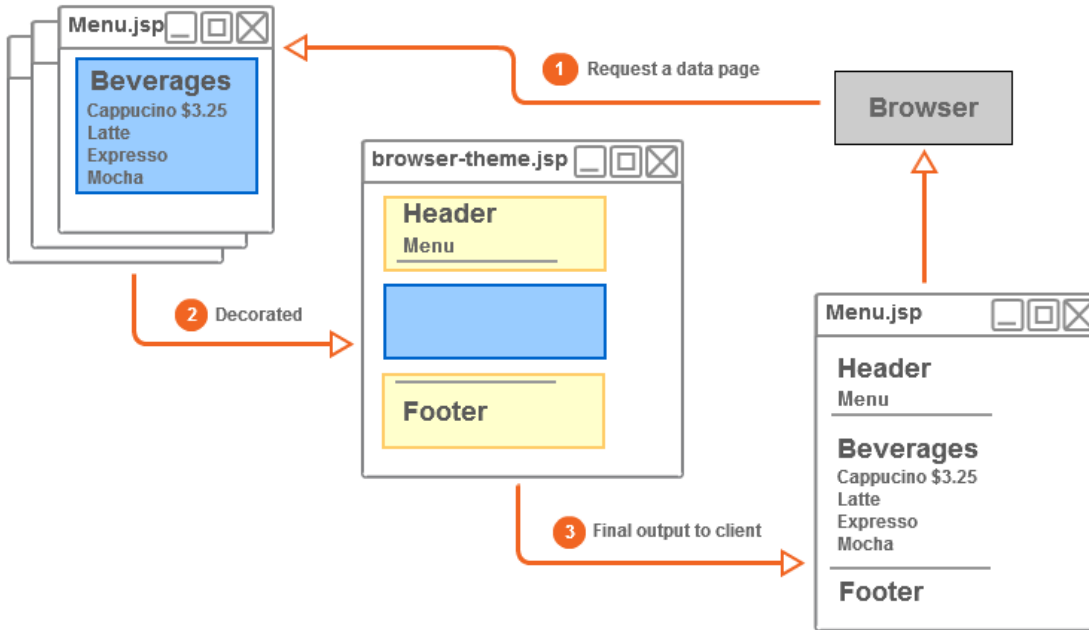


Start Using SiteMesh in 10 Minutes

Introduction

The best approach to understanding SiteMesh is to **use** it. Assuming that SiteMesh is setup in your web application this tutorial will show you how to master the most powerful aspect of SiteMesh, decorating pages as illustrated below,



The magic happens at step 2 where the page `Menu.jsp` is rendered as html. Just before the html page is sent down to the client browser, the page is decorated by a **single** file, `basic-theme.jsp`.

In this example, a menu is added, a footer is added **without any extra code** being added to `Menu.jsp`.

Demo Website

We will start by creating a simple website about a fictional coffee & cake shop called Café Mirabeau.

In your web folder, usually called **WebContent**, create a folder called **data**. Inside of **data** create or download (`menu.jsp` `hours.jsp`) these 2 simple html pages.

menu.jsp

```
<?xml version="1.0" encoding="UTF-8" ?>
<!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>Menu</title>
</head>
<body>
  <h1>Beverages</h1>
  <p>Cappucino $3.25</p>
  <p>Latte $3.35</p>
  <p>Espresso $2.00</p>
  <p>Mocha $3.50</p>
</body>
</html>
```

And your second page,

hours.jsp

```
<?xml version="1.0" encoding="UTF-8" ?>
<!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>Hours</title>
</head>
<body>
  <h1>Weekdays</h1>
  <p>5:00pm - 10:00pm</p>
  <p>Weekends</p>
  <p>5:00pm - 10:00pm</p>
</body>
</html>
```

The key thing to keep in mind is that these are just simple HTML pages which all contain a head and body. The files have a jsp extension to prevent caching issues.

Setup Decorators

The decorators are what define the look and feel of the site.

In your web folder, usually called **WebContent**, create a folder called **decorators**. Inside **decorators** create or download your first decorator [basic-theme.jsp](#),

basic-theme.jsp

```
<?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>Header</h1>
  <p><b>Navigation</b></p>
  <hr />
  <decorator:body />
  <hr />
  <h1>Footer</b></h1>
</body>
</html>
```

Line 2 - the **taglib** declaration allows the JSP file to use the SiteMesh tag library.

Line 12 - **decorator:body** places the rendered contents of the decorated pages into the decorator.

Define Decorator and Pattern

The last step is to modify or download **WebContent/WEB-INF/decorators.xml** to,

- Reference the decorator **basic-theme.jsp**
- Specify the what resources should be decorated

decorator.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<decorators defaultdir="/decorators">
  <decorator name="basic-theme" page="basic-theme.jsp">
    <pattern>/data/*</pattern>
  </decorator>
</decorators>
```



Because **decorators.xml** was modified, the server must be restarted for the changes to take effect.

Start your application server and browse to **menu.jsp** and **hours.jsp** to see the results of the pages combined with the decorator.



Not familiar with Java, JSP or Web Applications? Download the [SiteMesh Starter Kit](#) and follow the instructions on how to [Launch the Default Example](#).

Summary

Using SiteMesh you can separate the look and feel from the data. Building on this example, there are more possibilities. For example additional templates could be loaded for,

- Mobile Devices
- Printable Web Pages
- Web Accessibility

In large team this also allows easy enforcement of roles and responsibilities between presentation layer and back-end developers.

Even More!

SiteMesh provides an easy mechanism for page templates and has a wide range of uses beyond what this example shows. Some [feature](#) highlights include,

- Technology agnostic decoration for example, HTML, PHP, Perl etcetera
- Logic based decorating
- Parameter passing
- Integration with other frameworks (Freemarker, WebWorks, Spring, Struts, Velocity etcetera)

If you want to dive right in, read the 10 minute tutorial, [SiteMesh Adept](#), which provides all the necessary leanings to effectively use SiteMesh on your own website.