

Tapestry5Layoutcomponent

Making a component Layout with Tapestry 5 is really simple.

An example is given at: <http://tapestry.apache.org/tapestry5/guide/layout.html> in the official tapestry 5 documentation but I propose a more sophisticated component with content fallback, able for instance to override its left column content if needed.

We'll finish with something that will look like a bit like the Tiles Template system (Struts), but much much simpler than Tiles.

In the package components, create a Layout component and template

Layout.java :

```
import org.apache.tapestry.annotations.Parameter;
import org.apache.tapestry.ioc.annotations.Inject;
import org.apache.tapestry.services.Request;

/**
 * @author Michael Courcy
 *
 * This component aims to provide layout service for a page
 */
@IncludeStylesheet("context:layout.css")
public class Layout {

    @Parameter
    private String title = "My Tapestry powered site";

    @Inject
    private Request request;

    public String getTitle() {
        return title;
    }

    public void setTitle(String title) {
        this.title = title;
    }

    public Request getRequest() {
        return request;
    }

    public void setRequest(Request request) {
        this.request = request;
    }
}
```

Layout.tml

```
<html xmlns:t="http://tapestry.apache.org/schema/tapestry_5_0_0.xsd">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>${title}</title>
</head>
<body>

  <table id="main">
    <tr>
      <td id="header" colspan="2">
        The header
      </td>
    </tr>
    <tr>
      <td id="left">
        Here come the left content
      </td>
      <td id="content">
        <t:body/>
      </td>
    </tr>
    <tr>
      <td id="footer" colspan="2">
        Tapestry powered Site
      </td>
    </tr>
  </table>

</body>
</html>
```

Then add this css file in the root context of your webapp.

layout.css

```

table#main {
width: 700px;
margin-left:auto;
margin-right:auto;
}

td#header {
height: 120x;
border: 1px solid blue;
padding: 10px;
margin: 10px;
}

td#footer {
height: 120x;
border: 1px solid blue;
padding: 10px;
margin: 10px;
}

td#left{
width: 200px;
border: 1px solid red;
padding: 10px;
margin: 10px;
}

td#content{
border: 1px solid red;
padding: 10px;
margin: 10px;
}

```

Let's try it! In the package page, create a page [TestLayout.java](#) which does ot do anything as its only purpose is to test the component.

```

public class TestLayout {

}

```

and in the root context of your webapp, create the template [TestLayout.tml](#) :

```

<t:layout title="literal:My first use of layout component" xmlns:t="http://tapestry.apache.org/schema/tapestry_5_0_0.xsd">
My content
</t:layout>

```

You should get this:

The header	
Here come the left content	My content
Tapestry powered Site	

Note that the title of the page is passed as a parameter.

But could we regard this solution as fulfilling our requirements?

Not really, because the content of the left column is now static and we're about 100% sure that for some pages we'll have to override it.

Let's change the component using Delegate, Block, Parameter and a component of our own : Left.

Left.java (in the components package)

```
/**
 * @author Michael Courcy
 *
 * This component is just a content holder to fill the left column of the layout component
 *
 */
public class Left {

}
```

Really simple, isn't it? That's barely a container.

Now let's change Layout.tml.

```
<html xmlns:t="http://tapestry.apache.org/schema/tapestry_5_0_0.xsd">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>${title}</title>
<link href="${request.contextPath}/layout.css" rel="stylesheet" type="text/css" />
</head>
<body>

  <table id="main">
    <tr>
      <td id="header" colspan="2">
        The header
      </td>
    </tr>
    <tr>
      <td id="left">
        <t:delegate to="leftContent" />
        <t:block>
          <div t:id="leftRegularContent">
            Here come the left regular content
          </div>
        </t:block>
      </td>
      <td id="content">
        <t:body/>
      </td>
    </tr>
    <tr>
      <td id="footer" colspan="2">
        Tapestry powered Site
      </td>
    </tr>
  </table>

</body>
</html>
```

What you have to focus on is the change done in the left column, the content of the component leftRegularContent will be a "content fallback".

You have to change Layout.java as well

```

package org.apache.tutorial.tapestrySpringHibernate.components;

import org.apache.tapestry.Block;
import org.apache.tapestry.annotations.Component;
import org.apache.tapestry.annotations.Parameter;
import org.apache.tapestry.ioc.annotations.Inject;
import org.apache.tapestry.services.Request;

/**
 * @author Michael Courcy
 *
 * This component aims to create a layout around the main content
 * You can optionnaly defines left content that overrides the regular left
 * content if needed through the <t:parameter name="left">some overiding left content</t:parameter>
 */
public class Layout {

    @Parameter
    private String title = "My Tapestry powered site";

    @Parameter
    private Block left;

    @Component
    private Left leftRegularContent;

    @Inject
    private Request request;

    /**
     * This method check if the left parameter has been set by the user
     * if not the regular content is shown otherwise the content of this parameter
     *
     * @return the component we want to display
     */
    public Object getLeftContent(){
        return left==null ? leftRegularContent : left;
    }

    public String getTitle() {
        return title;
    }

    public void setTitle(String title) {
        this.title = title;
    }

    public Request getRequest() {
        return request;
    }

    public void setRequest(Request request) {
        this.request = request;
    }
}

```

Et voila !! Without defining a left parameter :

```

<t:layout title="literal:My first use of layout component" xmlns:t="http://tapestry.apache.org/schema
/tapestry_5_0_0.xsd">
My content
</t:layout>

```

you get this:

The header	
Here come the left regular content	My content
Tapestry powered Site	

With a left parameter defined :

```
<t:layout title="literal:My first use of layout component" xmlns:t="http://tapestry.apache.org/schema/tapestry_5_0_0.xsd">
  My content
  <t:parameter name="left">
    Other left content
  </t:parameter>
</t:layout>
```

You get this:

The header	
Other left content	My content
Tapestry powered Site	

(originally posted on Michael Courcy's blog - the images only work if you right click and view image, if anyone can fix this or can host the images where they'll work inline - please do so)

NOTE: Inline images have now been fixed (11/23/2008)