SiteMesh Adept in 10 Minutes

Introduction

If you have gotten this far you are on the **last** tutorial required to **effectively** use SiteMesh for your own website. As a review it took,

- 5 Minutes to Setup SiteMesh
- 10 Minutes to Start Using SiteMesh

This tutorial will take 10 minutes and build on the Café Mirabeau example to,

- Show how decorators reference external files such as images.
- Setup excludes to ignore common resources that should not be decorated.

After this you can consider yourself a SiteMesh Adept and using SiteMesh for real world websites.

Reference External Files in the Decorator

Building upon Start Using SiteMesh we will add an image to the decorator file **basic-theme.jsp**.

First download this image taken from the Darpuke Template or use your own image file and place it in the **decorators** folder.

Full Path Link

If using pure html, to reference the image, basic-them.jsp would be updated with an html img tag look like this,

```
<img src="/cafemirabeau/decorators/logo.jpg" alt="logo" />
```

It is not obvious, but the working directory for the page being loaded is **not** decorators. When hours.jsp or menu.jsp is being loaded the working directory is WebContent/**data**.

As such, full paths for **must** be used when referring to external files in **decorator** files.

However, if you ever decide to change your context root (that is **/cafemirabeau**) you must update your decorator files accordingly. Luckily the idea of using full paths for global resources is common and there are a number of options.

Expression Language to Generate Context Path

As of version 2.0 JSP technology now includes support for EL (Expression Language) which dynamically generates the context root.

Download or update your basic-theme.jsp to use EL as shown,

```
<?xml version="1.0" encoding="UTF-8" ?>
<%@ taglib uri="http://www.opensymphony.com/sitemesh/decorator" prefix="decorator" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title></title>
</head>
<body>
    <h1>Header8</h1>
    <p><b>Navigation</b></p>
    <img src="${pageContext.request.contextPath}/decorators/logo.jpg" alt="logo" />
    <hr />
    <decorator:body />
    <hr />
    <h1>Footer</h1>
</body></html>
```

Line 11 uses the EL variable to generate the context root based on the web application.

Verify Your Changes

Reload either menu.jsp or hours.jsp page and ensure the image still loads.

Exclude

The last critical concept to understand is the use of **exclude**.

Certain files such as images or JavaScript should never be decorated. Also, there may be a need to not decorate certain folders.

This can be accomplished by defining exclude patterns in **decorators.xml**.

⚠ Any changes made to decorators.xml will require a server restart.

Patterns can be file based, folder based or a combination of the two. Here is a sample specifically to show the flexibility,

```
<?xml version="1.0" encoding="UTF-8"?>
<decorators defaultdir="/decorators">

  <!-- Excludes will never be decorated by Sitemesh. -->
  <excludes>
    <pattern>*/exclude/*</pattern>
    <pattern>/data/plaintext/*</pattern>
    <pattern>*.txt</pattern>
  </excludes>

  <decorator name="basic-theme" page="basic-theme.jsp">
    <pattern>/data/*</pattern>
  </decorator>
</decorators>
```

Line 6 - Any resources inside a folder matching the name **exclude** will not be decorated.

Line 7 - The folder from the context root **/data/paintext** will not be decorated.

Line 8 - Any file with the txt extension will not be decorated.

Start Meshing

This is all you need to get going so start meshing away! Really that is it.

Once you implemented SiteMesh, you might want to come back and look at the SiteMesh Bells and Whistles tutorials and eventually move to Advanced SiteMesh.