

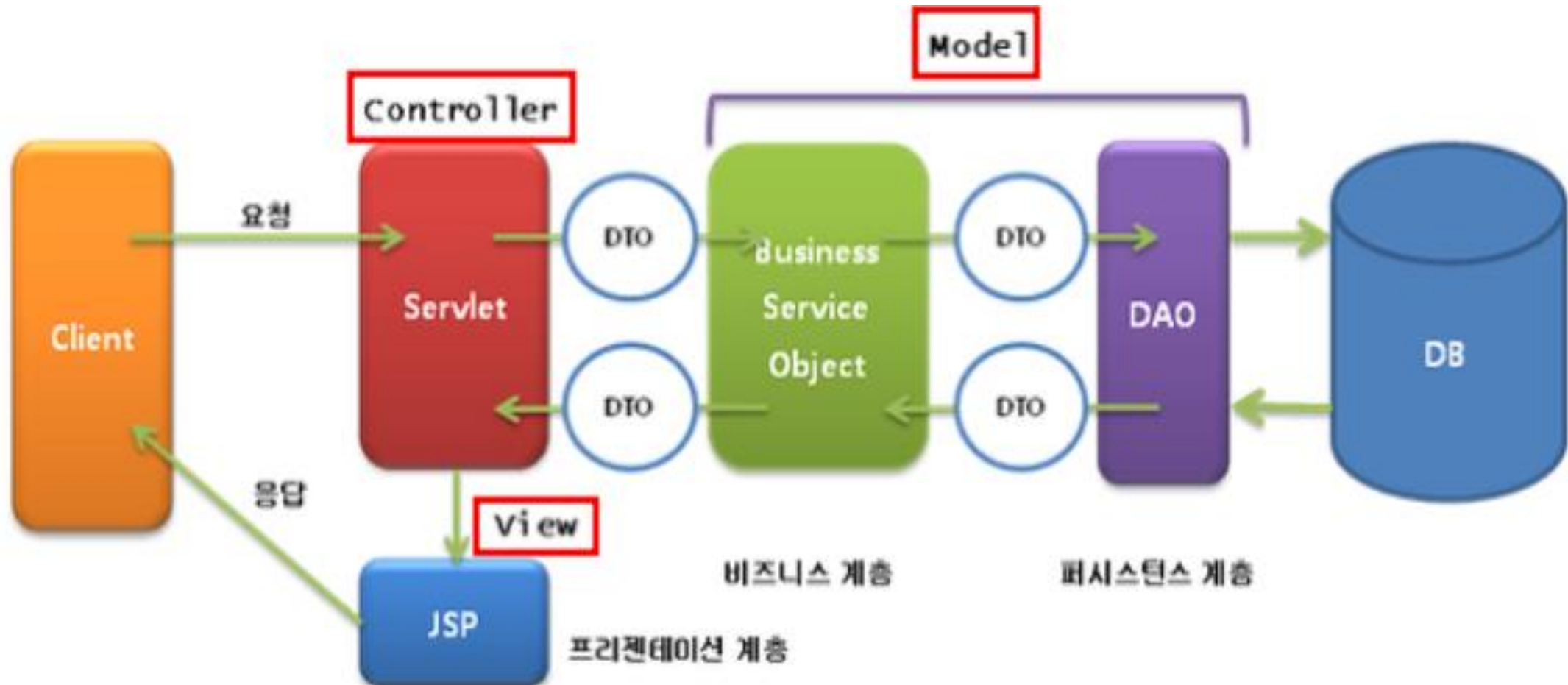


JSTL (JSP Standard Tag Library)

안 화 수

모델2 설계방식

❖ MVC Architecture



모델2 설계방식

❖ MVC 패턴

MVC는 Model / View / Controller 의 약자로 애플리케이션을 세 역할로 나누어서 개발하는 개발 방법론이다.

➤ Model

- 애플리케이션의 데이터 처리를 담당함
- Service 클래스 + DAO 클래스로 구현함

➤ View

- 사용자 인터페이스를 처리함
- JSP 를 이용해서 구현함 : EL(Expression Language, 표현언어) + JSTL(JSP Standard Tag Library)

➤ Controller

- 클라이언트의 요청을 받아 Model과 View사이에서 흐름을 조정한다.
- **Java Servlet**으로 구현함

JSTL

❖ JSTL (JSP Standard Tag Library)이란?

