ClearingPersistentProperties

NOTE: This is outdated information that applies only to Tapestry 4. For current information see Persistent Page Data.

Tapestry has excellent support for the inclusion of persistent properties in both Pages and Components. Persistent properties map directly to storing the content into the HttpSession that is bound to the parent page. Its desirable to clear the persistent properties for two reasons.

Firstly, using persistent properties on many pages can result in the server memory storage requirements growing unacceptably thus limiting scalability of the application. Secondly, there is a risk of left over state incorrectly being utilised therefore causing undesired behaviour. Thus often you would want to clear the persistent properties once you have finished using a page.

The tapestry engine (IEngine) manages the resources and thus provides the mechanism to clear the persistent properties. Normally when an action has been triggered and a listener is performing its task, the last step is to cycle to the next page. Just before cycling to the page call the following method.

```
                this.getEngine().forgetPage(this.getPageName());
```

This will clear the page state thus page properties are removed. This method works fine from direct link listeners though there is one situation that this does not work. Internally, the form listener uses the a page recorder to store and restore the persistent properties. If you attempt to forget the page within a form listener an exception will be thrown as the recorder is performing its work. In this case it is necessary to provide a hint for removal so that once the recorder has finished the page state can be removed.

```
                this.getEngine().getPageRecorder(this.getPageName(), cycle).markForDiscard();
```

The only issue is making sure that all exit points are covered with a call to the appropriate method of state removal. In the future (possibly post 3.0) its proposed that tapestry will provide optional automatic life cycle state management so that in many situations this aspect will not need to be managed by the developer.

I started to use a simple

```
        cycle.discardPage(getPage().getPageName());
```

everywhere before cycling to the next page. This also works in forms. I don't know why the API doc says "This is used in certain rare cases...".