[翻译]教程：Joomla! 插件开发指南(序言)

Joomla! 开源天空 作者：joomla 2008-01-09 17:17

- 摘要：本文讲述了 Joomla!插件的开发基本过程，并给出一些 Joomla! 插件开发的参考实例。

Joomla! 开发百科 指南 插件(Plugins)

Joomla! 插件是事件响应程序，可响应 Joomla! 内核事件和用户自定义事件。编写插件是扩展 Joomla! 基本功能的一种有效手段。欲了解插件是如何在 Joomla! 全局构架上运行的，请参考框架概览。

Joomla! 插件架构服从 Observer 设计模式。利用继承自基类 JObserver 的类 JPlugin 提供的方法，可将用户自定义插件注册为核心事件或用户自定义事件的响应程序。当事件被触发时，继承自基类 JObservable 的类 JEventDispatcher 将调用注册到当前事件的所有插件。

如何创建自己的插件
如何在代码中使用插件
创建一个权限管理插件

参考：嵌入：基于事件的松散集成

在 Joomla! 1.5 之前的版本中，插件被称为 mambots。 Mambots 基于特定的标志名称，且只能用于修饰内容。在那之后，其功能逐渐扩展，因此开发人员决定将其改称为插件(Plugins)，以突出其新的功能。Joomla! 1.5 中延续了对 Joomla! 1.0 中的 mambots 的支持。

Joomla 文章内容插件的实例 （一） XML 文件

Joomla! 开源天空 作者：管理员 2008-01-28 09:26

- 摘要：本文描述了如何实现文章内容插件，主要集中简述了 XML 文件。

概述
利用文章内容插件可以完成许多工作，通常需要两个文件，一个是 xml 文件，另一个是 php 文件。这里我们通过两个例子给读者展示文章插件的功能，同时为了国际化，增加了 ini 文件部分。

XML 文件

XML 文件与 php 文件的名字相同，通常是 UTF-8 编码的。

```
<?xml version="1.0" encoding="utf-8"?>
 <!DOCTYPE install PUBLIC
  "-//Joomla! 1.5//DTD plugin 1.0//EN" "
http://dev.joomla.org/xml/1.5/plugin-install.dtd">
```

下面这一行定义了插件是文章内容插件：

```
<install version="1.5" type="plugin" group="content">
```

type 定义说明这是一个插件，而 group 定义这是一个文章内容插件。接下来，要定义一些插件紧密相关的信息：

```
<name>输入插件名称</name>
<creationDate>开发日期</creationDate>
<author>作者</author>
<authorEmail>e-mail</authorEmail>
<authorUrl>作者主页</authorUrl>
<copyright>Copyright</copyright>
<license>License</license>
<version>版本</version>
<description>描述
</description>
```

接下来需要通过 file 标签，定义你的 php 文件，代码如下：

```
<files>
   <filename plugin="nameofplugin">nameofplugin.php</filename>
</files>
```

接下来是国际化语言文件的定义,代码如下：

```
<languages>
   <language
tag="en-GB">language/en-GB/en-GB.plg_content_nameofplugin.INI</language>
</languages>
```

你可能还需要定义一些插件的参数，代码如下：

<params>
<param name="paramname" type="typeofparameter" default="defaultsetting" label="title" description="description"/>
</params>

Param name:参数名称，在 php 文件要通过这个名字来访问变量值
Param type: 参数类型，详情请参考
Param default: 参数默认值
Param label: 参数标签
Param description: 参数的描述