- JSP 표준 태그 라이브러리 (JSP Standard Tag Library)이다.
- JSP에서 자주 사용하는 기능을 구현하는 커스텀 태그 라이브러리 모음이다.
- JSP를 보완하기 위해 등장 하였다.

❖ JSTL 사용 목적

- 라이브러리 형태로 만들어진 태그를 사용하므로 개발 속도가 향상된다.
- 스크립트릿(Scriptlet)을 사용하지 않으며, 로직이 태그화 되어있기 때문에 가독성이 좋고 코드의 양이 줄어든다.
- EL을 기본으로 사용하기 때문에 null 체크와 같은 별도의 예외처리가 필요 없다.
- 다양한 Format, Function 태그를 지원한다.

JSTL

❖ JSTL 환경 구축

- JSTL 라이브러리 구하기 (방법1)

<https://tomcat.apache.org/taglibs/standard/>

Standard Taglib

JSP(tm) Standard Tag Library implementations

Apache hosts the Apache Standard Taglib, an implementation of the [JSP Standard Tag Library \(JSTL\)](#) specification. Various versions are available.

Version	JSTL version	Requirements	Getting the Taglib
Standard 1.2.3	<u>JSTL 1.2</u>	Servlet 2.5, JavaServer Pages 2.1	download (javadoc)
Standard 1.1	JSTL 1.1	Servlet 2.4, JavaServer Pages 2.0	download
Standard 1.0	JSTL 1.0	Servlet 2.3, JavaServer Pages 1.2	download

❖ JSTL 환경 구축

Apache Tomcat

[Home](#)
[Taglibs](#)
[Maven Plugin](#)

Download

[Which version?](#)
[Tomcat 11 \(alpha\)](#)
[Tomcat 10](#)
[Tomcat 9](#)
[Tomcat 8](#)
[Tomcat Migration Tool for Jakarta EE](#)
[Tomcat Connectors](#)
[Tomcat Native](#)
[Taglibs](#)
[Archives](#)

Documentation

[Tomcat 11.0 \(alpha\)](#)
[Tomcat 10.1](#)
[Tomcat 9.0](#)
[Tomcat 8.5](#)
[Tomcat Connectors](#)
[Tomcat Native 2](#)
[Tomcat Native 1.2](#)
[Wiki](#)
[Migration Guide](#)
[Presentations](#)
[Specifications](#)

Problems?

[Security Reports](#)
[Find help](#)

Release Integrity

You **must** [verify](#) the integrity of the downloaded files. We provide OpenPGP signatures for every release file. This signature should be matched against the [KEYS](#) file which contains the OpenPGP keys of Taglib's Release Managers. We also provide a [SHA512](#) checksum for every release file. After you download the file, you should calculate a checksum for your download, and make sure it is the same as ours.

Mirrors

You are currently using <https://dlcdn.apache.org/>. If you encounter a problem with this mirror, please select another mirror. If all mirrors are failing, there are *backup* mirrors (at the end of the mirrors list) that should be available.

Other mirrors:

Standard-1.2.5

Source Code Distributions

- [Source README](#)
- [zip](#) ([pgp](#), [sha512](#))

Jar Files

- [Binary README](#)
- Impl:
 - [taglibs-standard-impl-1.2.5.jar](#) ([pgp](#), [sha512](#))
- Spec:
 - [taglibs-standard-spec-1.2.5.jar](#) ([pgp](#), [sha512](#))
- EL:
 - [taglibs-standard-jstlel-1.2.5.jar](#) ([pgp](#), [sha512](#))
- Compat:
 - [taglibs-standard-compatible-1.2.5.jar](#) ([pgp](#), [sha512](#))

JSTL

❖ JSTL 라이브러리 구하기 (방법2)

<https://mvnrepository.com>

MVN REPOSITORY JSTL Search Categories Popular

Repository

- Central 89
- Sonatype 20
- Spring Lib M 14
- Spring Plugins 11
- Mulesoft 9
- Redhat GA 8
- EmergyaPub 6
- IBiblio 5

Group

- com.github 14
- org.apache 11
- javax.servlet 7
- org.eclipse 7
- org.glassfish 7
- io.github 5
- org.beigesoft 5
- maven-repositor... 4

Category

- Java Spec 10
- JSP Tag Library 4

Found 128 results

Sort: **relevance** | popular | newest

- 1. JSTL**
javax.servlet.jsp.jstl » jstl
JSTL
Last Release on May 14, 2015
6 usages
GPL CDDL
- 2. JSTL**
javax.servlet » jstl
JSTL
Last Release on Jun 23, 2011
2,067 usages
GPL CDDL
- 3. JSTL**
jstl » jstl
JSTL
Last Release on May 12, 2006
291 usages
- 4. JavaServer Pages(TM) Standard Tag Library**
javax.servlet.jsp.jstl » jstl-api
JavaServer Pages(TM) Standard Tag Library
Last Release on Jun 15, 2011
121 usages
GPL GPL CDDL

JSTL

- ❖ JSTL 라이브러리 구하기
- JSTL 1.2 버전을 클릭한다.



