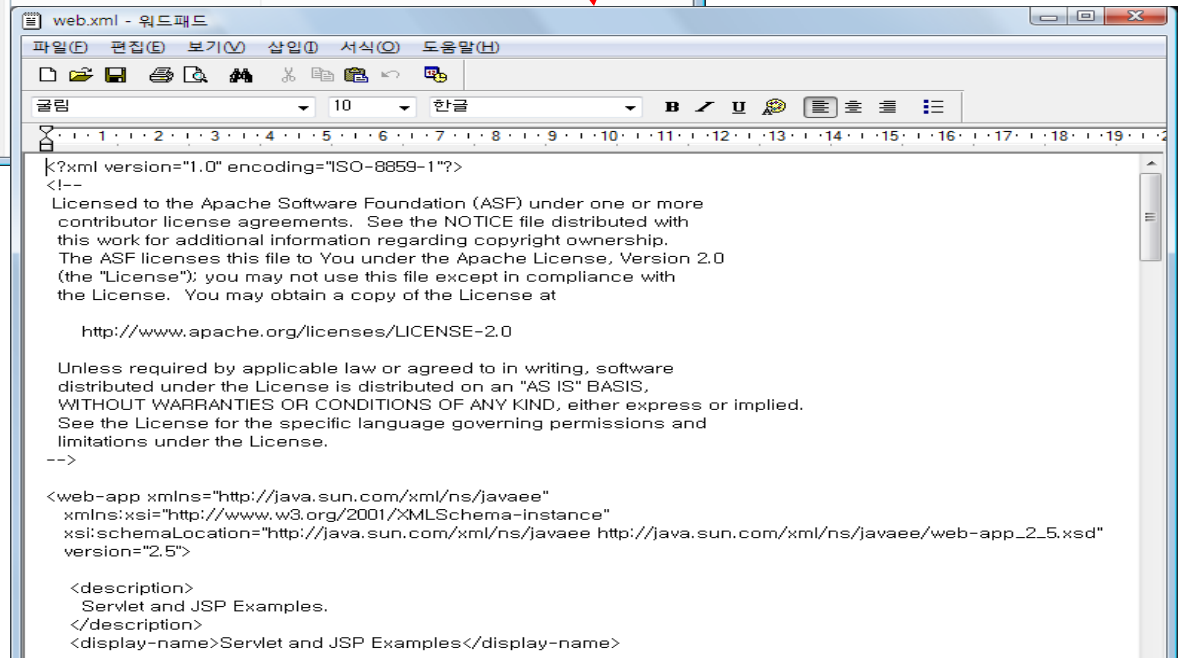
2. 서블릿 클래스의 작성, 컴파일, 설치, 등록

❖ 서블릿 클래스 등록하기



② 거기에 있는 web.xml 파일을 텍스트 에디터로 열어보세요

① 톰캣의 examples 웹 애플리케이션의 WEB-INF 서브디렉터리로 가세요.



2. 서블릿 클래스의 작성, 컴파일, 설치, 등록

❖ 알아두세요(XML 문법의 기초)

- XML은 HTML과 마찬가지로 텍스트 내용에 태그(또는 마크업)를 첨가하기 위해 사용되는 문법이다.
- XML과 HTML의 서로 다른 점은 다음과 같다.
 - 첫째 : XML 문서의 제일 앞에는 XML 선언이 올 수 있다. XML 선언은 XML 문서의 작성에 사용된 XML 규격서의 버전, XML 문서를 저장하는 데 사용된 문자 코드의 인코딩 방식을 표시하는 역할을 한다.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

XML 규격서의 버전

문자셋의 인코딩 방식

단, XML 문서가 ASCII 문자로만 구성되었을 경우에는 XML 선언을 생략할 수도 있다.

- 둘째 : HTML에서는 모든 HTML 문서의 작성 방법이 동일하지만, XML에서는 XML 문서의 종류에 따라 작성 방법이 달라질 수 있다.