如果没有参数的话，你可以加入<params/>

Joomla 文章内容插件的实例(二) 文章投票的插件

Joomla! 开源天空 作者：管理员 2008-01-28 09:30

- 摘要：Joomla 插件的实际例子，文章投票插件
-

例子一：
这是一个文章投票的插件，以下是代码清单已经注释。

```
<?php

//First start with some information about the Content Plugin, its author and probably some other information
/**
* @version  $Id: vote.php 9764 2007-12-30 07:48:11Z ircmaxell $
* @package  Joomla
* @copyright Copyright (C) 2005 - 2008 Open Source Matters. All rights reserved.
* @license  GNU/GPL, see LICENSE.php
* Joomla! is free software. This version may have been modified pursuant
* to the GNU General Public License, and as distributed it includes or
* is derivative of works licensed under the GNU General Public License or
* other free or open source software licenses.
```

```
* See COPYRIGHT.php for copyright notices and details.
*/

//You can not directly access the file when you put this code at the beginning:
// no direct access
defined( '_JEXEC' ) or die( 'Restricted access' );

//下面的程序在系统中注册了插件，在显示正文之前，将触发 plgContentVote 函数调用
$mainframe->registerEvent( 'onBeforeDisplayContent', 'plgContentVote' );

function plgContentVote( &$row, &$params, $page=0 )
{

//获取 URI
 $uri = & JFactory::getURI();

 $id  = $row->id;

//初始化变量
 $html  = '';

//获取参数和 rating_count
 if (isset($row->rating_count) && $params->get( 'show_vote' )
&& !$params->get( 'popup' ))
 {

//加载语言文件
  JPlugin::loadLanguage( 'plg_content_vote' );

//添加 html 输出
  $html .= '<form method="post" action="' . $uri->toString( ) . '">';
  $img = '';

//查找图片
  $starImageOn  = JHTML::_('image.site',  'rating_star.png', '/images/M_images/' );
```

```
  $starImageOff  = JHTML::_('image.site',  'rating_star_blank.png',
'/images/M_images/' );

//如果评价大约 0，显示几个星星
  for ($i=0; $i < $row->rating; $i++) {
   $img .= $starImageOn;
  }

//如果 rate 大于 5，不显示星星
  for ($i=$row->rating; $i < 5; $i++) {
   $img .= $starImageOff;
  }

//HTML 输出
  $html .= '<span class="content_rating">';
  $html .= JText::_( 'User Rating' ) .':'. $img .' / ';
  $html .= intval( $row->rating_count );
  $html .= "</span>\n<br />\n";

//获取参数
  if (!$params->get( 'intro_only' ))
  {

//设置输出
   $html .= '<span class="content_vote">';
   $html .= JText::_( 'Poor' );
   $html .= '<input type="radio" alt="vote 1 star" name="user_rating" value="1" />';
   $html .= '<input type="radio" alt="vote 2 star" name="user_rating" value="2" />';
   $html .= '<input type="radio" alt="vote 3 star" name="user_rating" value="3" />';
   $html .= '<input type="radio" alt="vote 4 star" name="user_rating" value="4" />';
   $html .= '<input type="radio" alt="vote 5 star" name="user_rating" value="5"
checked="checked" />';
   $html .= JText::_( 'Best' );
   $html .= ' <input class="button" type="submit" name="submit_vote" value="'.
JText::_( 'Rate' ) .'" />';
```

```
    $html .= '<input type="hidden" name="task" value="vote" />';
    $html .= '<input type="hidden" name="option" value="com_content" />';
    $html .= '<input type="hidden" name="cid" value="'. $id .'" />';
    $html .= '<input type="hidden" name="url" value="'.  $uri->toString( ) .'" />';
    $html .= '</span>';
  }
//HTML output; close the <form> tag
  $html .= '</form>';
 }
//返回 html
 return $html;
}
```

