

Joomla 1.0 Extension Development Training

Learning to program for Joomla



Objectives & Requirements

- Learn to develop basic Joomla Mambots, Modules and Components.
- Familiar with PHP and MySQL programming.
- Familiar with Joomla! administration is preferable.
- **Required** Joomla! preinstalled.
- **Required** phpMyAdmin.
- **Required** good PHP editor e.g. phpDesigner 2008, phpEclipse.

Agenda

- Introduction to Joomla
- Joomla Template
- Joomla Module Development
- Module Packaging
- Introduction to Joomla Component Development
- Component Front end development
- Component Backend development
- Component Packaging
- Mambot development
- Mambot Packaging

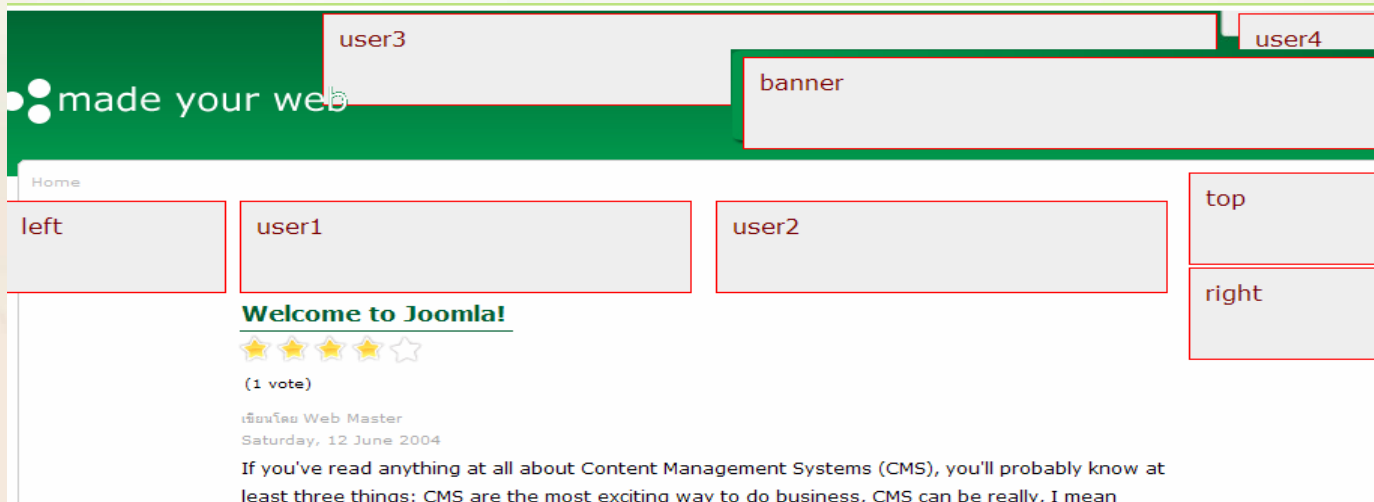
What is Joomla

- Open source, PHP based library; Object Oriented Programming (OOP)
- Best for web master
- Content Management System (CMS)
- Online Editor; WYSIWYG
- Web Application Framework
- Installation System
- Template Management System
- Design Patterns i.e. Model-View-Controller
- Internationalization version 1.0.12 upward uses UTF-8



Joomla Template

- Site layouts; One for all pages with exception for a page.
- Separate logic from design.
- Empowered by CSS and Joomla Template commands.
- Consist of module positions and main area.
- To view like following picture, use `index.php?tp=1`



Template directory

- Templates (folder)
 - **template name (folder)**
 - **index.php**
 - **index.html**
 - **template_thumbnail.png**
 - **templateDetails.xml**
 - **css (folder)**
 - **template_css.css**
 - **Other files**
 - **Images (folder)**

Template commands

- `mosShowHead()` →display header
- `mosPathway()`→display Path way navigation
- `mosCountModules($area)`→return >0 if more thane one modules exist in area "\$area"
- `mosLoadModules($area, $option)`→display modules in \$area
 - **Option = 1** **Horizontal display**
 - **Option = 0** **Normal display**
 - **Option = -1** **Raw**
 - **Option = -2** **XHTML compliant**
 - **Option = -3** **Extra DIV (rounded corner)**