2. 서블릿 클래스의 작성, 컴파일, 설치, 등록

❖ 알아두세요(XML 문법의 기초)

- 셋째 : HTML에서는 엘리먼트 이름과 애트리뷰트 이름에 있는 대소문자를 구분하지 않지만 XML에서는 엄격하게 구분한다.



- 넷째 : HTML에서는 단독으로 사용되는 태그가 <로 시작해서> 로 끝나야 하지만, XML에서는 <로 시작해서 />로 끝나야 한다.



2. 서블릿 클래스의 작성, 컴파일, 설치, 등록

❖ 알아두세요(XML 문법의 기초)

- 다섯째 : HTML에서는 애트리뷰트 값을 따옴표로 묶지 않고 쓸 수도 있지만, XML에서는 반드시 따옴표로 묶어서 써야한다.

★HTML 문서★

```
<IMG src=/img/car.jpg >
```

올바른 문법

★XML 문서★

```
<role rolename=admin />
```

잘못된 문법

- 여섯째 : HTML에서는 다소 문법이 맞지 않는 부분이 있으면 웹 브라우저가 이를 보정해서 처리하지만, XML에서는 문법에 조금이라도 맞지 않는 부분이 있으면 XML 문서 전체가 올바르게 처리 되지 않는다.

★HTML 문서★

```
<H1>회사 개요</H2>
```

딱이 맞지 않아도
웹 브라우저가 보정해 준다

★XML 문서★

```
<servlet-name>MyServlet</servlet-class>
```

딱이 맞지 않으면 문서 전체가
올바르게 처리되지 않는다.



2. 서블릿 클래스의 작성, 컴파일, 설치, 등록

❖ 서블릿 클래스 등록하기

- web.xml 파일은 웹 애플리케이션 디렉터리마다 하나씩만 만들 수 있다.
- web.xml 파일을 새로 만들 때는 루트 엘리먼트인 <web-app>을 만드는 일부터 시작하는 것이 좋다.

```
<web-app>  
</web-app>
```

web.xml 파일의 루트 엘리먼트

- 웹 서버가 웹 브라우저로부터 URL을 받았을 때 서블릿 클래스를 찾아서 호출하기 위해 필요한 두 정보는 다음의 두 엘리먼트 안에 기록해야 한다.

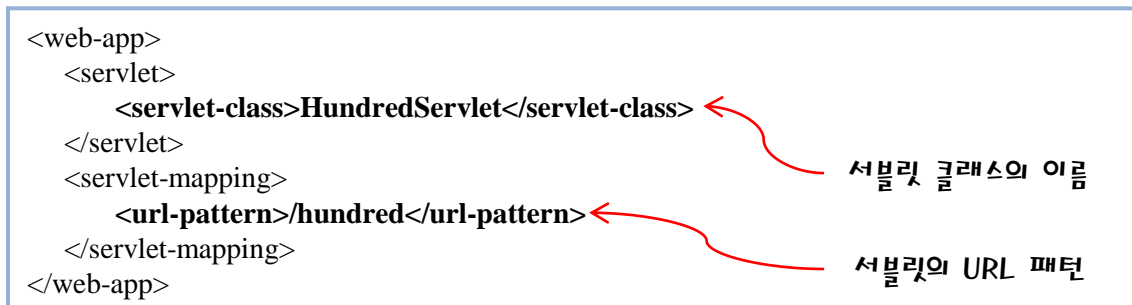
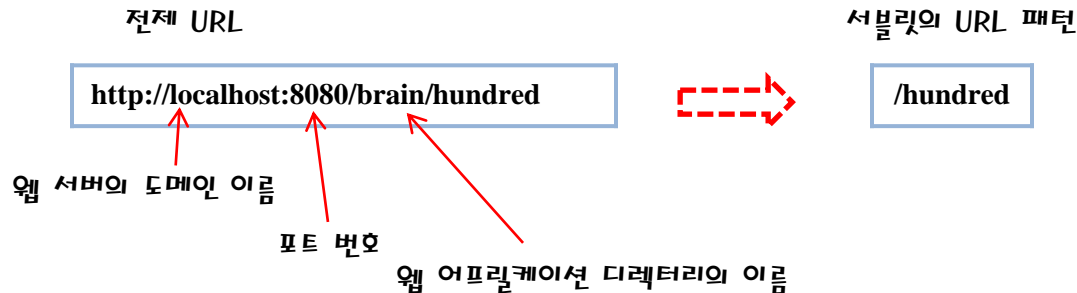
```
<web-app>  
  <servlet>  
    ← 서블릿 클래스의 이름이 들어갈 부분  
  </servlet>  
  <servlet-mapping>  
    ← 서블릿 클래스를 호출할 때 사용할 URL이 들어갈 부분  
  </servlet-mapping>  
</web-app>
```