Joomla 文章内容插件的实例（三）调用模块内容的插件

Joomla! 开源天空 作者：管理员 2008-01-29 10:08

- 摘要：本文是 Joomla 文章内容插件的一个例子，在文章正文中调用模块内容

例子二：
这个例子中，插件将调用模块，展示在文章最终页面中。个人觉得这个例子更有参考价值！

```
<?php
/**
* @version  $Id: loadmodule.php 9764 2007-12-30 07:48:11Z ircmaxell $
* @package  Joomla
* @copyright Copyright (C) 2005 - 2008 Open Source Matters. All rights reserved.
* @license  GNU/GPL, see LICENSE.php
* Joomla! is free software. This version may have been modified pursuant
* to the GNU General Public License, and as distributed it includes or
* is derivative of works licensed under the GNU General Public License or
* other free or open source software licenses.
* See COPYRIGHT.php for copyright notices and details.
*/
```

```
//You can not directly access the file when you put this code at the beginning:
// no direct access
defined( '_JEXEC' ) or die( 'Restricted access' );

//下面的程序在系统中注册了插件，检查正文中的内容,在合适的地方，将调用
plgContentLoadModule 函数调用
$mainframe->registerEvent( 'onPrepareContent', 'plgContentLoadModule' );

//函数代码
function plgContentLoadModule( &$row, &$params, $page=0 )
{
//访问数据库
 $db =& JFactory::getDBO();
//检查是否应该执行这个插件的功能
 if ( JString::strpos( $row->text, 'loadposition' ) === false ) {
  return true;
 }

//获取插件信息
 $plugin =& JPluginHelper::getPlugin('content', 'loadmodule');

//查找正文中的的标签
  $regex = '//i';

//获取参数
  $pluginParams = new JParameter( $plugin->params );

//检查插件是否已经发布
 if ( !$pluginParams->get( 'enabled', 1 ) ) {
  $row->text = preg_replace( $regex, '', $row->text );
  return true;
 }

//查找所有正文中匹配的地方
 preg_match_all( $regex, $row->text, $matches );
```

```
//匹配的总数
  $count = count( $matches[0] );

//如果有匹配，那么就执行插件
  if ( $count ) {
//取得参数
   $style = $pluginParams->def( 'style', -2 );

   plgContentProcessPositions( $row, $matches, $count, $regex, $style );
 }
}
//正文内容匹配位置处理函数
function plgContentProcessPositions ( &$row, &$matches, $count, $regex, $style )
{
  for ( $i=0; $i < $count; $i++ )
 {

//处理匹配部分内容，去掉
   $load = str_replace( 'loadposition', '', $matches[0][$i] );
   $load = str_replace( '{', '', $load );
   $load = str_replace( '}', '', $load );
   $load = trim( $load );

//在匹配位置调用 plgContentLoadPosition
  $modules = plgContentLoadPosition( $load, $style );
  $row->text  = preg_replace( '{'. $matches[0][$i] .'}', $modules, $row->text );
  }

//去除无关标记
 $row->text = preg_replace( $regex, '', $row->text );
}

//调用模块的函数
function plgContentLoadPosition( $position, $style=-2 )
```

```
{

//获取 document 实例
 $document = &JFactory::getDocument();

//加载模块
 $renderer = $document->loadRenderer('module');

//加载参数
 $params  = array('style'=>$style);

 $contents = '';

 foreach (JModuleHelper::getModules($position) as $mod)  {
  $contents .= $renderer->render($mod, $params);
 }

//返回模块内容
 return $contents;
```

Joomla 文章内容插件的实例（四） 语言文件

Joomla! 开源天空 作者：管理员 2008-01-29 10:20

- 摘要：为了国际化，最好使用语言文件，在需要显示文字串的地方，采用 JText::_('string')来显示

语言文件（INI 文件）
为了国际化，最好使用语言文件，在需要显示文字串的地方，采用 JText::_('string')来显示

比如，以下的参数

<param name="mode" type="list" default="1" label="Mode" description="Select how the emails will be displayed">

```
    <option value="0">Nonlinkable Text</option>
    <option value="1">As linkable mailto address</option>
</param>
```

实际应该在语言文件中设置相应的字串：

```
MODE=Mode
SELECT HOW THE EMAILS WILL BE DISPLAYED=Select how the e-mails will be displayed
NONLINKABLE TEXT=Nonlinkable text
AS LINKABLE MAILTO ADDRESS=As linkable mailto addressThe file looks repetitive, but will be very useful for translaters.
```

