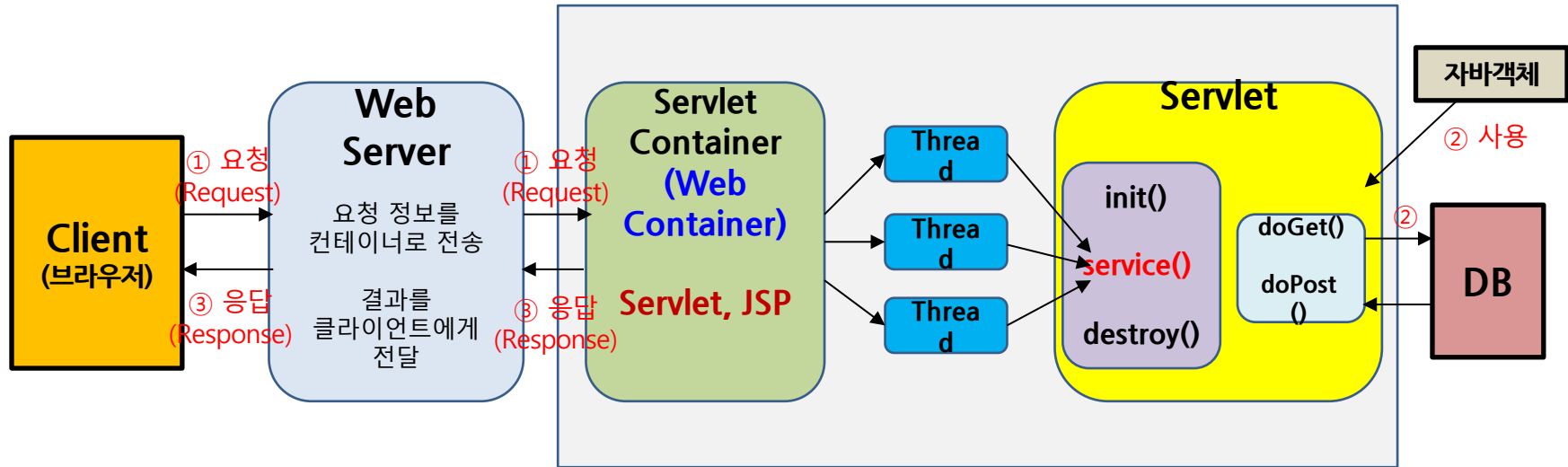


Servlet Program의 기본적인 동작 과정

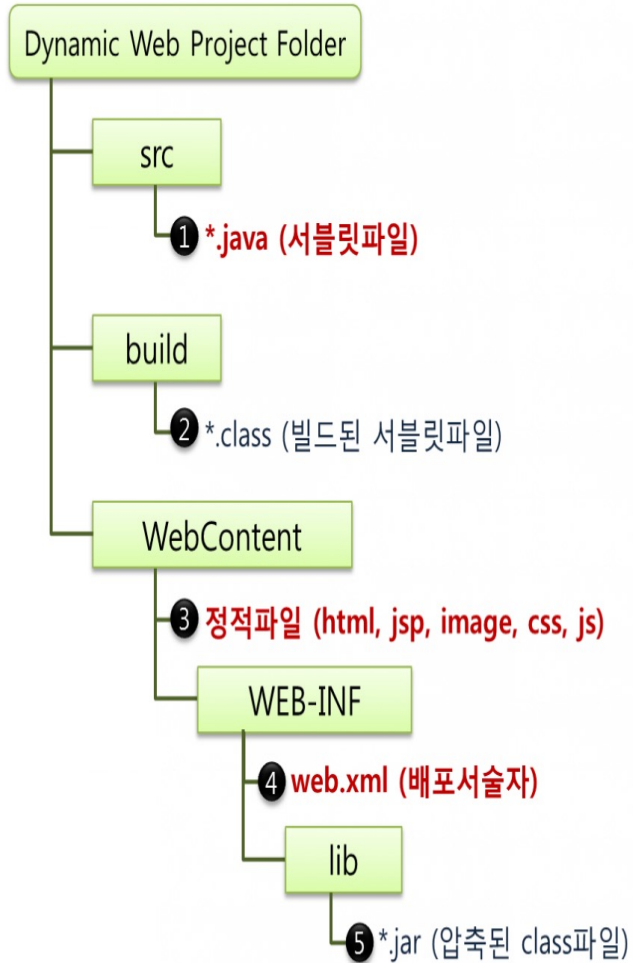


1. Web Server는 HTTP request를 Web Container(Servlet Container)에게 위임한다.
 - 1) web.xml 설정에서 어떤 URL과 매핑되어 있는지 확인
 - 2) 클라이언트(browser)의 요청 URL을 보고 적절한 Servlet을 실행
2. Web Container는 service() 메서드를 호출하기 전에 Servlet 객체를 메모리에 올린다.