Template Head

```
<?php defined( '_VALID_MOS' ) or die( 'Direct Access to this location is not allowed.' ); ?>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml" lang="<?php echo
  _LANGUAGE; ?>" xml:lang="<?php echo _LANGUAGE; ?>"
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; <?php echo _ISO;
  ?>" />
```

```
<?php if ($my->id) { initEditor(); } ?>
```

```
<?php mosShowHead(); ?>
```

```
<script type="text/javascript"> </script>
```

```
<link href="templates/<?php echo $cur_template; ?>/css/template_css.css"
  rel="stylesheet" type="text/css" media="screen" />
```

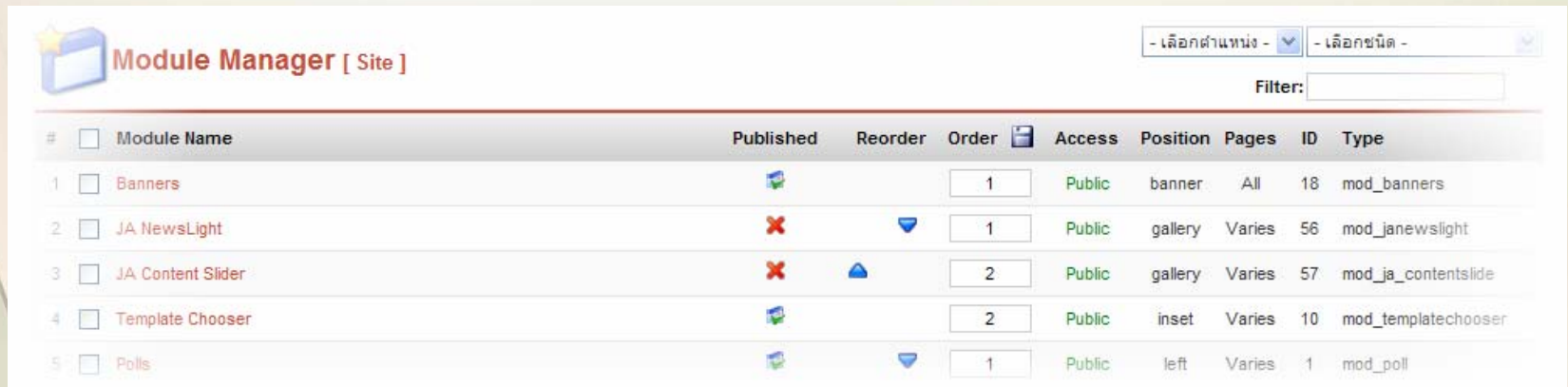
```
</head>
```

Template Body








```
<body>  
  <!-- 1 --><?php echo $mosConfig_sitename;?>  
  <!-- 2 --><?php mospathway ()?>  
  <!-- 3 --><?php mosLoadModules ('top');?>  
  <!-- 4 --><?php mosLoadModules ('left');?>  
  <!-- 5 --><?php mosMainBody();?>  
  <!-- 6 --><?php mosLoadModules ('right');?>  
  <!-- 7 --><?php include_once(  
    $mosConfig_absolute_path './includes/footer.php' );?>  
</body>  
</html>
```

Joomla Module Development

- Module is a small “windows”, “portlets”, “blocks”
- Fill in template “position” or “module positions”
- Small packet of functions
- Example of modules → menu system
- Link with menu (display some module on some pages)



The screenshot shows the Joomla! Module Manager interface. At the top left is the Joomla! logo and the text "Module Manager [Site]". On the top right, there are two dropdown menus: "- เลือกตำแหน่ง -" and "- เลือกชนิด -", and a "Filter:" input field. Below these is a table with the following columns: #, Module Name, Published, Reorder, Order, Access, Position, Pages, ID, and Type. The table contains five rows of module data.

