SylvainWallez

Thanks for visiting my little corner of the Cocoon wiki! Want to know more about me? Visit my Apache home page and my blog

Cocoon wishlist

Here's a dump of random ideas that pop in my head about Cocoon. Ideas come and go, and good ones sometimes get lost before coming back later. So here they are, both as a personal reminder and for you to pick them for discussion and/or implementation.

Cocoon Forms

Styling as attributes of <ft:widget> - *done*

Having to open a <ft:widget> and add a <fi:styling> for a simple attribute as type="textarea" is cumbersome. In the template language implementation (both jx and transformer), automatically create a <fi:styling> if <ft:widget> has some "foreign" attributes. E.g:

```
<ft:widget id="foo" type="textarea" cols="80" rows="30"/>
```

Will be equivalent to:

```
<ft:widget id="foo">
  <fi:styling type="textarea" cols="80" rows="30"/>
</ft:widget>
```

Automatic <ft:continuation-id>

Insertion of the continuation id as an input field can be automated if the <ft:form-template>'s action attribute doesn't contain the continuation id.

Easier access to resources in jar - *done*

CForms and the BrowserUpdateTransformer (which is available in the core) come with client-side resources (js, css, gifs, etc) stored in jars. To ease access to these resources, define a naming convention allowing a generic pipeline in the root sitemap:

```
<map:match pattern="system/*/**">
  <map:read src="resource://org/apache/cocoon/{1}/resources/{2}"/>
</map:match>
```

Maybe also use the resource-exists action to allow local override in context://resources/{1}/{2}.

Add a SelectionList.contains()

Several people have asked how to check if a selection-list contains a particular value.

Non-exclusive selection-lists (comboboxes)

Add an attribute to <fd:selection-list> ("exclusive", "open", "strict" or "closed"?) indicating if this is a strict list or not (default is strict).

This will allow for combo-boxes, where the user can choose an existing value but also give a new one.

"Cocoon suggests" styling - *done*

Non-exclusive selection-lists also allow "Google suggests"-type of styling. Selection-lists must therefore be "filterable", i.e. provide a limited list given the current user input. For URL-based selection-lists, the current input can be passed as an query parameter.

Add the notion of selection to repeaters

See this post.

Autosave forms

Suggested by Ugo: allow autosaving of forms at a regular interval. This is not about posting the form, as if you loose the continuation you loose the form also. Add a "autosave-listener" to the form object?

Error handling

Define a LocatableException *done*

Cocoon uses a lot of exception cascading. Having a LocatableException can allow to easily provide a "cocoon stack trace" by gathering all locations of LocatableExceptions in the exception chain.

Use Locator and SAXParseException

The Locator object is available everywhere in pipeline components and often tracks the location of the generator's source file. It can therefore be used in the processing chain when an error is found to provide the user with location information. The SAXParseException extends SAXException with a Locator and is therefore just what we need to provide better error messages.