 - 1) Web Container는 적절한 Servlet 파일을 컴파일(.class 파일 생성)한다.
 - 2) .class 파일을 메모리에 올려 Servlet 객체를 만든다.
 - 3) 메모리에 로드될 때 Servlet 객체를 초기화하는 init() 메서드가 실행된다.
3. Web Container는 Request가 올 때마다 thread를 생성하여 처리한다.

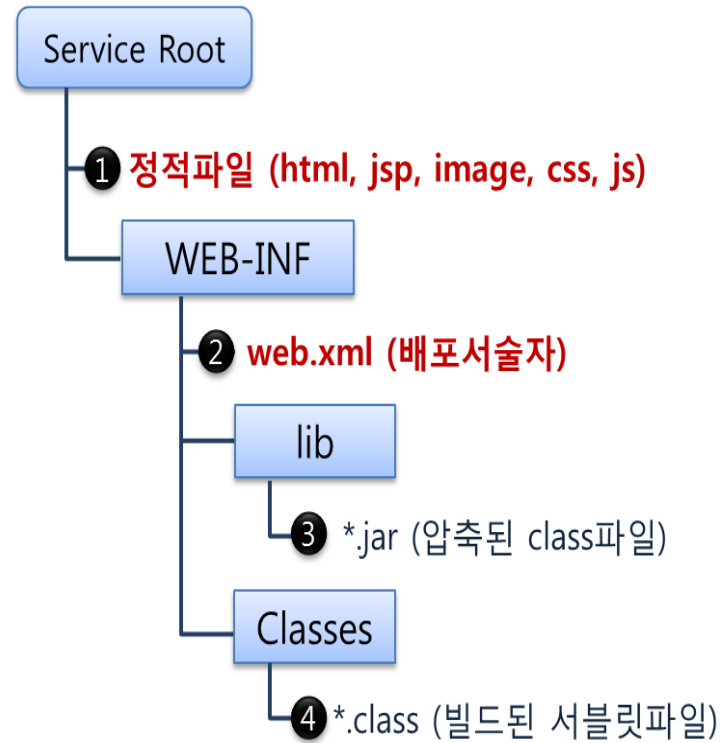
각 thread는 Servlet의 단일 객체에 대한 service() 메서드를 실행한다.