#	Module Name	Published	Reorder	Order	Access	Position	Pages	ID	Type
1	Banners			<input type="text" value="1"/>	Public	banner	All	18	mod_banners
2	JA NewsLight			<input type="text" value="1"/>	Public	gallery	Varies	56	mod_janewslight
3	JA Content Slider			<input type="text" value="2"/>	Public	gallery	Varies	57	mod_ja_contentslide
4	Template Chooser			<input type="text" value="2"/>	Public	inset	Varies	10	mod_templatechooser
5	Polls			<input type="text" value="1"/>	Public	left	Varies	1	mod_poll

How to code it?

- Consist of at least 2 files
 - **mod_modulename.php** →function of module
 - **mod_modulename.xml** →use by installer to install module

“modulename” is name of your module



mod_banner.php
mod_banner.xml

Module Package File (XML)

```
<?xml version="1.0" encoding="utf-8"?>  
<mosinstall version="1.0.15" type="module">  
  <name></name>  
  <version></version>  
  <creationDate></creationDate>  
  <author></author>  
  <authorEmail></authorEmail>  
  <authorUrl></authorUrl>
```

Module Package File (XML)

```
<copyright> </copyright>  
<license> </license>  
<description> </description>  
<files>  
  <filename  
  module="mod_recentregistered" >mod_recentregistered.  
  php</filename>  
</files>  
<params />  
</mosinstall>
```

Types of standard param.

- **text** - a normal text field
- **textarea** - a normal textarea field. You can also add attributes for the rows and cols.
- **list** - a normal select list form field. Lists can have any number of option child tags.
- **radio** - a radio group. The radio type can have any number of option child tags
- **spacer** - shows a html horizontal rule.
- **imagelist** - shows a select list of images. You also provide a required directory attribute (the default is /images/stories), an optional hide_none attribute (either 0 or 1, where 1 will show a "Use no image" option) and a hide_default attribute (either 0 or 1, where 1 will show a "Use default image" option).
- **mos_category** - shows a select list of Joomla content categories
- **mos_section** - shows a select list of Joomla content sections.
- **mos_menu** - shows a list of Joomla menu items.

Module File (php)

```
<?php
```

```
// no direct access
```

```
defined( '_VALID_MOS' ) or die( 'Restricted access' );
```

```
$count = intval( $params->get( 'count', 10 ) );
```

```
$ordering = $params->get( 'ordering', 'hits' );
```

```
$skin = $params->get( 'skin', 'default' );
```

```
?>
```

More about Modules

- Standalone modules
- Component based modules
- User modules
- Did you know, you can have one or more modules in each position. Just copy, rename, set parameter and publish.
- Module parameters define in XML file, save in database.

\$database object variable

- MySQL connector object with lot of good methods.
- You can insert/update without writing any query.
- `setQuery()`
- `query()`
- `loadResult ()`
- `loadObject ($object)`
- `loadObjectList ()`
- `insertObject ($tablename, $object, $pk)`
- `updateObject ($tablename, $object, $pk)`
- `getNullDate();`
- `getErrorMsg();`

\$database – How to

```
global $database;
```

```
$sql = "SELECT * FROM #__weblinks WHERE published=1";
```

```
$database->setQuery($sql);
```

```
$rows = $database->loadObjectList();
```

```
$content = "";
```

```
for ($i=0; $i<count($rows); $i++) {
```

```
    $row = &$rows [$i];
```

```
    $content = "<br /><a href=\"\" alt=\"{$row->description}\">";
```

```
    $content .= $row->title;
```

```
    $content .= "</a>\n";
```

```
}
```

Workshop - Modules

- Display web links in **specified category**
 - `cat_id` as parameter

Component Development

- Main functional unit.
- Extends Joomla! functionality.
- Examples are web board, document management, Site map.
- Frontend part and Backend Part.
- No rules for what in frontend and what in backend.
- Components may come together with modules or plugins/mambots.