JSTL

JSTL

License	CDDL GPL 2.0
Categories	Java Specifications
Tags	servlet javax specs standard
Ranking	#232 in MvnRepository (See Top Artifacts) #15 in Java Specifications
Used By	2,067 artifacts

Note: This artifact was moved to:

[javax.servlet.jsp.jstl](#) » [jstl](#)

Central (11)

EmergyaPub (1)

Less is More (1)

	Version	Vulnerabilities	Repository	Usages	Date
1.2.x	1.2		Central	1,580	Jun 23, 2011
	1.1.2		Central	449	Nov 26, 2005
1.1.x	1.1.1		Central	10	Nov 26, 2005
	1.1.0		Central	30	Nov 26, 2005
	1.0.6		Central	29	Nov 26, 2005
1.0.x	1.0.5		Central	4	Nov 26, 2005
	1.0.4		Central	0	Nov 26, 2005
	1.0.3		Central	4	Nov 26, 2005
	1.0.2		Central	26	Nov 26, 2005
	1.0.1		Central	0	Nov 26, 2005
	1.0		Central	19	Nov 26, 2005

JSTL

❖ JSTL 라이브러리 구하기

➤ jar (404 KB) 버튼을 클릭해서 JSTL 라이브러리를 다운로드 받는다.



JSTL » 1.2

JSTL

License	CDDL GPL 2.0
Categories	Java Specifications
Tags	servlet javax specs standard
Date	Jun 23, 2011
Files	pom (356 bytes) jar (404 KB) View All
Repositories	Central Gradle Plugins OneBusAway Pub Redhat GA WSO2 Dist
Ranking	#232 in MvnRepository (See Top Artifacts) #15 in Java Specifications
Used By	2,067 artifacts

[Maven](#) [Gradle](#) [Gradle \(Short\)](#) [Gradle \(Kotlin\)](#) [SBT](#) [Ivy](#) [Grape](#) [Leiningen](#) [Buildr](#)

```
// https://mvnrepository.com/artifact/javax.servlet/jstl
implementation 'javax.servlet:jstl:1.2'
```

☒ Include comment with link to declaration

JSTL

❖ JSTL 환경 구축

- 📁 jspproject2
 - > 📄 Deployment Descriptor: jspproject2
 - > 📁 JAX-WS Web Services
 - > 📁 Java Resources
 - > 📁 build
 - ▼ 📁 src
 - ▼ 📁 main
 - > 📁 java
 - ▼ 📁 webapp
 - > 📁 el
 - > 📁 META-INF
 - > 📁 servlet
 - ▼ 📁 WEB-INF
 - ▼ 📁 lib
 - 📄 jstl-1.2.jar
 - 📄 ojdbc6.jar
 - 📄 web.xml

JSTL

❖ JSTL 태그 종류

라이브러리	하위 기능	접두어	관련URI
코어	변수지원 흐름 제어 URL 처리	c	http://java.sun.com/jsp/jstl/core
XML	XML 코어 흐름 제어 XML 변환	x	http://java.sun.com/jsp/jstl/xml
국제화	지역 메시지 형식 숫자 및 날짜 형식	fmt	http://java.sun.com/jsp/jstl/fmt
데이터베이스	SQL	sql	http://java.sun.com/jsp/jstl/sql
함수	컬렉션 처리 String 처리	fn	http://java.sun.com/jsp/jstl/functions

코어 라이브러리

기능분류	태그	설명
변수 지원	set	JSP에서 사용될 변수를 설정한다.
	remove	설정한 변수를 제거한다.
흐름 제어	if	조건에 따라 내부 코드를 수행한다.
	choose	다중 조건을 처리할 때 사용된다.
	forEach	컬렉션이나 Map의 각 항목을 처리할 때 사용된다.
	forEachTokens	구분자로 분리된 각각의 토큰을 처리할 때 사용된다.
URL 처리	import	URL을 사용하여 다른 자원의 결과를 삽입한다.
	redirect	지정한 경로로 리다이렉트 한다.
	url	URL을 재작성 한다.
기타 태그	catch	예외 처리에 사용된다.
	out	<u>JspWriter</u> 에 내용을 알맞게 처리한 후 출력한다.

