Border

Scope

ComponentBackground | componentJavadoc=Border

Background

A Border is a reusable component that holds markup and other components. Is special type of Panel with the ability to wrap code with a header and a footer specified by the Border component.

You can use Markup inheritance with Borders too.

See Panels and borders to see the difference between Panels and Borders and Fragment to see how to use 'in-line' panels.

Borders can be used to create dynamic markup hierarchies.

FAQ

Examples

Typical usage

BaseBorder.java:

```
public class BaseBorder extends Border {
  public BaseBorder (String id, User user) {
    super(id);
    add(new Label("username", user.getUsername()));
    add(new Label("lastLogin", user.getLastLogin()));
  }
}
```

BaseBorder.html:

```
<wicket:border>

  <h1>Header</h1>
  User <span wicket:id="username">dummy name</span> (last log-in <span wicket:id="lastLogin">2037-09-10 10:
21:59h</span>)

  <wicket:body/>

  Footer with copy right notice, disclaimer etc.

</wicket:border>
```

That's for your reusable panel, which you can now use anywhere like:

```
add(new BaseBorder ("myBorder", user);
```

and:

```
<span wicket:id = "myBorder">
<h2>Welcome</h2>
This is just a place holder for page content ...
</span>
```

this will generate the following HTML output:

```
<h1>Header</h1>
User dummy name (last log-in 2037-09-10 10:21:59h)

<h2>Welcome</h2>
This is just a place holder for page content ...

Footer with copy right notice, disclaimer etc.
```

In other words, the body of the myBorder component is substituted into the border's associated markup at the position indicated by the <wicket:body> tag.

Regarding <wicket:body/> you have two options. Either use <wicket:body/> (open-close tag) which will automatically be expanded to <wicket:body>body content</wicket:body> or use <wicket:body>preview region</wicket:body> in your border's markup. The preview region (everything in between the open and close tag) will automatically be removed.