

Tapestry5SimpleAndPowerfulLayouts

Here, we are going to create a simple and reusable component, which makes it extremely easy to create layout components.

First, we create a "Slot" component. It is used as a placeholder in our layout components.

Slot.java :

```
package org.myorg.t5demo.components;

import org.apache.tapestry5.BindingConstants;
import org.apache.tapestry5.Block;
import org.apache.tapestry5.ComponentResources;
import org.apache.tapestry5.annotations.Parameter;
import org.apache.tapestry5.ioc.annotations.Inject;

public class Slot {

    @Inject
    private ComponentResources resources;
    @Parameter(name = "id", defaultPrefix = BindingConstants.LITERAL, required = true)
    private String id;

    Object beginRender() {
        ComponentResources res = resources.getContainerResources();
        Block toRender = null;
        while (res != null) {
            Block temp = res.findBlock(id);
            if (temp != null) {
                toRender = temp;
            }
            res = res.getContainerResources();
        }
        return toRender;
    }
}
```

Now, we create a layout component,

Layout.tml

```
<html xmlns:t="http://tapestry.apache.org/schema/tapestry_5_0_0.xsd">
  <head></head>
  <body>
    <h1>DefaultLayout</h1>
    <t:Slot id="part1"></t:slot>
    <t:Slot id="part2"></t:slot>
  </body>
</html>
```

Layout.java :

```
org.myorg.t5demo.components;

public class Layout {
}
```

and use it to create a page.

Index.tml

```
<html t:type="Layout" xmlns:t="http://tapestry.apache.org/schema/tapestry_5_0_0.xsd">
  <t:block id="part2">
    <h1>${part2}</h1>
  </t:block>
</html>
```

index.java :

```
package org.myorg.t5demo.pages;

public class Index {

    public String getPart2() {
        return "Part 2";
    }
}
```

Now, try it yourself.

Note that we do not have a single line of java code to handle layouting logic in the layout component. And also note that we do not have to add a block in the page for each slot defined in the layout component.

Sublayout can be easily created, providing defaults for slots defined in upper layouts, or refining the layouts by introducing new slots.