Joomla 文章内容插件的实例（五） 其他参考

Joomla! 开源天空 作者：管理员 2008-01-29 10:47

- 摘要：Joomla 中已有一些文章内容插件，下面是列表，学习这些插件，你可以获得关于插件的更多知识。

Joomla 中已有一些文章内容插件，下面是列表，学习这些插件，你可以获得关于插件的更多知识：

```
Plugin Content - Emailcloak
\plugins\content\emailcloak.XML
\plugins\congent\emailcloak.PHP
Plugin Content - Example
\plugins\content\example.XML
\plugins\congent\example.PHP
Plugin Content - Geshi
\plugins\content\geshi.XML
\plugins\congent\geshi.PHP
Plugin Content - Load Module
\plugins\content\loadmodule.XML
\plugins\congent\loadmodule.PHP
Plugin Content - Pagebreak
\plugins\content\pagebreak.XML
\plugins\congent\pagebreak.PHP
```

Plugin Content - Page navigation

\plugins\content\pagenavigation.XML

\plugins\congent\pagenavigation.PHP

Plugin Content - Vote

\plugins\content\vote.XML

\plugins\congent\vote.PHP

\language\en-GB\en-GB.plg_content_vote.INI

个人觉得第二个自理更有使用价值，估计 pagebreak 插件跟第二个例子差不多！

【附】

[翻译]教程：Joomla! 插件开发指南(事件)

Joomla! 开源天空 作者：joomla 2008-01-09 17:30

- 摘要：
-

Joomla! 开发百科 » 指南 » 插件(Plugins) » 事件(Events)

Joomla! 提供了一系列内核事件，分为四类：

- 系统事件(sytem)
- 用户事件(user)
- 编辑事件(editor)
- 内容事件(content)

以下是 Joomla! 1.5 中提供的内核事件列表(按字母排序，后面括号中是事件的类别)：

- onAfterDeleteUser (user)
- onAfterDisplay (content)
- onAfterDisplayTitle (content)
- onAfterDisplayContent (content)

- onAfterInitialise (system)
- onAfterRoute (system)
- onAfterDispatch (system)
- onAfterRender (system)
- onAfterStoreUser (user)
- onAuthenticate (authentication)
- onAuthenticateFailure (authentication)
- onBeforeDeleteUser (user)
- onBeforeDisplay (content)
- onBeforeDisplayContent (content)
- onBeforeStoreUser (user)
- onCustomEditorButton (editors-xtd)
- onDisplay (editors)
- onGetContent (editors)
- onGetInsertMethod (editors)
- onGetWebServices (xmlrpc)
- onInit (editors)
- onLoginUser (user)
- onLogoutUser (user)
- onPrepareContent (content)
- onSave (editors)
- onSearch (search)
- onSearchAreas (search)
- onSetContent (editors)

创建 Joomla 1.5 的认证插件教程 上

Joomla! 开源天空 作者：管理员 2008-01-26 12:48

- 摘要：本文描述如何创建 Joomla 1.5 的认证插件

简介

新的 Joomla 1.5 插件认证系统非常灵活，通过这个系统可以认证任何数据源的用户，比如内部数据库，Open ID 系统，轻量级活动目录，或者任何能用 php 访问的认证系统。

本指南通过一个认证插件的实例，展示如何创建自定义的认证插件。

plgAuthenticationMyauth 类

Joomla 1.5 的插件都是 JPlugin 的子类，而 JPlugin 提供了所有鼻血的基本框架和功能，在扩展过程中我们所需要做的只是给出处理预期事件的必要方法。

认证插件的类名称必须以 plgAuthentication 开始，以插件的名字结尾，本例中，插件的名字是 Myauth，因而扩展插件子类的名称是 plgAuthenticationMyauth。这个类有两个方法，一个是构造器，一个是 onAuthenticate()方法，这些方法都非常简单。

plgAuthenticationMyauth() 方法

构造器有一个引用参数，构造器的功能仅仅是将这个参数传递给父类的构造器。构造器的代码如下：

```
function plgAuthenticationMyauth(& $subject) {
  parent::construct($subject);
}
```

