



템플릿 사용하기

Table of Contents

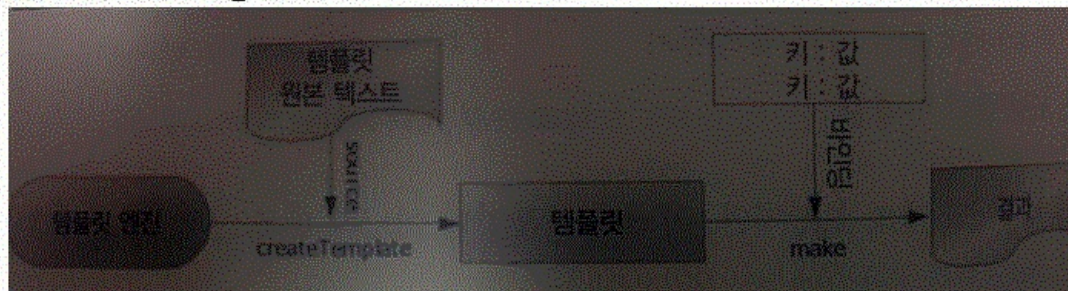
- 템플릿 사용하기
- 그루블릿 사용하기
- 그루블릿 바인딩335
 - 그루블릿에서 사용할 수 있는 정보들
- 그루블릿 템플릿

템플릿의 형식(format)은 jsp문법 jspEL, 등등 에서 아이디어를 얻어 만들어졌다. 개념은 텍스트에 위치 지정자를 넣는것이다.

지정자	용도
<code>\${variable}</code>	변수 <code>variable</code> 의 값을넣는다
<code>\${groovycode}</code>	한줄짜리 그루비 코드를 실행하고 결과를 넣는다
<code><%=groovycode%></code>	그루비 코드 (groovycode) 를 실행하고 결과를 텍스트에 넣는다
<code><%groovycode%></code>	그루비 코드 (groovycode) 를 실행한다.

groovy.text 패키지에는 템플릿 엔진 등이 정의되어있다.

createTemplate 이라는 메서드가 있는데 이메서드는 String, Reader, File, URL 등에서 텍스트를 입력받아 템플릿 (Template) 객체를 생성한다.



템플릿 예제 그저 놀라울따름이다. 입이 벌어진다.

```

1 package _C14
2
3 //##1
4 //mailReminder = ""
5 //Dear ${salutation?salutation+' ':''}$lastname,
6 //another month has passed and again it's time for these
7 //<%=tasks.size()%> tasks:
8 //<% tasks.each { %>- $it
9 //<% } %>
10 //your collaboration is very much appreciated
11 //""
12 //앞에서 나온 out변수는 Writer클래스의 객체이며 디폴트로 바인딩된다.
13 mailReminder = ""
14 Dear ${salutation?salutation+' ':''}$lastname,
15 another month has passed and again it's time for these
16 <%=tasks.size()%> tasks:
17 <% tasks.each { out.println('-'+it) } %>
18 your collaboration is very much appreciated
19 ""
20
21 //def engine = new groovy.text.SimpleTemplateEngine()
22 def engine = new groovy.text.SimpleTemplateEngine(true) //true를 하면 내부동작 정보 출력할수있다.
23 def template = engine.createTemplate(mailReminder)
24 def binding = [
25     salutation: 'Mrs.', //##2
26     lastname: 'Davis', //##2
27     tasks: ['visit the Groovy in Action (GinA) page', //##2
28             'chat with GinA readers'] //##2
29 ]
30
31 //##3
32 println template.make(binding).toString();
33
34 /*
35 결과
36
37 Dear Mrs. Davis,
38 another month has passed and again it's time for these
39 2 tasks:
  
```

```

40 - visit the Groovy in Action (GinA) page
41 - chat with GinA readers
42
43 your collaboration is very much appreciated
44 */

```

템플릿을 사용할 때 주의사항

1. 작은따옴표를 이용해 문자열을 만들어서 쌍따옴표를 쓰면 GString으로 인식한다. GString이 되면 make호출할때가 아니라 createTemplate를 할 때 \$와 \${}의 치환이 이루어진다.
2. 템플릿에는 치환 문자가 정의되어있지 않다. 아주 드물지만 텍스트 글자를 그대로 %> 포함시켜야 하는 경우에는 변수에 '%>'을 넣고 바인딩으로 전달하는것이다. 그리고 \$variable을통해서 출력하면된다.

groovy.text 패키지에는 템플릿 엔진이 세개가 있다 모두 동일한 위치지정자를 지원하지만 특징이 다르다

1. SimpleTemplateEngine은 앞서 본 방식으로 동작한다 make가 호출되는 시점에 스크립트가 한줄씩 출력을 내보낸다 스크립트는 캐시에 보관된다
2. GStringTemplateEngine은 템플릿을 쓰기 가능한 클로저를 다룬다 , 커다란 템플릿이나 상태를 저장하지 않는 스트리밍 에 적용하는 거용에는 속도가 확장성이 더 좋을수 있다.
3. XmlTemplateEngine은 템플릿의 소스와 결과물이 모두 적합한 XML일때로 최적화 되어있다 DOM의 노드들을 기반으로 동작하기 때문에 결과물이 깔끔하다 다른 엔진들과 달리 이엔진은 시스템에 따라 다른 개행 문자를 출력한다,

