

11 : 웹 애플리케이션 배포



학습 목표

- 웹 애플리케이션 배포 시 필요한 것들을 알아본다.
- war 파일과 환경 설정에 대해 이해한다.

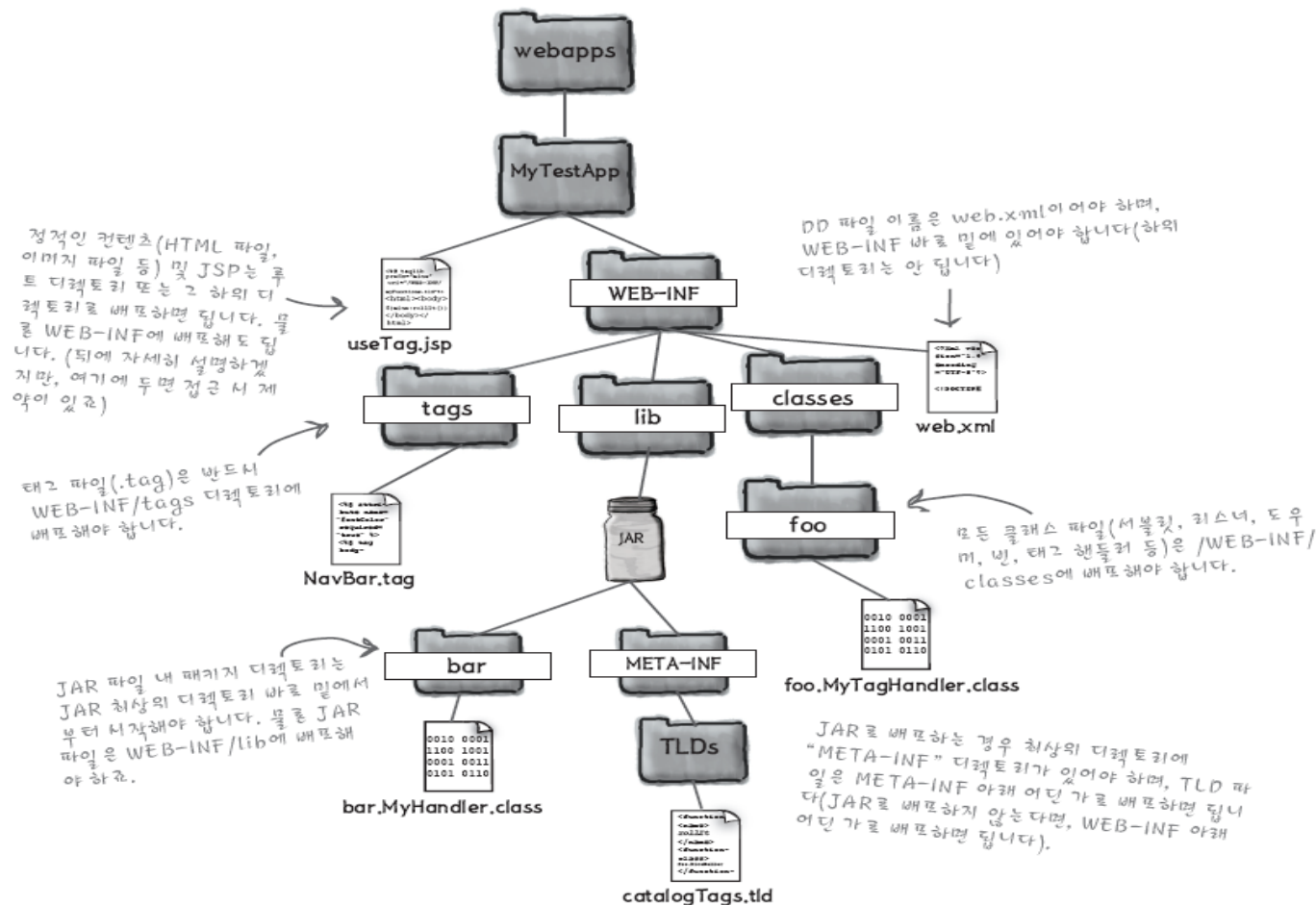


1. 배포 디렉토리 구성
2. WAR 파일
3. 환경 파일 (web.xml)