2. 서블릿 클래스의 작성, 컴파일, 설치, 등록

❖ 서블릿 클래스 등록하기

- `<servlet-mapping>` 엘리먼트 안에는 전체 URL이 아니라, 웹 서버의 도메인 이름, 포트 번호, 웹 어플리케이션 디렉터리 이름을 제외한 나머지 부분만 써야 한다.



2. 서블릿 클래스의 작성, 컴파일, 설치, 등록

❖ 서블릿 클래스 등록하기

- web.xml 파일에는 나중에 다른 <servlet>, <servlet-mapping> 엘리먼트가 추가 될 수도 있으므로, 연관된 두 엘리먼트가 서로 찾을 수 있도록 연결시켜야 한다.

방법: <servlet> 엘리먼트 안에 서블릿 식별자를 쓰고, <servlet-mapping> 엘리먼트에서 그 식별자를 이용해서 <servlet> 엘리먼트를 참조하도록 만든다.

```
<web-app>
  <servlet>
    <servlet-name>hundred-servlet</servlet-name>
    <servlet-class>HundredServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>hundred-servlet</servlet-name>
    <url-pattern>/hundred</url-pattern>
  </servlet-mapping>
</web-app>
```

참조



2. 서블릿 클래스의 작성, 컴파일, 설치, 등록

❖ 서블릿 클래스 등록하기

- 모든 web.xml 파일 안에 반드시 써 넣어야 하는 두 가지 정보

web.xml 파일의 작성에 사용된 문법의 식별자

```
<web-app xmlns="http://java.sun.com/xml/ns/javaee"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
          version="2.5">
```

그 문법의 버전



2. 서블릿 클래스의 작성, 컴파일, 설치, 등록

❖ 서블릿 클래스 등록하기

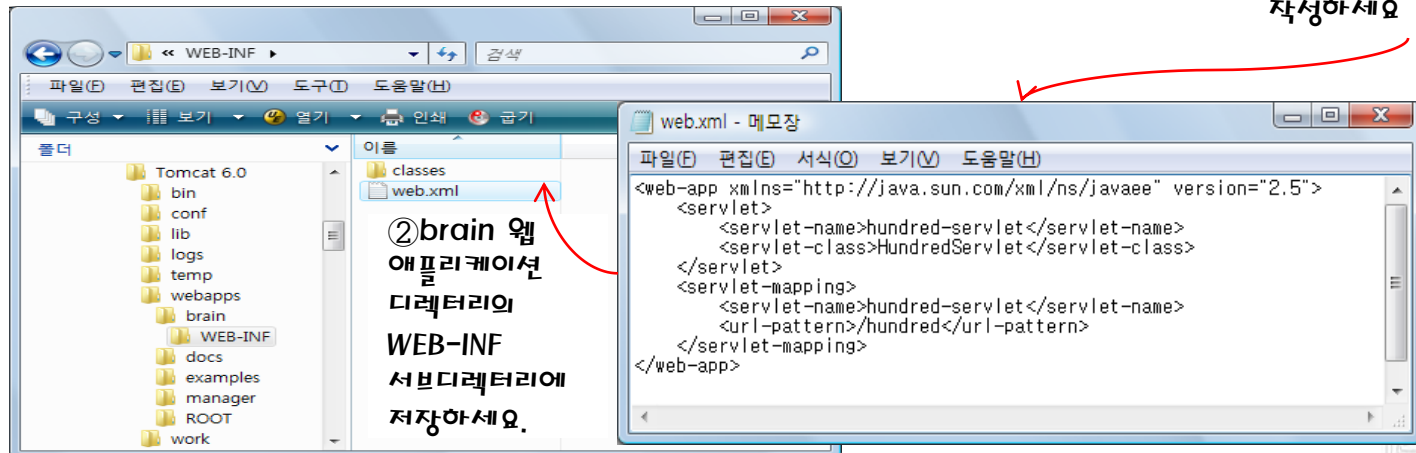
- <web-app> 엘리먼트의 시작 태그에 다음과 같은 내용을 추가하라.