Development Steps

- Create empty *.php files
- Create XML installation package.
- Zip the files, and install via Joomla! backend (localhost).
- Open installed files and start development on those files.
- If menu is needed, you can create menu.
- Using this method, no need to deal with #__components table and easy for menu and submenu creation at backend.

Component Package File

- General part, `<name></name>` is defined component name.
- Frontend files part.
- Installation and uninstallation queries part.
- Post-installation and Post-uninstallation part.
- Backend files, menu and submenu, parameters.
- Be aware that this XML file, you have to put HTML tags in CDATA.

Joomla! 1.0.X Design Patterns

- Basic design pattern. It uses a lot switch statement relies on passing of \$task and \$option variables by \$_GET and \$_POST.
- Normal frontend requests come through index.php.
- Joomla! filters, set up environment, session management and user management.
- Index2.php uses for popup on frontend. Used by PDF generation (remove template from outside component area)
- /index2.php?option=com_example&task=do_something&no_html=1&popup=1

Design Patterns (Cont.)

- `/administrator/index.php` uses for login page only.
- `/administrator/index2.php` uses for normal backend request. We normally link to this file.
- `/administrator/index3.php` uses for backend popup.

\$Itemid, \$option, \$task

- \$Itemid uses on frontend only. It is a menu id (id in #__menu table).
- Provide current active menu display and assigned template and modules.
- \$option is what Joomla! uses to decide which component to load. It matches with option field in #__components table.
- \$task is use by component as controller as they switch using value.

Using \$task

```
Switch ($task) {  
    case 'new':  
        break;  
    case 'edit':  
        break;  
    case 'save':  
        break;  
    default:  
        listItems($option);  
        break;  
}
```

Design Patterns-Files

- **Frontend**

- /components

- /com_comname

- comname.php → infrastructure

- comname.html.php → view

- comname.class.php → model

- **Backend**

- /administrator

- /components

- /com_comname

- admin.comname.php

- admin.comname.html.php

- toolbar.comname.php

- toolbar.comname.html.php

- comname.xml

Global variables

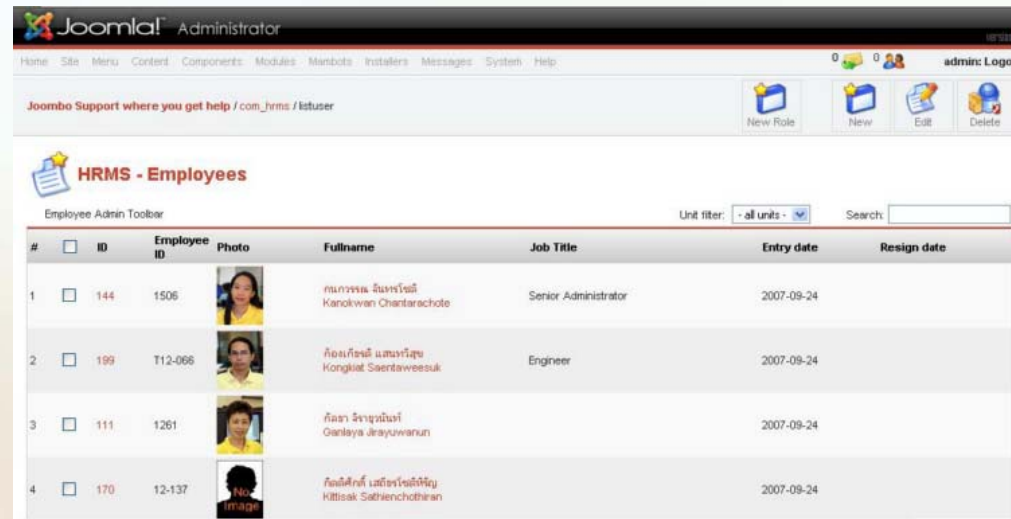
- `$database` → database connection
- `$mainframe` → main application framework
- `$my` → user using site
- `$Config_live_site = $mainframe->getCfg('live_site');`
→ Site URL
- `$Config_absolute_path = $mainframe->getCfg('absolute_path');` → absolute path of site