배포 >> 디렉토리 구성

■ 웹 애플리케이션 디렉토리 구성





배포 >>WAR 파일

■ WAR 파일

- 웹 애플리케이션 디렉토리를 압축(WEB-INF 바로 위에서 압축)한 파일
- 확장자 : .war
- 파일명 : 웹 애플리케이션 명
- 배포 : webapps 디렉토리에 복사



배포 >> WAR 파일

직접 접근할 수 있는 위치

올바른 요청

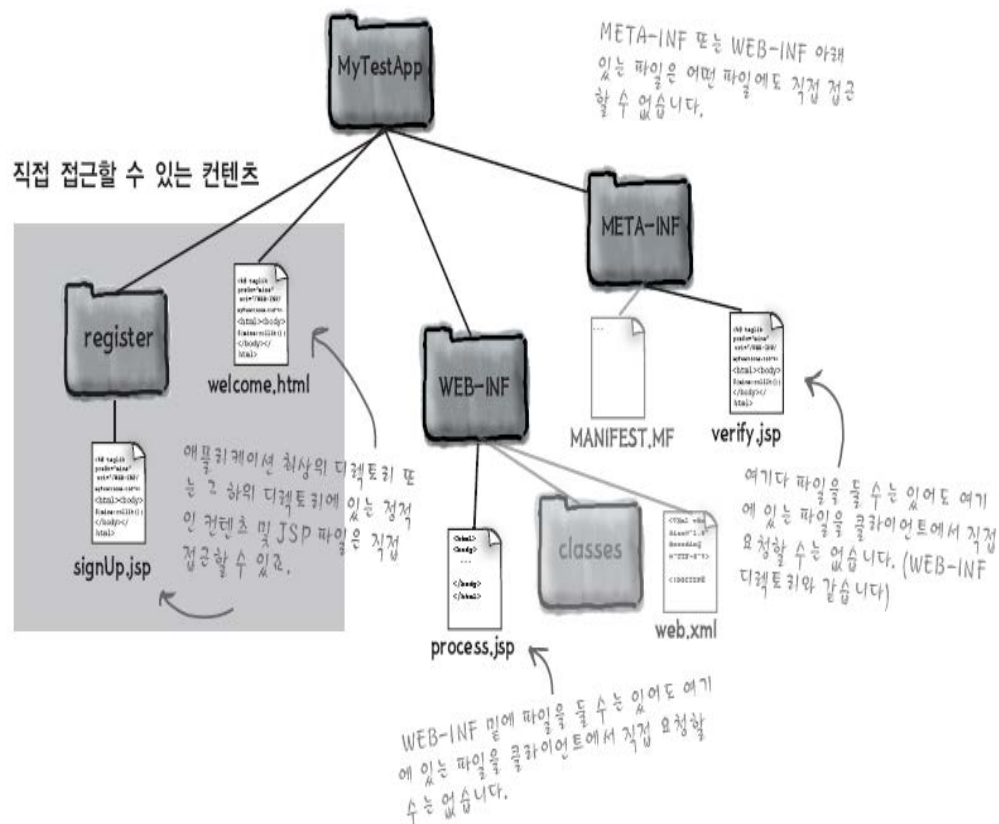
<http://www.wickedlysmart.com/MyTestApp/register/signUp.jsp>

문제 있는 요청("404 Not Found" 오류가 납니다)

<http://www.wickedlysmart.com/MyTestApp/WEB-INF/process.jsp>

웹 애플리케이션에 직접 접근하기
위한 경로

안 됩니다. WEB-INF 아래 있는
파일은 어떤 파일에도 직접 접근할
수 없습니다.





