

Freemarker Decorators

As of SiteMesh 2.0.2 [Freemarker](#) (.ftl) decorators are supported.

Here is an example of how such a decorator might look like:

```
<#include "/includes/decorators/header.dec">
  <h2>${title}</h2>
  ${head}
  
  <td valign="top" class="body">
    <div class="header">
      <span class="pagetitle">${title}</span>
    </div>
    ${body}
  </td>
<#include "/includes/decorators/footer.dec">
```

Installation

Download [Freemarker](#) 2.3 (recommended) and copy it into **WEB-INF/lib**. The SiteMesh distribution comes with **freemarker.jar** v2.3rc3.

Add the following to **WEB-INF/web.xml** within the `<web-app>` tag:

```
<servlet>
  <servlet-name>sitemesh-freemarker</servlet-name>

  <servlet-class>com.opensymphony.module.sitemesh.freemarker.FreemarkerDecoratorServlet<
/servlet-class>
  <init-param>
    <param-name>TemplatePath</param-name>
    <param-value>/</param-value>
  </init-param>
  <init-param>
    <param-name>default_encoding</param-name>
    <param-value>ISO-8859-1</param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
</servlet>

<servlet-mapping>
  <servlet-name>sitemesh-freemarker</servlet-name>
  <url-pattern>*.dec</url-pattern>
</servlet-mapping>
```

Modify **decorators.xml** to reference a .dec file.

```

<#include "/includes/decorators/header.dec">
  <h2>${title}</h2>
  ${head}
  
  <td valign="top" class="body">
    <div class="header">
      <span class="pagetitle">${title}</span>
    </div>
    ${body}
  </td>
<#include "/includes/decorators/footer.dec">

```

Context

FreemarkerDecoratorServlet puts some things into the context object that you should be aware of:

Basic context attributes

It makes all request, request parameters, session, and servlet context attributes available to templates through Request, RequestParameters, Session, and Application variables. For example :

```

${Session["user"]}

```

The scope variables are also available via automatic scope discovery. That is, writing Application.attrName, Session.attrName, Request.attrName is not mandatory; it's enough to write attrName, and if no such variable was created in the template, it will search the variable in Request, and then in Session, and finally in Application.

```

<#assign ww=JspTaglibs["/WEB-INF/webwork.tld"]>

```

It creates a variable with name JspTaglibs, that can be used to load JSP taglibs. For example:

```

<#assign ww=JspTaglibs["/WEB-INF/webwork.tld"]>
...
<@ww.property value="myVar" />

```

Sitemesh context attributes

base	request.getContextPath()
title	Parsed page title (<title>...</title>)
head	Parsed page head
body	Parsed page body
page	SiteMesh's internal Page object