

The Portal Basket (2.1 legacy document)

Table of contents

1 Comments.....4

Table of contents

1 Overview.....	3
2 General Configuration.....	3
3 Coplets as Basket Content.....	3
4 URLs as Basket Content.....	3
5 Viewing the content of the basket.....	4
6 Persistence.....	4
7 Viewing items.....	4

Warning:

This document was copied as is from the Cocoon 2.1 documentation, but has not yet been fully reviewed or moved to its new home.

1. Overview

This document describes the basket portlet. This is a special functionality that can be used inside your portal application to implement a per user basket.

The sample portal that comes with the Cocoon distribution contains a working sample for the basket. So, you can test it out of the box and skip the next chapter.

If you don't need the basket functionality have a look at the end of the next chapter to see what you can remove from the portal.

2. General Configuration

Each coplet that is able to be added to the basket, gets a button "add" in the window title. For rendering this title, you need to add the *basket* renderer to the render that is rendering your window. Add the following aspect to your *window* renderer:

```
<aspect type="basket"/>
```

The basket uses a manager component that manages the persistence of the baskets. You have to add this component to your *cocoon.xconf* and configure the directory that is used to store the baskets accordingly:

```
<component class="org.apache.cocoon.portal.coplets.basket.BasketManager"
role="org.apache.cocoon.portal.coplets.basket.BasketManager"> <parameter
name="directory" value="D:/baskets"/> </component>
```

The default directory is the work directory.

If you need a different persistence strategy, subclass this component and configure it accordingly.

If you don't need the basket functionality, remove the above component definition, remove the aspect renderer usage and configuration.

3. Coplets as Basket Content

The basket allows to add a coplet or the content of a coplet to the basket. If a coplet should be able to be added to the basket, it has to be configured accordingly. Add one of the following two attributes to the coplet to either allow to add the coplet or the content of the coplet to a basket:

```
<attribute> <name>basket-link</name> <value xsi:type="java:java.lang.Boolean"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">true</value> </attribute>
<attribute> <name>basket-content</name> <value xsi:type="java:java.lang.Boolean"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">true</value> </attribute>
```

The attributes have to be added to the coplet data configuration of a coplet.

4. URLs as Basket Content

The basket allows to add a URL or the content of a URL to the basket. You can add this feature to your coplet by using the *basket transformer*. This transformer listens for the *add-item* element and transform this into an html link:

```
<p xmlns:basket="http://apache.org/cocoon/portal/basket/1.0"> <basket:add-item
href="http://cocoon.apache.org">Add</basket:add-item> </p>
```

With the example from above you add the link to a URI to the basket. If you add the attribute *content* with the value *true* to the *add-item* element, not the URI but the content of the URI is added to the basket.

5. Viewing the content of the basket

The *basket* generator streams out the contents of the basket as XML. You can use this generator in your pipeline to render the content of the current user basket.

This generator has also a management functionality buildin. By adding the parameter *admin-mode* with the value *true* to the generator, the generator streams out the admin functions as XML. These functions allow to delete a basket for a user and to see all user baskets.

6. Persistence

Each basket is persisted using the basket manager component. The default installation persists each basket of a user into a separate file in the temporary directory. You can change the location of the directory by configuring the basket manager in the *cocoon.xconf*.

If you need a different persistence strategy, you can subclass the basket manager and for example store the baskets in a database etc.

7. Viewing items

The user is able to select a single item and display the contents in a different coplet. Therefore the *basket generator* needs two additional parameters: the *show-coplet* and the *show-layout* parameter.

The *show-layout* parameter defines a coplet layout element in the portal layout of the user that will display the selected item.

The *show-coplet* parameter defines a coplet data that will be used to create an instance that will display the content.

1. Comments

add your comments