父类的构造器将事件的观察者关联在事件分派者。

onAuthenticate() 方法

onAuthenticate 方法在系统验证用户的时候被调用，这个方法有三个参数，用户名，密码，以及一个 JAuthenticationResponse 对象的引用。这个方法验证用户密码是否匹配，并将结果在 JAuthenticationResponse 对象中返回。

本例中认证检查非常简单，我们仅仅检查用户名是否在用户表中匹配，并且密码是否匹配。代码如下：

```
$db =& JFactory::getDBO();
$query = 'SELECT `id`'
    . ' FROM #__users'
    . ' WHERE username=' . $db->quote( $username );
$db->setQuery( $query );
$result = $db->loadResult();


$db =& JFactory::getDBO();
$query = 'SELECT `id`'
    . ' FROM #__users'
    . ' WHERE username=' . $db->quote( $username );
$db->setQuery( $query );
$result = $db->loadResult();

if (!$result) {
    $response->status = JAUTHENTICATE_STATUS_FAILURE;
    $response->error_message = 'User does not exist';
}

if($result && ($username == strrev( $password )))
{
    $email = JUser::getInstance($result); // Bring this in line with the rest of the
system
    $response->email = $email->email;
    $response->status = JAUTHENTICATE_STATUS_SUCCESS;
}
else
{
    $response->status = JAUTHENTICATE_STATUS_FAILURE;
    $response->error_message = 'Invalid username and password';
}
```

如果认证失败，response 对象需要设置两个属性，状态属性和错误提示。状态属性有三个可选

JAUTHENTICATE_STATUS_SUCCESS

JAUTHENTICATE_STATUS_FAILURE

JAUTHENTICATE_STATUS_CANCEL.
关于状态值的更多信息，请查看 libraries/joomla/user/authentication.php

认证失败的时候需要设置错误提示属性，本例中属性有两个可能，一个是“用户不存在”，一个是“非法的用户和密码”，需要注意的是这些并不返回给用户，由于安全原因，用户只能看到用户密码不匹配，或者成功登录。

如果认证成功，我盟可以有选择的返回一些附加信息，本例中我们返回了 email 信息。要知道更详细的可以在 response 对象中返回什么数据，请查看http://api.joomla.org .这些信息可以被时间中 plugins 使用来创建用户或者执行其他任务。

完整的 myauth.php 清单：

```
<?php
/**
 * @version    $Id: myauth.php 7180 2007-04-23 16:51:53Z jinx $
 * @package    Joomla.Tutorials
 * @subpackage Plugins
 * @license    GNU/GPL
 */

// Check to ensure this file is included in Joomla!
defined('_JEXEC') or die();

jimport('joomla.event.plugin');

/**
 * Example Authentication Plugin.  Based on the example.php plugin in the Joomla! Core installation
 *
 * @package    Joomla.Tutorials
 * @subpackage Plugins
 * @license    GNU/GPL
 */
class plgAuthenticationMyauth extends JPlugin
{
    /**
     * Constructor
     *
     * For php4 compatability we must not use the __constructor as a constructor for
```

```
plugins
     * because func_get_args ( void ) returns a copy of all passed arguments NOT references.
     * This causes problems with cross-referencing necessary for the observer design
pattern.
     *
     * @param object $subject The object to observe
     * @since 1.5
     */
    function plgAuthenticationMyauth(& $subject) {
        parent::__construct($subject);
    }

    /**
     * This method should handle any authentication and report back to the subject
     * This example uses simple authentication - it checks if the password is the reverse
     * of the username (and the user exists in the database).
     *
     * @access    public
     * @param    string    $username    Username for authentication
     * @param    string    $password    Password for authentication
     * @param    object    $response    Authentication response object
     * @return    boolean
     * @since 1.5
     */
    function onAuthenticate( $username, $password, &$response )
    {
        /*
         * Here you would do whatever you need for an authentication routine with the
credentials
         *
         * In this example the mixed variable $return would be set to false
         * if the authentication routine fails or an integer userid of the authenticated
         * user if the routine passes
         */
        $db =& JFactory::getDBO();
        $query = 'SELECT `id`'
```

```
            . ' FROM #__users'
            . ' WHERE username=' . $db->quote( $username );
        $db->setQuery( $query );
        $result = $db->loadResult();

        if (!$result) {
            $response->status = JAUTHENTICATE_STATUS_FAILURE;
            $response->error_message = 'User does not exist';
        }
        // to authenticate, the username must exist in the database, and the password
should be equal
        // to the reverse of the username (so user joeblow would have password wolbeoj)
        if($result && ($username == strrev( $password )))
        {
            $email = JUser::getInstance($result); // Bring this in line with the rest
of the system
            $response->email = $email->email;
            $response->status = JAUTHENTICATE_STATUS_SUCCESS;
        }
        else
        {
            $response->status = JAUTHENTICATE_STATUS_FAILURE;
            $response->error_message = 'Invalid username and password';
        }
    }
}
?>
```