배포 >> WAR 파일

■ 서블릿 매핑 : web.xml

```
<web-app xmlns="http://java.sun.com/xml/ns/j2ee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd"
  version="2.4">
```

<servlet>

<servlet-name>*Beer***</servlet-name>**

<servlet-class>com.example.BeerSelect**</servlet-class>**

</servlet>

<servlet-mapping>

<servlet-name>*Beer***</servlet-name>**

<url-pattern>/Beer/SelectBeer.do**</url-pattern>**

</servlet-mapping>

</web-app>

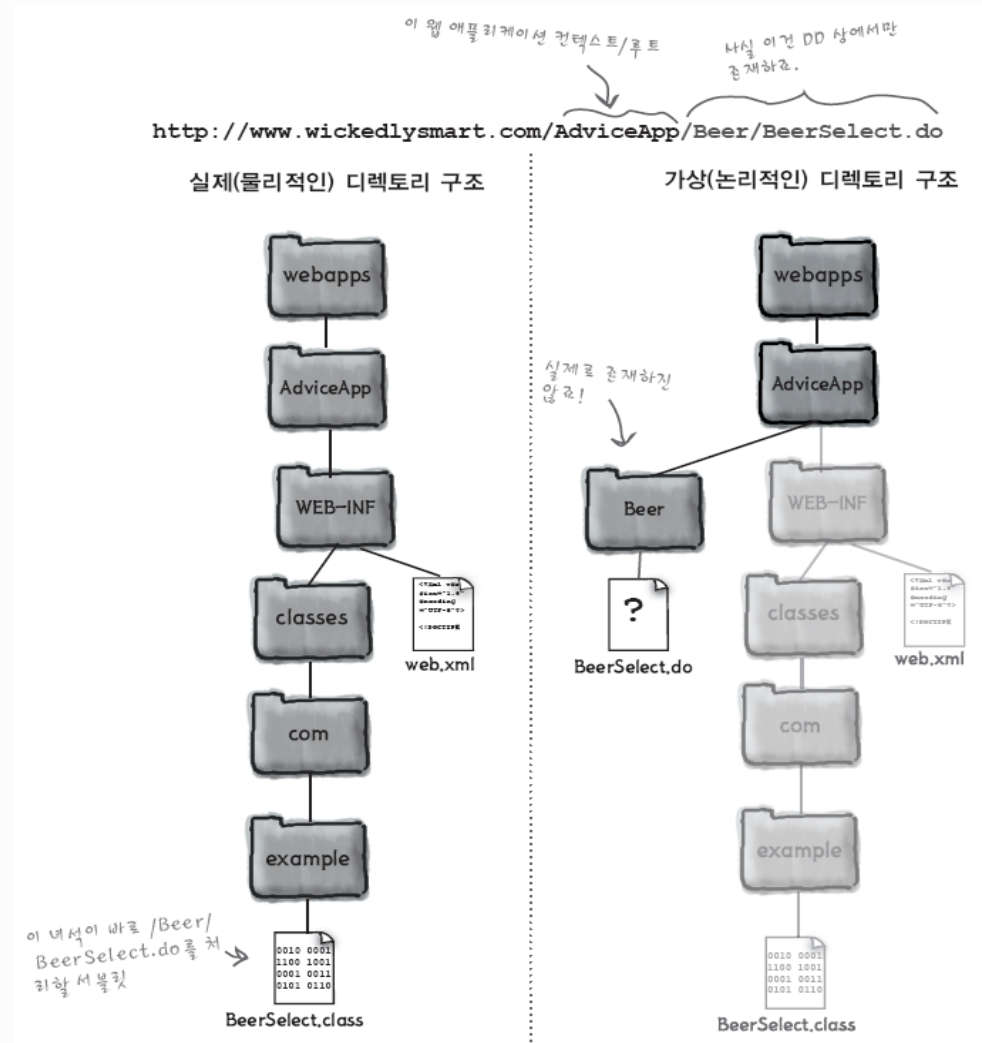
이 이름은 DD 내부에서 사용하기 위한 것입니다.
클라이언트는 이 이름을 몰라도 됩니다.

이 이름으로 요청이 들어오면, 컨테이너는
<servlet-name>에 있는 이름과 동일한
<servlet> 항목을 찾습니다. 그래서 지금 들어온
요청을 처리할 클래스가 무엇인지 알 수 있습니다.



배포 >> WAR 파일

■ 서블릿 매핑 : 디렉토리 구조





배포 >> WAR 파일

■ 서블릿 매핑 : URL 패턴

- URL 패턴 : 가상 이름
- `<url-pattern>` 은 `<servlet-name>` 으로 매핑된다

3가지 `<url-pattern>` 타입

① 정확히 이름까지 일치하는 것

```
<url-pattern>/Beer/SelectBeer.do</url-pattern>
```

반드시 /로 시작

확장자가 있어도 됨, 하지만 필수 사항은 아님.

