

WoodyCopletForPortalEngine

I found some questions on the user mailing list about how to use woody with the Portal Engine. I have taken some time to reproduce the process with basic samples provided in Cocoon.

To use woody in a coplet portal isn't very difficult but you need a special handling for the continuation. Got a Cocoon-2.1.4-dev, and we gonna put the woody/registration form (because it simple) inside a coplet.

Note: Cocoon-2.1.4-dev also contains a different approach that is documented in the portal documentation (form handling) and the portal demo included in Cocoon contains a demo for this after you login and select the "Apps" tab (Carsten).

Step 1: The first step is to modify one existing coplet to use a woody form. Edit the `portal/layout/copletdata/portal.xml` file and change the value of the uri of the Portal-Demo with `cocoon://samples/woody/registration`.

```
...
<coplet-data id="Portal-demo" name="standard">
  ...
  <value xsi:type="java:java.lang.string"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    cocoon://samples/woody/registration
  </value>
  ...
</coplet>
...
```

Log into the portal as guest/guest and in the main tab instead of the introduction coplet, you will see the woody form.

Now the problem is that the KontId isn't handling. It's always the same uri called for this coplet. The simplest trick will be to put the KontId in an hidden field and change the woody sitemap to handle it.

Step 2: Add the continuation hidden field. Edit the `woody/forms/registration_template.xml` and add after `wt:form-template` the tag

```
<wt:continuation-id/>
```

Then set the form action to an empty string, because we want use the portal page, not just the form. Your file should look like

```
{{{<page ...
<content>
<wt:form-template action="" method="POST">
<wt:continuation-id/>
<wt:widget-label id="name"/>
...</page>
}}}
```

Step 3: Configure the sitemap to handle the continuation. we gonna switch the pipeline based on the request method. Edit the `woody/sitemap.xmap` and replace the match pattern registration with

```
{{{<map:match pattern="registration">
<map:select type="request-method">
<map:when test="GET">
<map:call function="registration"/>
</map:when>
<map:when test="POST">
<map:call continuation="{request-param:continuation-id}"/>
</map:when>
</map:select>
</map:match>
}}}
```

That's all. Right now you can enjoy the woody inside a Coplet.

Optional Steps: The registration form is not design to put itself inside a coplet. You will need to remove the html/head/body part. It will be cleaner.

When you change the menu tab, you can see that you lose the woody context. It's because you lose also the KontId. To solve it you can put it in a coplet attribute instead of the request parameter.

Hopes, it's clear enough

[Laurent Trillaud](#)

New Way: Since February 2004, (Cocoon-2.1.5-dev) there is a new way to use woody in the Portal Engine, because it comes with the application coplet adapter.

Check your [<http://localhost:8888/docs/developing/portal/forms.html>] for documentation.

Therefore to put any woody form (registration for exemple) inside a coplet, you have just to change the `temporary:application-uri` of one coplet with `cocoon://samples/woody/registration` in the `layout/copletdata/portal.xml` file.

It will work fine except for client javascript, that needs to be given to the portal main page.

[Laurent Trillaud](#)

However: your form must specify an `action`, i.e. page that it submits to. The normal behaviour of a form with no `action` is that it submits back to itself, but this won't work with the Application coplet adaptor. This is so that the coplet adaptor can change the value of `action` so it can make sure that the rest of your portal loads around your coplet when you submit your form.

e.g.

```
<form method="post" action="submitpage">
  <!-- [...] -->
</form>
```

[Jon Evans](#)

Beware! your form must be a valid html page in order to be accepted by a coplet. The `forms-samples-styling.xsl` stylesheet of the distribution formats a form in this way:

```
<form ...>
  <!-- [...] -->
</form>
```

You have to modify the stylesheet in order to add the `html` and `head` and `body` tags if you want your form to appear in a coplet.

[Vlad Tanasescu](#)