web.xml 파일의 작성에 사용된 문법의 식별자

그 문법의 버전

```
<web-app xmlns="http://java.sun.com/xml/ns/javaee" version="2.5">
  <servlet>
    <servlet-name>hundred-servlet</servlet-name>
    <servlet-class>HundredServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>hundred-servlet</servlet-name>
    <url-pattern>/hundred</url-pattern>
  </servlet-mapping>
</web-app>
```

① 텍스트 에디터로
web.xml 파일을
작성하세요

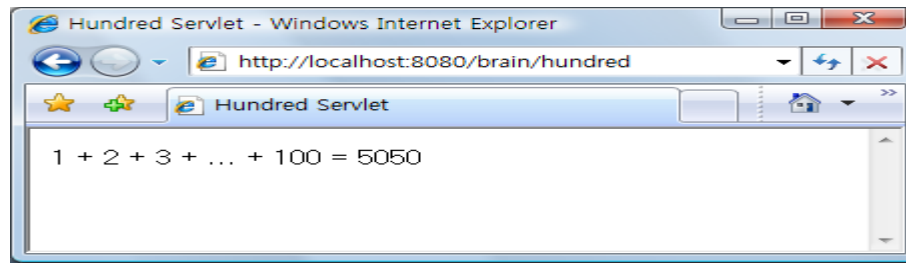


[그림 2-14] 서블릿 클래스를 web.xml 파일에 등록하는 방법

2. 서블릿 클래스의 작성, 컴파일, 설치, 등록

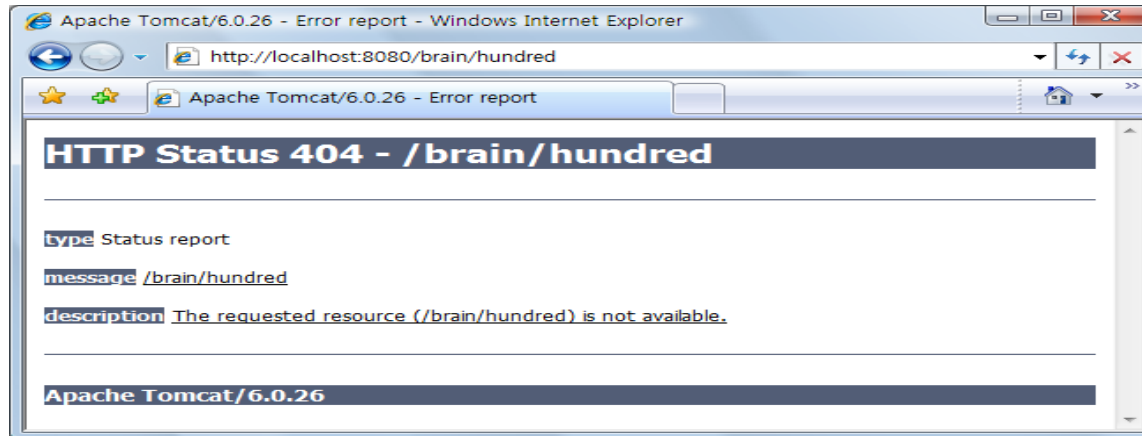
❖ 서블릿 실행하기

- 웹 브라우저를 열고 주소 창에 `http://localhost:8080/brain/hunderd`라는 URL을 입력했을 때 나오는 [예제 2-1]의 정상적인 실행 결과이다.