创建 Joomla 1.5 的认证插件教程 下

Joomla! 开源天空 作者：管理员 2008-01-26 13:02

- 摘要：本文是如何创建 Joomla 1.5 认证插件的续篇

XML 安装文件

以下是安装文件的代码清单：

```
<?xml version="1.0" encoding="utf-8"?>
<install version="1.5" type="plugin" group="authentication">
    <name>Authentication - Myauth</name>
    <author>Joomla! Documentation Project</author>
    <creationDate>May 30, 2007</creationDate>
    <copyright>(C) 2005 - 2007 Open Source Matters. All rights reserved.</copyright>
    <license>http://www.gnu.org/copyleft/gpl.html GNU/GPL</license>
    <authorEmail> ian.maclennan@help.joomla.org</authorEmail>
    <authorUrl>www.joomla.org</authorUrl>
    <version>1.5</version>
    <description>An sample authentication plugin</description>
    <files>
        <filename plugin="myauth">myauth.php</filename>
    </files>
    <params/>
</install>
```

这个文件与其他的 Joomla xml 安装文件非常相似，但是有几点需要注意。

首先根元素的 group 属性是 authentication，其次 version 属性必须是 1.5. name 元素的值是 Authentication - Myauth，这个名在插件管理器中显示，你可以改变这个名字。最后要注意 filename 元素有一个属性 plugin，属性的值是 plugin 的名称，本例中是 myauth。

到目前为止，我们已经创建了认证插件，安装试用吧。

如何在程序代码中使用插件

Joomla! 开源天空 作者：管理员 2008-01-26 16:20

- 摘要：本文描述了如何在代码中使用插件
-

插件使用起来非常简单，重要的是更好的设计，因为某种程度上来说，你在提供其他人使用的接口。

你可以使用下面的代码来激活插件。

$result = $mainframe->triggerEvent('onSomething', $param);

这一行的程序会调用所有注册为‘onSomething’事件处理函数的插件，并传递一个$param 参数。triggerEvent 返回一个 results 数组，参数和返回结果都是可选，可以根据实际情况使用。

[翻译]教程：Joomla! 插件开发指南(创建插件)

Joomla! 开源天空 作者：joomla 2008-01-10 09:30

- 摘要：
-

Joomla! 开发百科 » 指南 » 插件(Plugins) » 创建插件

在 Joomla! 1.5 中创建插件

如何创建自己的插件

本文将展示开发你自己的插件所需的基本知识。 大部分插件仅包含一个单独的代码文件，但为了正确地安装插件，必须将它打包成 Joomla! 安装程序(installer)能够识别和处理的安装文件。