② 디렉토리까지만 일치하는 것

```
<url-pattern>/Beer/*</url-pattern>
```

반드시 /로 시작

항상 /*로 끝남.

이것은 가상 디렉토리일 수도 실제 존재하는 디렉토리일 수도 있음.

③ 확장자만 일치하는 것

```
<url-pattern>*.do</url-pattern>
```

* 다음에 도트(.)가 반드시 있고 뒤에 확장자(.do, .jsp 등)가 있음.

*로 시작함(슬래시(/) 없음)



배포 >>환경 파일 (DD, web.xml)

■ 환영 파일

- 웹 애플리케이션의 기본 파일
- 여러 개를 지정할 수 있다. 처음 파일이 없는 경우 다음 파일로 이동

DD 코드:

```
<web-app ...>  
  <welcome-file-list>  
    <welcome-file>index.html</welcome-file>  
    <welcome-file>default.jsp</welcome-file>  
  </welcome-file-list>  
</web-app>
```

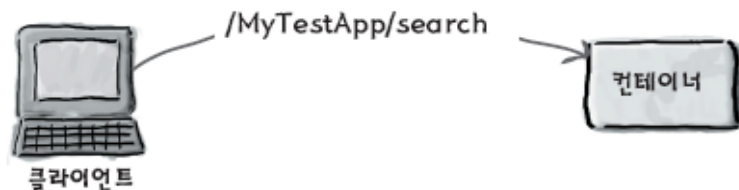
↑ 앞에 /로 시작하지 않으며 뒤에
에도 /가 없습니다.



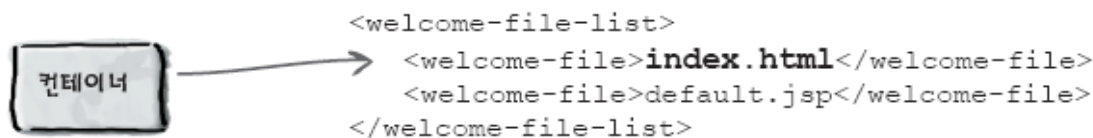
배포 >> 환경 파일 (DD, web.xml)

■ 환영 파일 선택 순서 (1)

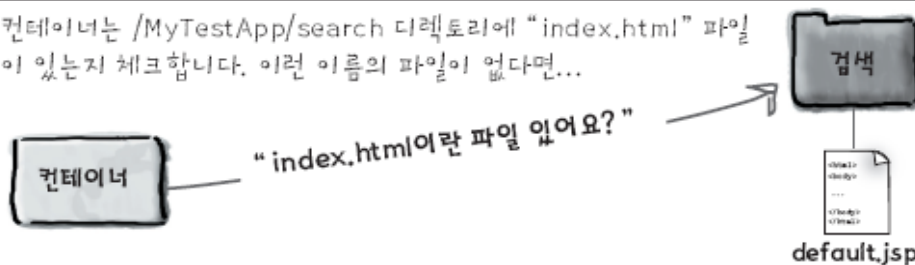
- ① 클라이언트 요청: `http://www.wickedlysmart.com/MyTestApp/search`



- ② 컨테이너는 우선 DD 서블릿 매핑란을 먼저 봅니다. 여기서 일치하는 것이 없으면, 다음으로 `<welcome-file-list>`의 제일 위에 있는 "index.html" 파일을 가지고...



- ③ 컨테이너는 `/MyTestApp/search` 디렉토리에 "index.html" 파일이 있는지 체크합니다. 이런 이름의 파일이 없다면...





배포 >> 환경 파일 (DD, web.xml)

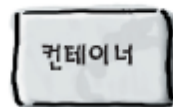
■ 환영 파일 선택 순서 (2)

④

컨테이너는 <welcome-file-list>의 두 번째 <welcome-file> 항목인 "default.jsp"를 가지고...