Widely use classes





- Database- > Database connector and helper (and global variable \$database)
- mosDBTable -> Basic article class
- mosHTML (static class)
- mosCommonHTML
- Javascript libraries
 - jsCalendar
 - Overlib
 - Custom JavaScript → \$mainframe->addCustomHeadTag()
 - Joomla JavaScript use mainly in backend.
- mosPagnivation

Component Backend

- Tasks and toolbars → toolbar.comname.php
- Draws toolbar → toolbar.comname.html.php
- Toolbar called in Admin template



The screenshot shows the Joomla! Administrator interface for the HRMS - Employees component. The page features a navigation menu at the top, a toolbar with icons for 'New Role', 'New', 'Edit', and 'Delete', and a table listing employees. The table has columns for '#', 'ID', 'Employee ID', 'Photo', 'Fullname', 'Job Title', 'Entry date', and 'Resign date'. The data is as follows:

#	ID	Employee ID	Photo	Fullname	Job Title	Entry date	Resign date
1	<input type="checkbox"/>	144	1506	 กนกวรรณ ชันลาเรชญ์ Kanokwan Chanlarachote	Senior Administrator	2007-09-24	
2	<input type="checkbox"/>	199	T12-066	 ก้องเกียรติ แสนาวีสุก Kongkiat Saertaweesak	Engineer	2007-09-24	
3	<input type="checkbox"/>	111	1261	 กานลาญาณ์ อารยวนัน Ganlaya Arayuwannun		2007-09-24	
4	<input type="checkbox"/>	170	12-137	 กิติศักดิ์ แสนชื่นฉิม Kittisak Saetuechchim		2007-09-24	

Backend Toolbar

- Both files include in admin template.
- Determine on each \$task what toolbar to display.
- Based on Joomla JavaScript, each icon has associate task.
- Not all \$task has toolbar e.g. 'save'.
- Default toolbar e.g. when entering system show list of items. Toolbar shown are 'edit',' delete'.
- On 'edit' task we show 'save', 'cancel' toolbar.

Joomla Support where you get help / com_hrms / list

 **HRMS :: Job Titles**

<input type="checkbox"/>	ID	Thai Title	English Title	Job Grade	Published
<input type="checkbox"/>	1	กรรมการผู้จัดการ	President	14	
<input type="checkbox"/>	2	ผู้อำนวยการอาวุโส	Senior Vice President	11	
<input type="checkbox"/>	3	ผู้อำนวยการ	Vice President	10	
<input type="checkbox"/>	4	ผู้ช่วยผู้อำนวยการ	Assistant Vice President	9	

Toolbar.example.php

```
require_once ($mainframe->getPath( 'toolbar_html' ) );  
Switch ($task) {  
    case 'new':  
    case 'edit':  
        TOOLBAR_example::_EDIT();  
    break;  
    default:  
        TOOLBAR_example::_DEFAULT();  
    break;  
}
```

Toolbar.example.html.php

```
Class TOOLBAR_example {
```

```
    function _EDIT() {  
        mosMenuBar::startTable();  
        mosMenuBar::save();  
        mosMenuBar::spacer();  
        mosMenuBar::endTable();  
    }
```

```
    function _DEFAULT() {  
        mosMenuBar::startTable();  
        mosMenuBar::delete();  
        mosMenuBar::spacer();  
        mosMenuBar::edit();  
        mosMenuBar::endTable();  
    }  
}
```

More on mosMunuBar

- `editList ($task='edit', $alt='Edit');`
- `custom ($task="", $icon="", $iconOver="", $alt="", $listSelect=true);`
- `media_manager ($directory="", $alt='upload');`
- Total 34 function to use.
- ***Cannot change image path for ICON.***
- Toolbar relies on Joomla! JavaScript. Includes by default in backend.

Build all functions into admin page

```
<form name="adminForm" method=POST" action=
    "index2.php?option=<?php echo $option;?>" >
```

```
<input type="hidden" name="boxchecked" value="" />
```

```
<input type="hidden" name="task" value="" />
```

```
<input type="hidden" name="hidemainmenu" value="" />
```

```
</form>
```

List view-building list and gadgets.