创建安装文件

同 Joomla! 的其他扩展附件一样，打包成 .zip 文件(或 .tar.gz 文件)的插件安装起来十分容易，不过安装包中必须包含一个格式正确的 XML 文件。这里有一个例子，是目录搜索机器人(categories searchbot)插件的 XML 安装文件：

```
<?xml version="1.0" encoding="iso-8859-1"?>
<install version="1.5" type="plugin" group="search">
    <name>Categories searchbot</name>
    <author>Joomla! Project</author>
    <creationDate>November 2005</creationDate>
    <copyright>(C) 2005 Open Source Matters. All rights
```

```
reserved.</copyright>
    <license>GNU/GPL</license>
    <authorEmail> admin@joomla.org</authorEmail>
    <authorUrl>www.joomla.org</authorUrl>
    <version>1.1</version>
    <description>Allows searching of Categories
information</description>
    <files>
        <filename
plugin="categories.searchbot">categories.searchbot.php</filename>
    </files>
    <params>
        <param name="search_limit" type="text" size="5" default="50"
label="Search Limit" description="Number of search items to
return"/>
    </params>
</install>
```

正如你所看到的，它与其他 Joomla! XML 安装文件非常相似。你只需找到 <install> 标签中的属性 group=”xxx”，以及 <filename> 标签中的信息。这些信息告诉 Joomla! 这个插件属于哪一个类别(group)，及应当把文件复制到哪个文件夹下。

创建插件

Joomla! 1.5 提供了一种新的、更符合对象思想的方式来编写插件。为了向前兼容，旧的方式仍然可以使用(见下一小节)。

```
<?php
// 禁止直接访问
defined( '_JEXEC' ) or die( 'Restricted access' );

// 导入所需的库文件
jimport('joomla.event.plugin');

class plg<PluginGroup><PluginName> extends JPlugin
{
    /**
    * 构造函数
```

```
* 为兼容 php4，我们不能直接使用 __constructor 作为插件的构架函数，因为
func_get_args ( void )
* 返回的是所有参数的拷贝，而不是引用，在交叉引用——这在观察者(observer)设计模式中
是必须的——
* 将导致错误。
*/
    function plg<PluginGroup><PluginName>( &$subject )
    {
            parent::__construct( $subject );

            // load plugin parameters
            $this->_plugin = > JPluginHelper::getPlugin( '<GroupName>',
'<PluginName>' );
            $this->_params = new JParameter( $this->_plugin->params );
    }

    /**
    * 与事件同名的插件方法将被自动调用。
    */
    function <EventName>()
    {
        global $mainframe;

        // Plugin code goes here.

        return true;
    }

}
```

创建插件(兼容模式)

这一节中将介绍 Joomla! 1.5 之前的版本中使用的插件编写方法，出于向前兼容的目的，这种方法目前仍然受支持。部分 Joomla! 核心插件可能仍然使用这种方式编写，不过随着时间推移，这些插件将逐步被重写。

你希望在事件触发时执行的代码应该被写成 PHP 函数的形式。在函数定义之前，你应当调用 registerEvent() 方法，以使 Joomla! 事件系统能将你的函数和相应的事件关联起来。

请看以下的代码框架示意：

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

$mainframe->registerEvent( '<EventName>', '<FunctionName>' );

function <FunctionName>( <ParameterList> ) {

    //Plugin code goes here

}
```

通过函数 $mainframe→registerEvent()，你的插件在 Joomla! 事件系统中通过注册。这意味着在此之后，当名为 ‘<EventName>’ 的事件被触发时，函数 ‘<FunctionName>’ 将被调用。现在，你可以编写任何你想要的插件函数了。如果你想给函数添加参数，没问题，就像平常一样使用它们。你可以在一个文件里注册任意多个事件和插件函数。完成之后，你的插件就可用了。

在代码中调用插件

现在，你已经创建了自己的插件，你很可能想在代码中调用它。当然，Joomla! 内核中包括了一系列内置事件，你可能将你的插件绑定到这些内置事件上，在这种情形下你不必再多此一举了。如果你希望触发一个事件，可以这样编码：