그루블릿 사용하기

web.xml

```

1 <!DOCTYPE web-app
2 PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.2//EN"
3 "http://java.sun.com/j2ee/dtds/web-app_2_2.dtd" >
4
5 <web-app>
6   <display-name>Groovlet Demonstrator</display-name>
7   <description>
8     Showing the use of Groovlets for Groovy in Action
9   </description>
10
11   <servlet>
12     <servlet-name>Groovy</servlet-name>
13     <servlet-class>groovy.servlet.GroovyServlet</servlet-class>
14   </servlet>
15
16   <servlet-mapping>
17     <servlet-name>Groovy</servlet-name>
18     <url-pattern>*</url-pattern>
19   </servlet-mapping>
20
21 </web-app>

```

myGroovlets/Start.groovy

```

1 html.html{
2   head {
3     title 'Groovlet Demonstrator'
4   }
5   body { h1 'Welcome to the World of Groovlets' }
6 }

```

http://localhost:8080/myGroovlets/Start.groovy 결과확인가능
그루비 특성상 소스를 수정후 서버를 재기동할필요가 없다.

그루블릿 바인딩335

다른 그루비 스크립트처럼 그루블릿에도 정보가 들어 있는 바인딩(binding)이있고 binding이라는 프로퍼트를 통해 얻을수 있다.

Inspect.groovy

```

1 html.html{
2   head {
3     title 'Groovlet Demonstrator'
4   }
5   body {
6     h1 'Variables in the Binding:'
7     table(summary:'binding') {
8       tbody {
9         binding.variables.each { key, value ->
10           tr {
11             td key.toString()
12             td(value ? value.toString() : 'null')
13           } } } } } }

```

그루블릿에서 사용할 수 있는 정보들

이름	내용	용례
headers	HTTP 요청 헤더맵	headers.host
params	HTTP 요청인자맵	params.myParam
session	ServletSession null인경우도있음	session?.myParam
request	HttpServletResponse	response.contentType='text/xml'
context	ServletContext	context.myParam
application	ServletContext(context동일)	application.myParam
out	response.writer	바인딩에 들어가는 시점에는 초기화 되지 않는다 나중에 초기화됨
sout	response.outputStram	바인딩에 들어가는 시점에는 초기화 되지 않는다 나중에 초기화됨
html	new MarkupBuilder(out)로 초기화된 빌더	바인딩에 들어가는 시점에는 초기화 되지않는다 나중에 초기화된다

그루블릿 템플릿

그루블릿에 바인딩되어 있는 out변수를 이용하면 응답 객체 response object에 직접출력할수있다.

Number.template.html

```

1  /*
2  out<<'<html>'
3  ...
4  out<<'</html>'
5  또는 다음처럼 오늘날짜 시각을 GString으로 출력가능하다
6  out << "<HTML><BODY> ${new Date.toGMTString()}</BODY></HTML>";
7  */
8  <html>
9  <head>
10 <title>Think of a Number</title>
11 </head>
12 <body>
13 <h1>Think of a Number</h1>
14 Your guess $guess is <%
15     switch (guess) {
16         case (it == goal): out << 'correct!'; break
17         case (it < goal) : out << 'too low'; break
18         case (it > goal) : out << 'too high'; break
19     }
20 %>
21 <p>What"s your guess (0..100)? </p>
22 <form action="Templater.groovy" method="GET">
23     <input type="text" name="guess">
24     <button>Guess</button>
25     <button name="restart" value="true">New Game</button>
26 </form>
27 </body>
28 </html>
```

```

1  def engine = new groovy.text.SimpleTemplateEngine()
2  def source = getClass().classLoader.
3              getResource('/Number.template.html')
4  def template = engine.createTemplate(source)
5
6  out << template.make(goal:50, guess:49)
```

guess변수를 출력하는 GString도있고 사이에 그루비 코드도 있다.

올바르게 해석되면 GString위치지정자는 변수로 치환되고 삽입된코드는 실행된다.

이런방식에서 주의할 사항은 그루블릿이 템플릿 소스를 읽어 들이기 위해서는 템플릿 소스가 클래스 패스상의 리소스이어야 한다는 점이다 그렇지 않으면 읽어 들일 파일의 위치를 알수 없다. 따라서 웹어플리케이션의 classes 디렉터리에 있다.

그루비는 웹어플리케이션에서 제어 로직보다 템플릿 기능에 무게를 두는 사람을 위해 완전한 JSP와 같은 기능으 ㄹ제공하는 GSP (Groovy Server Pages)도 지원한다.

또한 그루비와 함께 제공되는 특별한 그루블릿인 TemplateServlet을 컨트롤러로 사용할수 있다 web.xml에 다음과 같이 설정하면된다.

```
1 <!DOCTYPE web-app
2 PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.2//EN"
3 "http://java.sun.com/j2ee/dtds/web-app_2_2.dtd" >
4
5 <web-app>
6   <display-name>Groovlet Demonstrator</display-name>
7   <description>
8     Showing the use of Groovlets for Groovy in Action
9   </description>
10
11   <servlet>
12     <servlet-name>Groovy</servlet-name>
13     <servlet-class>groovy.servlet.TemplateServlet</servlet-class>
14   </servlet>
15
16   <servlet-mapping>
17     <servlet-name>Groovy</servlet-name>
18     <url-pattern>*.html</url-pattern>
19   </servlet-mapping>
20
21 </web-app>
```