Check all Checkbox

```
<input type="checkbox" name="toggle" value="" onClick="checkAll  
  ( count ( <?php echo $rows; ?> ) );" />
```

Checkbox

```
$k=0;  
for ($i=0; $i < count ($rows); $i++) {  
  $row = & $rows [$i]  
  $checkbox = mosHTML::idBox($i, $row->id);  
}
```

Same as....

```
$checkbox = "<input type=\"checkbox\" id=\"cb{$i}\"  
  name=\"cid[ ]\" value=\"{$row->id}\" onclick=\"isChecked  
  (this.checked);\" />";
```

<input type="checkbox"/>	ID	Thai Title
<input type="checkbox"/>	1	กรรมการผู้จัดการ
<input type="checkbox"/>	2	ผู้อำนวยการอาวุโส
<input type="checkbox"/>	3	ผู้อำนวยการ
<input type="checkbox"/>	4	ผู้ช่วยผู้อำนวยการ
<input type="checkbox"/>	5	ผู้จัดการอาวุโส
<input type="checkbox"/>	6	ผู้จัดการฝ่ายขายอาวุโส

List view (Cont.)

- **Edit Link (To edit, click on link or Click on toolbar)**

```
$editlink = "<a href=\"#edit\" onclick=\"return  
listItemTask('cb{ $i}','edit')\">";
```

- **Display alternating row color (tr class)**

```
$k=0;
```

```
for ($i=0; $i < count ($rows); $i++) {
```

```
    $row = & $rows [$i]
```

```
    $k = 1 - $k;
```

```
    $class = "row{ $k}; // for frontend tablesectionentry{ $k}";
```

```
}
```

Getting inputs

```
$id = (int) mosGetParam ($_POST, 'id', 0);
```

```
$cid = array();
```

```
$cid = mosGetParam ($_REQUEST, 'cid', array(0));
```

```
$firstName = mosGetParam ($_GET, 'first_name', "");
```

Pagination

- Require `pageNavigation.php` in `/administrator/includes` for Backend or `/includes` for Frontend.
- Get values of `total`, `limit`, `limit start of record` before create `$pageNav` object.
- To get `limit`, `limitsatrt`, frontend uses `mosGetParam(...)`
- Backend uses `getUserStateFromRequest(...)`

Pagination (cont.)

- Includes `pageNavigation.php`.
- Get total number of records.
- `$total = asterProvHelper::totalSipAccounts($search);`
- `$pageNav = new mosPageNav($total, $limitstart, $limit);`
- `$rows = asterProvHelper::listSipAccounts($pageNav->limitstart, $pageNav->limit);`

How to filter result

AJAX and Joomla!

- Use your own way (Not recommend for AJAX libraries)
- Use mambot/plugin and AJAX libraries like XAJAX
 - Resolve AJAX conflicts between Components and Modules when using AJAX library
 - Minimize distributed include files by many components

Joomla XAJAX

- Install xajax4joomla.plugin.zip, configure and publish
- In our component create file xajax.componentname.php e.g. if component name is hrms then create xajax.hrms.php.
- You can write program normally in this file, that is you can use any Joomla variables and classes.

```
<?php
/* Stop direct access */
defined( '_VALID_MOS' ) or die( 'Restricted access' );
define( '_XAJAX_HRMS', dirname(__FILE__));
require_once($mainframe->getPath("class","com_hrms"));
$xajaxFunctions[] = array('getEmployeeInUnit',
        _XAJAX_HRMS);
if (!isset($_HRMS)) {
    $_HRMS = new hrmsSystem("FRONTEND");
}
/** XAJAX server function */
function getEmployeeInUnit($unit_id){
    $html = listEmployees ($unit_id);
    $objResponse = new xajaxResponse();
    $objResponse->addAssign("employee_list",
"innerHTML", $html);
    return $objResponse->getXML();
}
```

Where to Get More Information

- Joomla help site <http://help.joomla.org>
- Joomla! Web site <http://www.joomla.com>