forrest:views - x output formats, one config

This plugin depends on a ViewHelper and a BusinessHelper implementation.

Table of contents

1. view

Together with the ViewHelper and the BusinessHelper (to be implemented) plugin this plugin is following the DispatcherView pattern and building the final view which will be delivered to the client. This implementation follows exactly the pattern shown on corej2eepatterns.com

A view will be configured by a config file which basically contains the following tags (it is still evolving):

```
<forrest:view type="xhtml">
  <forrest:css url="common.css"/>
  <forrest:contract name="meta"/>
  <forrest:hook name="container">
   <forrest:contract name="feedback"/>
  </forrest:hook />
</forrest:view>
```

The view tag's @type determines the final output format. The idea is to configure different output formats within a forrest:views. That means a forrest:views can contain n different "forrest:view" configurations for n different formats.
In short: x formats, one config. :)

```
<forrest:views
  xmlns:forrest="http://apache.org/forrest/templates/1.0">
<forrest:view type="xhtml">
  <forrest:css url="common.css"/>
  <forrest:contract name="meta"/>
  <forrest:hook name="container">
   <forrest:contract name="feedback"/>
  </forrest:hook />
</forrest:view>
<forrest:view type="fo">
  <forrest:hook name="container">
   <forrest:contract name="feedback"/>
  </forrest:hook />
</forrest:view>
/forrest:views>
```

2. viewHelper

viewHelper class/implementation - a class/implementation which will access the presentation model and transform its models. In other words it is a presentation producing factory.

> **Note:**
> The <forrest:contract name="nav-main"/> expects a certain model (now produced by leather-dev - tab2menu.xsl) to transform it. In the future this dependency will be on a businessHelper implementation that has to be defined in the view. For now it depends on leather-dev. I guess leather-dev is still not dead till the businessHelper plugin comes (which will implement the businessHelper of leather-dev). ;-)

Each forrest:css will be transformed into a <link rel="stylesheet" type="text/css" url="{$root}skin/{url}"> tag (in html output [only format that has an implementation]). This tags are designed to help **web-designers** to freely implemend their own css-implementations for a view.

Each forrest:hook will be transformed into a <div/> tag (in html output [only format that has an implementation]). This tags are designed to help **web-designers** to freely move around and group contracts into design container.

Each forrest:contract will dispatch a xml file (ViewHelper) that contains xsl:stylesheets (for now, but other implementations to transform the presentation model are/should be possible) for the different output formats of a contract. For html, the only format that we have implemented so far, we have three different kind of templates:

<xsl:templates name="{contract/@name}-css/>
Here the contract specific css will be stored.

<xsl:templates name="{contract/@name}-head/>
Here goes everything that belongs in the html head. e.g. scripts.

<xsl:templates name="{contract/@name}-body/>
Everything that belongs into the body.

3. BusinessHelper

businessHelper class/implementation - a class/implementation which will access businessServices to produce models for the presentation model. In other word it is a data producing factory.

> **Note:**
> Right now we are using the default forrest businessHelper (defined by the skin you are using). There are contracts that depend on a businessHelper implementation of leather-dev (tab2menu.xsl, site2xhtml.xsl,...). We will create a businessHelper plugin to overcome this problem.

Right now it is implemented for the forrest way of producing the models for the view but we will create an independent BusinessHelper to access all BusinessServices you may have in your application.

4. links

> **Note:**
> There are/were lots of mails on the dev-list. Please, search the list

Background mailings around the views concept

- [RT] plugin infrastructure
- [RT] Forrest templates for second generation skining - towards Corium