웹 프로젝트 개발 환경의 폴더 구조

Eclipse 개발 환경의 폴더 구조



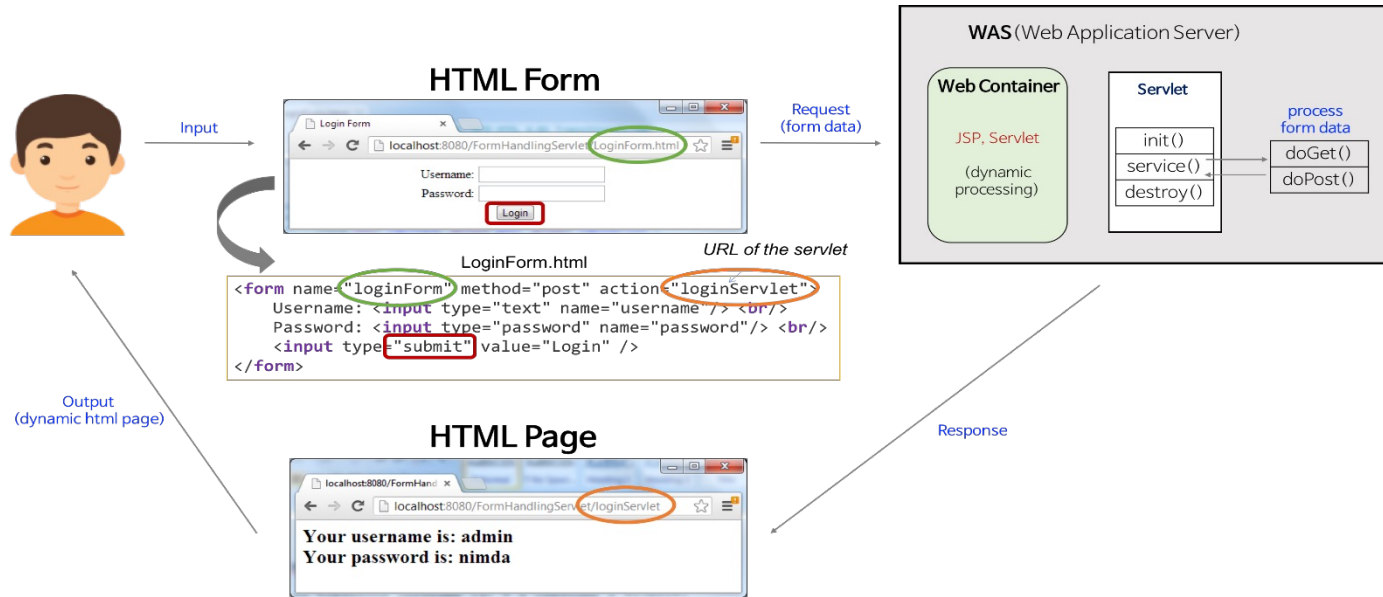
웹 서버의 배포 폴더 구조



Web.xml에서 servlet매핑 예제

```
<servlet>  
  <servlet-name> 서블릿 이름 </servlet-name>  
  <servlet-class> 클래스 전체 이름 </servlet-class>  
</servlet>  
  
<servlet-mapping>  
  <servlet-name> 서블릿 이름 </servlet-name>  
  <url-pattern> /호출url </url-pattern>  
</servlet-mapping>
```

Web Service의 기본적인 동작 과정



HTML Form -> Servlet -> HTML Page

1. 사용자가 웹 페이지 form(HTML Form)을 통해 자신의 정보를 입력한다. **(Input)**
2. Servlet의 `doGet()` 또는 `doPost()` 메서드는 입력한 form data에 맞게 DB 또는 다른 소스에서 관련된 정보를 검색한다.
3. 이 정보를 이용하여 사용자의 요청에 맞는 적절한 동적 컨텐츠(HTML Page)를 만들어서 제공한다. **(Output)**

1. HTML Form

예) LoginForm.html

```
<form name="loginForm" method="post" action="/login/loginServlet.do">  
    Username: <input type="text" name="username"> <br>  
    Password: <input type="password" name="password"> <br>  
    <input type="submit" value="Login">  
</form>
```

Form Tag 속성

- Form Tag 속성을 이용하여 어디로, 어떤 방식으로 전송할지 정한다.
- **action** : form을 전송할 서버 쪽 스크립트 파일 지정
- **name** : form을 식별하기 위한 이름 지정
- **accept-charset** : form 전송에 사용할 문자 인코딩을 지정
- **target** : action에서 지정한 스크립트 파일을 현재 창이 아닌 다른 위치에서 열리도록 지정
- **method** : form을 서버에 전송하는 방식으로, HTTP 메서드 지정 (GET 또는 POST)
- **enctype** : 넘기는 Content의 Type 지정 (주로 파일을 넘길 때 사용)

type은 *multipart/form-data* 로 지정해서 사용

Form Methods

- form을 서버에 전송하는 방식으로, 두 가지의 HTTP 메서드를 지정할 수 있다.

1. GET Method

- 사용자가 입력한 내용(form data)이 URL 뒤에 텍스트 문자열로 추가된다.

- 1) 크기 제한: **1024 characters**

- 2) data는 ?를 기준으로 action URL과 분리된다.

예) `http://www.test.com/hello?key1=value1&key2=value2`

- 브라우저에서 웹 서버로 정보를 전달하는 기본 메서드(Default Method)

- 1) HTTP 메서드를 지정하지 않으면 GET Method를 호출한다.

- 암호와 같은 민감한 정보를 서버에 전달하는 경우에는 GET을 사용하지 않는다.

- 1) URL은 모두에게 노출되는 정보이기 때문에 보안상 적절하지 않다.

- GET 메서드의 사용

- 1) DB에 영향을 주지 않는 단순히 읽기 위주(read operation)의 작업

- 2) 몇 번이고 반복해서 처리해도 같은 결과가 나오는 작업

Form Methods

2. POST Method

- 사용자가 입력한 내용(form data)을 별도의 메시지로 보낸다.
- Request Body에 data를 추가한다.

1) URL에 직접적으로 data가 노출되지 않기 때문에 GET Method보다 보안상으로 조금 더 안전하다.

- POST 메서드의 사용

1) DB에 영향을 주는 작업