Create components for Joomla! - Managing the front end

Text out-of-date. The instructions are for versions prior to 1.5, really front end code too is not up to date! so you can read...

Creating components for Joomla! 1.5 - Managing the front end

Write components with Joomla! Component Builder is quite easy since it is the program that takes care to write the skeleton for models, views and templates for views. You have only to define the layout with which to present the data; in this article we will explain how to.

Operation of the components for Joomla!1.5

We would not still explain the operation of the MVC design in Joomla!™, it has already been described by the authors themselves, and that the article in question, it it is available also the Italian translation, we just follow the flow of data across different classes.

> “
>
> *A curiosity: the 'metaphor' model-view-controller, is not proper a recent idea, in fact it dates back to 1979, and it is described in an article on SmallTalk by Trygve Reenskaug. We started talking about 'design pattern' since August 1994 after the publication of "Design Patterns - Elements of reusable Object-Oriented Software" by Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides, reading strongly recommend to anyone, with some knowledge of OOP, who wants to improve himself.*
>
> ”

Each component of Joomla! has, whether we speak of front end or back end, an **entry point.** This is a php file with name EQUAL to that of the component, which recives the requests that must be handled by the component itself. This file will then define which **view** and which **controller** load.

Front - End Side

From the front end point of view, the entry point's work is simple, in fact, normally, access happens via a menu item, the view is already defined in the URL created by Joomla!.

http://www.example.com/index.php?**option=com_ccat&view=ccatbrands**

For example, the URLin involved means that Joomla! look for the file **ccat.php** in the folder **/components/com_ccat/**:

```
... [ccat.php]
// Require the base controller
require_once (JPATH_COMPONENT.DS. 'controller.php');
...
// Create the controller
$classname = 'CcatController'. $controller; / / $ controller == ""
$controller = new $classname ();
// Perform the Request task
$controller->execute(JRequest:: getVar ( 'task'));
```

File **ccat.php,** as shown by the code, loads the default controller.

- / components / com_ccat / controller.php

The default controller created by **Joomla!Component Builder,** in fact, has no own code, but it causes the call of the **constructor** and the **method display** (default if **$ task** is not defined) of the base class **JController.** This will force, due to the presence of **view=ccatbrands** in the URL, Joomla! to load the following files:

- /components/com_ccat/views/ccatbrands/view.html.php
- /components/com_ccat/views/ccatbrands/tmpl/default.php
- /components/com_ccat/models/ccatbrands.php

A step back:

1. when statement **$controller->execute(JRequest::getVar ('task'))** is executed, in the file **/components/com_ccat/ccat.php**,
2. method **display()** in the file **/components/com_ccat/ controller.php** is called
3. The controller call the method **display()** of the class *CcatViewCcatbrands* in **/components /com_ccat/views/ccatbrands/view.html.php**

Within the last file we have, among others, the following lines of code:

```
function display ($tpl = null) (
 $data = $this->get ('Data');
 $this->assignRef( 'data', $data);
 ...
 parent::display ($ tpl);
)
```

The header of the method display() include the optional parameter $tpl, Attention! *is not the name of the template to load, but the name of the sub-template!*

1. The code **$data = $this->get('Data')** invokes the method getData() in the file **/components /com_ccat/models/ccatbrands.php** (class CcatModelCcatbrands), this method makes the query on the db and returns a recordset as an array of object,s in the standard Joomla!.
2. the code **$this->assignRef('data', $data)** creates the property $this->data within the class *CcatViewCcatbrands* so that such information is easily accessible from the template.
3. the code **parent::display($tpl)** invokes and executes the file /components/com_ccat/views

/ccatbrands/tmpl/default.php that displays the contents of the recordset.

In the file **/components/com_ccat/views/ccatbrands/tmpl/default.php** you will find a code like this:

```
<?php foreach($this->data as $dataItem): ?>
        <?php
                $link = JRoute::_( "index.php?option=com_ccat&view=ccatbrand&
brand_id={$dataItem->id}" );
        ?>
        <div class="ccat_viewBrandsItemName" >
                <a href="/<?php echo $link; ?>"><?php echo $dataItem->name; ?></a>
        </div>
<?php endforeach; ?>
```

The **foreach** loop iterates through all the records retrieved from the method getData(), in the model, and assign them to $this->data from view with assignRef().

the code **$link = JRoute::_("...")** creates the URL to see the single record, in the following lines we see how to use this URL (<a href ="<?php echo $ link;?> ">) And how to retrieve data from the record set ($DataItem->name, where "name" is the name of the field in the table), **Joomla!Component Builder** will return all the fields of the reference table, it's to you to decide which to show, and write the code for generating the best layout.

As for the display of single record, the process is practically the same, obviously in the template there is no foreach: simple, no?

next & more interesting: manage the back end

Commenti

#2 **Guest** 2013-09-26 06:25 **0**

Hello sir, I wont to make a registation form component using MVC and display at frantend and store it in the backend table. so, if you have any this type of tutorial or any suggestion then please reply me.

Citazione

#1 **Guest** 2013-05-01 07:29 **0**

g

Citazione

Aggiorna elenco commenti

RSS feed dei commenti di questo post.

Aggiungi commento

E-Mail

Sito web

1000 caratteri rimasti

Aggiorna

Invia

JComments