코어 라이브러리

❖ 변수 지원 태그

- **set** - 변수 설정

- EL 변수 값 설정 (생성 또는 변경)

- `<c:set var="변수명" value="값" [scope="영역"] />`

- `<c:set var="변수명" value="값" [scope="영역"]>값</c:set>`

- 특정 EL 변수의 프로퍼티 값 설정

- `<c:set target="대상" property="프로퍼티이름" value="값" />`

- `<c:set target="대상" property="프로퍼티이름">값</c:set>`

- **remove** - 변수 삭제

- `<c:remove var="varName" [scope="영역"] />`

- scope 미지정시 모든 영역의 변수 삭제

코어 라이브러리

❖ 흐름 제어 태그

- **if** - 조건이 true일 경우 몸체 내용 실행

```
<c:if test="조건">  
...  
</c:if>
```

- **choose - when - otherwise**
 - switch - case - default와 동일

```
<c:choose>  
  <c:when test="${member.level == 'trial'}" >  
    ...  
  </c:when>  
  <c:when test="${member.level == 'regular'}" >  
    ...  
  </c:when>  
  <c:otherwise>  
    ...  
  </c:otherwise>  
</c:choose>
```

코어 라이브러리

❖ 반복 처리 태그

- **forEach**

- 집합이나 컬렉션 데이터 사용

```
<c:forEach var="변수" items="아이템">  
  ... ${변수사용} ...  
</c:forEach>
```

- 특정 회수 반복

```
<c:forEach var="i" begin="1" end="10" [step="값"]>  
  ${i} 사용  
</c:forEach>
```

- varStatus 속성

```
<c:forEach var="item" items="<%= somelItemList %>" varStatus="status">  
  ${status.index + 1} 번째 항목 : ${item.name}  
</c:forEach>
```

index - 루프 실행에서 현재 인덱스, count - 루프 실행 회수
begin - begin 속성 값, end - end 속성 값, step - step 속성 값
first - 현재 실행이 첫 번째 실행인 경우 true
last - 현재 실행이 루프의 마지막 실행인 경우 true
current - 컬렉션 중 현재 루프에서 사용할 객체

코어 라이브러리

❖ URL 관련 태그

- **import** - 외부/내부 페이지를 현재 위치에 삽입

```
<c:import url="URL" [var="변수명"] [scope="영역"] [charEncoding="캐릭터셋"]>  
  <c:param name="파라미터이름" value="값" />  
  ...  
</c:import>
```

– 상대 URL import 시 `<jsp:include>`와 동일하게 동작

- **url** - 절대 URL과 상대 URL을 알맞게 생성

```
<c:url value="URL" [var="varName"] [scope="영역"]>  
  <c:param name="이름" value="값" />  
</c:url>
```

– 웹 컨텍스트 내에서 절대 경로 사용시 컨텍스트 경로 자동 추가

- **redirect** - 지정한 페이지로 리다이렉트

```
<c:redirect url="URL" [context="컨텍스트경로"]>  
  <c:param name="이름" value="값" />  
</c:redirect>
```


코어 라이브러리

❖ 기타 코어 태그

- **out** - 데이터를 출력

```
<c:out value="value" [escapeXml="(true|false)"] [default="defaultValue"] />  
  
<c:out value="value" [escapeXml="(true|false)"]>  
    default value  
</c:out>
```

– escapeXml 속성이 true일 경우 다음과 같이 특수 문자 처리

- < → < , > → >
- & → & , ' → ' , " → "

- **catch** - 몸체에서 발생한 예외를 변수에 저장

```
<c:catch var="exName">  
...  
예외가 발생할 수 있는 코드  
...  
</c:catch>  
${exName} 사용
```

국제화 라이브러리

기능분류	태그	설명
로케일 지정	setLocale	Locale을 지정한다.
	requestEncoding	요청 <u>파라미터의</u> 캐릭터 <u>인코딩을</u> 지정
메시지 처리	bundle	사용할 번들을 지정
	message	지역에 알맞은 메시지를 출력
	setBundle	리소스 번들을 읽어와 특정 변수에 저장
숫자 및 날짜 포매팅	formatNumber	숫자를 <u>포매팅</u>
	formatDate	Date 객체를 <u>포매팅</u>
	parseDate	문자열로 표시된 날짜를 분석해서 Date 객체로 변환
	parseNumber	문자열로 표시된 날짜를 분석해서 숫자로 변환
	setTimeZone	시간대 정보를 특정 변수에 저장
	timeZone	시간대를 지정