클라이언트



컨테이너

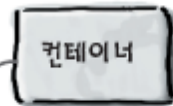
```
<welcome-file-list>
  <welcome-file>index.html</welcome-file>
  <welcome-file>default.jsp</welcome-file>
</welcome-file-list>
```

⑤

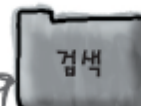
컨테이너는 /MyTestApp/search 디렉토리에 "default.jsp" 파일이 있는지 체크합니다. 이런 파일이 있다면 클라이언트에 응답으로 보냅니다.



클라이언트



컨테이너



default.jsp

응답

③

①

"default.jsp란 파일 있어요?"

②

"여기 있어요"



배포 >>환경 파일 (DD, web.xml)

■ 오류 페이지 설정

- 예외 사항별로 설정가능

```
<error-page>
  <exception-type>java.lang.ArithmeticException</exception-type>
  <location>/arithmeticError.jsp</location>
</error-page>
```

- HTTP 상태 코드별로 설정 가능

```
<error-page>
  <error-code>404</error-code>
  <location>/notFoundError.jsp</location>
</error-page>
```

- 개별 JSP 에서 page 지시자 errorPage 속성으로 재 정의 가능



배포 >>환경 파일 (DD, web.xml)

■ 시작 시 서블릿 로딩(초기화)하기

- `<load-on-startup>` : 0 보다 큰 값을 설정하면 배포 (서버 실행) 시점에 서블릿 초기화 가능

```
<servlet>
  <servlet-name>KathyOne</servlet-name>
  <servlet-class>foo.DeployTestOne</servlet-class>
  <load-on-startup>1</load-on-startup>
</servlet>
```

0보다 큰 값을 가지면 "최초 요청이 들어올 때까지 기다리지 말고, 배포 또는 서버 실행 시에 초기화하세요"라는 의미.



배포 >>환경 파일 (DD, web.xml)

■ EJB 참조 설정

● 지역 (로컬) 빈에 대한 참조

```
<ejb-local-ref>
  <ejb-ref-name>ejb/Customer</ejb-ref-name>
  <ejb-ref-type>Entity</ejb-ref-type>
  <local-home>com.wickedlysmart.CustomerHome</local-home>
  <local>com.wickedlysmart.Customer</local>
</ejb-local-ref>
```

코드에서 사용할 JNDI 검색 (lookup)명.

패키지 이름까지 모두 기술한 빈 인터페이스.

● 원격 (리모트) 빈에 대한 참조

```
<ejb-ref>
  <ejb-ref-name>ejb/LocalCustomer</ejb-ref-name>
  <ejb-ref-type>Entity</ejb-ref-type>
  <home>com.wickedlysmart.CustomerHome</home>
  <remote>com.wickedlysmart.Customer</remote>
</ejb-ref>
```

(시험에는 안 나오지만... 두 종류 모두 하위 항목으로 <description>과 <ejb-link>를 옵션으로 가질 수 있습니다)



배포 >>환경 파일 (DD, web.xml)

■ 환경 변수 지정

애플리케이션 JNDI 환경 엔트리 선언

```
<env-entry>
```

```
  <env-entry-name>rates/discountRate</env-entry-name>
```

```
  <env-entry-type>java.lang.Integer</env-entry-type>
```

```
  <env-entry-value>10</env-entry-value>
```

```
</env-entry>
```

코딩에서 사용할 검색명.

여기에 들어갈 수 있는 것으로는 생성자인
자로 String 하나를 받는 것이면 어떤 타
입이든 괜찮습니다. (또는 java.lang.
Character인 경우 Character 하나)

이 값은 String으로 넘어줍니다((env-entry-
type)이 java.lang.Character라면
Character로 넘어줍니다)

<description> 항목을 포함할 수도 있습
니다. 가급적 넣는 것이 여러 모로 좋겠군요.

- <env-entry-type> 에는 자바 기본형은 사용할 수 없음



배포 >>환경 파일 (DD, web.xml)

■ MIME Type 지정하기

〈mime-mapping〉 선언하기

```
<mime-mapping>
```

```
  <extension>mpg</extension>
```

```
  <mime-type>video/mpeg</mime-type>
```

```
</mime-mapping>
```

확장자 칸에 도트(.)는 안 들어갑니다!