```
$results = $mainframe->triggerEvent( '<EventName>', <ParameterArray> );
```

值得注意的是事件参数必须写在一个数组中，插件函数本身会逐个读取这些参数。返回值是一个由所有与此事件关联的插件的返回值组成的数组(因此可能是一个多维数组)。

结论

Joomla! 1.5 的插件结构鲁棒且具有可扩展性。插件不仅可以用于处理内核程序和各种扩展所触发的事件，而且使得第三方扩展的可扩展性更好，从而变得更强大。

如何使用 Joomla 用户事件的插件系统

Joomla! 开源天空 作者：管理员 2008-02-02 10:22

- 摘要：在这篇文章中，简单介绍了八个 Joomla 用户事件，并简单描述了在什么情况下，如何使用用户插件系统，并给出了一段示意程序。

因为遇到要删除用户的时候，清除这个用户所发表的所有评论的要求，所以看了一下 Joomla 插件系统的用户事件。

总共有八个用户事件分为两大类：

第一类是用户登录和认证相关事件：

onLoginUser

onLogoutUser

onAuthenticate

onAuthenticateFailure

第二类是用户管理过程中的事件

onBeforeStoreUser

onAfterStoreUser

onBeforeDeleteUser

onAfterDeleteUser

joomla 系统中有一个 plgUserJoomla 插件的例子，就是用户事件使用的实例。这个例子没有做什么实际的工作，就是完成用户事件插件的框架，我们可以根据这个例子开发自己的插件。

比如我自己的要在清除用户之前删除他所有的评论，就可以使用 onBeforeDeleteUser

```
function onBeforeDeleteUser($user)
{
 global $mainframe;
 $query='delete 语句'
 ......
 //执行
}
```

这样就完成了功能。

Joomla 的 yvcomment 评论插件的实现方式分析

Joomla! 开源天空 作者：管理员 2008-01-26 23:53

摘要：本文阐述了 Joomla yvcomment 插件中的一些优缺点。

-

最近的apache日志中总是出现很多500错误,大部分是HEAD方式请求文章的最终页面,用telnet一试，代码输出如下：

```
HEAD /index.php/hotspot/39-joomla-inspect/320-joomla-source-research.html HTTP/1.1
HOST:www.maycode.com

HTTP/1.1 500 Internal Server Error
Date: Sat, 26 Jan 2008 15:32:46 GMT
Server: Apache/2.2.6 (Unix) PHP/5.2.5
X-Powered-By: PHP/5.2.5
Set-Cookie: 2a7dc0628b73c1cc08bebba5022556b5=ub7n10kh4h0t5qke9m8l4vgpq6; path=/
P3P: CP="NOI ADM DEV PSAi COM NAV OUR OTRo STP IND DEM"
Status: 500 View not found [name, type, prefix]: article,html,yvcommentView
Expires: Mon, 1 Jan 2001 00:00:00 GMT
Last-Modified: Sat, 26 Jan 2008 15:32:53 GMT
Cache-Control: post-check=0, pre-check=0
Pragma: no-cache
Connection: close
Content-Type: text/html; charset=utf-8
```

原来我采用的yvcomment评论插件在head请求方式存在错误，只好看看yvcomment的源代码。看了以后觉得yvcomment的插件部分实现的很巧妙，但是同时也存在问题。

巧妙的是通过以下几行代码，就直接在插件中调用的组件的界面，输出了评论列表和评论输入框：

```
$controller = new yvcommentController($config);

//$id =  $controller->getArticleID();
//$strOut .= ', ArticleID=' . $id;

// Perform the Request task
$controller->execute($task);

$strOut .= $controller->getOutput();
```

但是问题出在为了实现这个问题，修改了viewname和task，而最终又没有恢复，这样的话，如果在这个插件之后被调用的文章中页面的插件肯定会取道错误的参数。同时，yvcomment.php 中前边的

```
if (!$yvComment) {
```

```
    $path = JPATH_SITE . DS . 'components' . DS . 'com_yvcomment' . DS . 'helpers.php';
    .........
}
```

部分应该没有有效的封装在类中，直接被调用，就是这些地方导致了在 HEAD 方式请求会出错。

最后没有修改代码，因为这部分代码实在不忍心看下去了。

呵呵，不该骂厨子！还是有空自己改一个吧。