Entry Point

Overview

The CMS is always accessed through a single point of entry: index.php for the Site Application or administrator/index.php for the Administrator Application. The application will then load the required component, based on the value of 'option' in the URL or in the POST data. For our component, the URL would be:

```
For search engine friendly URLs:
/hello

For non-SEF URLs:
/index.php?option=com_hello
```

This will load our main file, which can be seen as the single point of entry for our component: components/com_hello/hello.php.

Implementation

```
<?php
// No direct access
defined( '_JEXEC' ) or die( 'Restricted access' );

// Get the requested controller
$controllerName = JRequest::getCmd('controller', JRequest::getCmd('vie
w', 'one'));

// Ensure the controller exists
if (!file_exists(JPATH_COMPONENT . DS . 'controllers' . DS . $controll
erName . '.php'))
{
 $controllerName = 'one';
}
require_once(JPATH_COMPONENT . DS . 'controllers' . DS . $controllerNa
me . '.php');
$controllerName = 'ExampleController' . ucfirst(strtolower($controller
Name));

// Instantiate controller
$controller = new $controllerName();
```

```
// Execute whatever task(s)
$controller->execute();
// Redirect as needed
$controller->redirect();
```

The first statement is a security check.

JPATH_COMPONENT is the absolute path to the current component, in our case components/com_hello. If you specifically need either the Site component or the Administrator component, you can use JPATH_COMPONENT_SITE or JPATH_COMPONENT_ADMINISTRATOR.

DS is the directory separator of your system: either '/' or ''. This is automatically set by the framework so the developer doesn't have to worry about developing different versions for different server OSs. The 'DS' constant should always be used when referring to files on the local server.

First we look for a requested controller name. There is a default set in case none has been passed or if the requested controller is not found. With the controller name, we build the class name for the controller following the standard camel-cased pattern of {Component name}Controller{Controller name}

After the controller is created, we instruct the controller to execute the task, as defined in the URL: index.php?option=com_hello&task=sometask. If no task is set, the default task 'display' will be assumed. When display is used, the 'view' variable will decide what will be displayed. Other common tasks are save, edit, new...

The controller might decide to redirect the page, usually after a task like 'save' has been completed. This last statement takes care of the actual redirection.

The main entry point (hello.php) essentially passes control to the controller, which handles performing the task that was specified in the request.

Note that we don't use a closing php tag in this file: ?>. The reason for this is that we will not have any unwanted whitespace in the output code. This is default practice and will be used for all php-only files.