[그림 2-15] 예제 2-1의 실행 결과 - 성공적인 결과

- 등록 과정이 잘못되었다면 다음과 같은 에러 페이지가 나타난다.



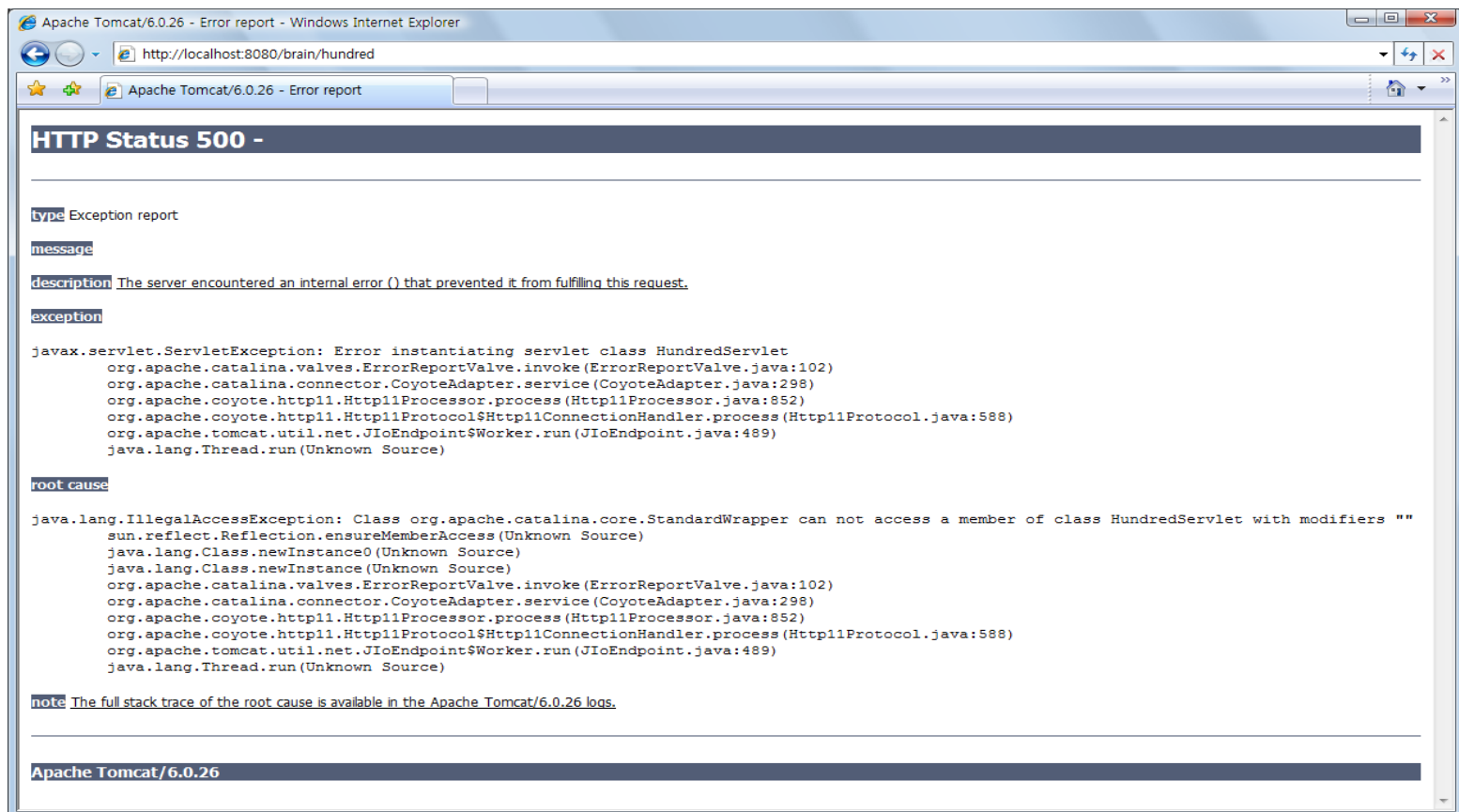
[그림 2-16] 예제 2-1의 실행 결과 - 서블릿의 등록이 잘못되었을 때