국제화 라이브러리

❖ 숫자 및 날짜 포매팅 태그

formatNumber 태그

- 숫자를 포매팅

```
<fmt:formatNumber value="숫자값" [type="값타입"] [pattern="패턴"]  
[currentCode="통화코드"] [currencySymbol="통화심볼"]  
[groupingUsed="(true|false)"] [var="변수명"] [scope="영역"] />
```

- 주요 속성

속성	표현식/EL	타입	설명
value	사용 가능	String 또는 Number	양식에 맞춰 출력할 숫자
type	사용 가능	String	어떤 양식으로 출력할지를 정한다. <u>number</u> 는 숫자형식, <u>percent</u> 는 % 형식, <u>currency</u> 는 통화형식으로 출력. 기본 값은 number.
pattern	사용 가능	String	직접 숫자가 출력되는 양식을 지정한다. <u>DecimalFormat</u> 클래스에서 정의되어 있는 패턴 사용
<u>var</u>	사용 불가	String	포매팅 한 결과를 저장할 변수 명. <u>var</u> 속성을 사용하지 않으면 결과가 곧바로 출력.
scope	사용 불가	String	변수를 저장할 영역. 기본 값은 page 이다.

국제화 라이브러리

❖ 숫자 및 날짜 포매팅 태그

formatDate 태그

- 날짜 정보를 담은 객체(Date)를 포매팅

```
<fmt:formatDate value="날짜값"  
  [type="타입"] [dateStyle="날짜스타일"] [timeStyle="시간스타일"]  
  [pattern="패턴"] [timeZone="타임존"]  
  [var="변수명"] [scope="영역"] />
```

- 주요 속성

속성	표현식/EL	타입	설명
value	사용 가능	java.util.Date	포매팅할 날짜 및 시간 값
type	사용 가능	String	날짜, 시간 또는 둘 다 <u>포매팅</u> 할 지의 여부를 지정
dateStyle	사용 가능	String	날짜에 대한 <u>포매팅</u> 스타일을 지정
timeStyle	사용 가능	String	시간에 대한 <u>포매팅</u> 스타일을 지정
pattern	사용 가능	String	직접 <u>파싱</u> 할 때 사용할 양식을 지정
<u>var</u>	사용 불가	String	<u>파싱</u> 한 결과를 저장할 변수 이름을 지정
scope	사용 불가	String	변수를 저장할 영역을 지정

국제화 라이브러리

❖ 숫자 및 날짜 포매팅 태그

timeZone과 setTimeZone

- 국제화 태그가 사용할 시간대 설정

```
<fmt:timeZone value="Hongkong">  
  <!-- 사용하는 시간을 Hongkong 시간대에 맞춘다. -->  
  <fmt:formatDate ... />  
</fmt:timeZone>
```

함수 라이브러리

태그	
<code>contains()</code>	검색 대상 문자열의 포함 여부를 확인
<code>containsIgnoreCase()</code>	대,소문자와 상관없이 검색대상문자열 포함 여부 확인
<code>startsWith()</code>	특정 문자열로 시작하는지 확인
<code>endsWith</code>	특정 문자열로 끝나는지 여부 확인
<code>escapeXml()</code>	문자열에 포함된 특수문자를 특정 코드로 변환
<code>indexOf()</code>	검색대상문자열의 첫위치값 반환
<code>split()</code>	문자열을 설정한 구분자로 분리하여 배열 형태로 반환
<code>join()</code>	배열 형태의 문자열을 설정한 구분자로 연결하여 반환
<code>length()</code>	문자열의 길이 반환
<code>substring()</code>	특정 위치의 문자열을 반환
<code>substringAfter()</code>	설정한 문자열 이후의 부분에 있는 문자열 반환
<code>substringBefore()</code>	설정한 문자열 이전의 부분에 문자열 반환
<code>replace()</code>	검색 대상 문자열을 설정한 문자열로 변경하여 반환
<code>toLowerCase()</code>	소문자로 변환
<code>toUpperCase()</code>	대문자로 변환
<code>trim()</code>	문자열 앞위의 공백 제거하여 반환