2. 서블릿 클래스의 작성, 컴파일, 설치, 등록

❖ 서블릿 실행하기

- 서블릿 클래스의 소스 코드에서 잘못된 부분이 있다면 다음과 같은 에러 페이지가 나타난다.



[그림 2-17] 예제 2-1의 실행 결과 - 서블릿 클래스의 소스 코드가 잘못되었을 때

3. 톰캣 관리자 프로그램 사용하기

❖ 톰캣 관리자 프로그램의 사용 방법

1 톰캣의 URL을 입력하세요.

2 'Tomcat Manager' 링크를 선택하면 로그인 창이 나타납니다.

localhost에 연결

Tomcat Manager Application의 서버 localhost를 <클> 사용하려면 사용자 이름과 암호가 필요합니다.
결과: 이 서버에서 연결한 것은 잘못(불안 연결 없이 리본 인증)으로 사용자 이름과 암호를 보냈으므로 연결하고 있습니다.

사용자 이름(U): admin
암호(P):
☐ 암호 저장(S)

확인 취소

3 관리자 아이디와 패스워드를 입력하고 '확인' 버튼을 누르면 톰캣 관리자 프로그램의 화면이 나타납니다.

웹 애플리케이션의 목록

Path	Display Name	Running	Sessions	Commands
/	Welcome to Tomcat	true	1	Start Stop Reload Undeploy Expire sessions with idle > 30 minutes
/docs	Tomcat Documentation	true	1	Start Stop Reload Undeploy Expire sessions with idle > 30 minutes
/examples	Servlet and JSP Examples	true	1	Start Stop Reload Undeploy Expire sessions with idle > 30 minutes
/manager	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle > 30 minutes

- 톰캣의 아이디와 패스워드가 기억나지 않는다면 톰캣의 conf 디렉터리에 있는 tomcat-users.xml 파일을 열어서 확인한다.

3. 톰캣 관리자 프로그램 사용하기

❖ 한글 HTML 문서를 출력하는 서블릿 클래스

- 한글이 포함된 HTML 문서를 출력하려면 doGet, doPost 메서드의 두 번째 파라미터인 HttpServletResponse 타입의 파라미터에 대해 다음과 같은 메서드를 호출해야 한다.

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

이 문서의 내용은 HTML 문법으로 작성된 텍스트이고

euc-kr 문자셋(한글 코드)로 인코딩되어 있음

- 이 명령문은 HTML을 출력하는 print, println, printf 메서드 호출문보다 앞에 와야 할 뿐만 아니라, response.getWriter 메서드 호출문보다도 먼저 와야 한다.



3. 톰캣 관리자 프로그램 사용하기

❖ 한글 HTML 문서를 출력하는 서블릿 클래스

- 한글을 포함한 HTML 문서를 출력하는 서블릿 클래스는 다음과 같다.

[예제2-2] 1부터 100까지 더하는 서블릿 클래스

```
import javax.servlet.http.*;
import javax.servlet.*;
import java.io.*;

public class HundredServlet extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws IOException, ServletException {
        int total = 0;
        for (int cnt = 1; cnt < 101; cnt++)
            total += cnt;
        response.setContentType( "text/html;charset=euc-kr " );
        PrintWriter out = response.getWriter();
        out.println( "<HTML> " );
        out.println( "<HEAD><TITLE>1부터 100까지 더하는 서블릿</TITLE></HEAD> " );
        out.println( "<BODY> " );
        out.printf( "1부터 100까지의 합은 = %d " , total);
        out.println( "</BODY> " );
        out.println( "</HTML> " );
    }
}
```

