

Moodle 二次开发之 block

于方军 山东省淄博市博山区山头中心学校

摘要: Moodle 是一个源代码开放的在线学习管理系统, 教师在平时的教学中可以通过添加、修改各种活动或资源, 来定制其课程形式, 实现交互式的教学过程。Moodle 是模块化的, 提供很多不同模块接口, 开发者可以贡献开发好的模块来增强 Moodle 的功能。本文介绍了如何在 Moodle 中创建基本的功能板块 block, 这是 Moodle 最基本、最常见的二次开发, 并介绍了 block 块的权限设置。

关键词: Moodle; 功能板块; 二次开发

● 创建一个 block

通过创建一个最简单的以 helloworld 命名的块来介绍如何创建一个 block, 本 block 只是用于显示文本: hello world!

首先, 在 Moodle 文件夹中的 blocks 文件夹中创建 helloworld 文件夹, 在这个文件夹中根据 Moodle 的命名规则 module-type_module-name.php, 创建 block_helloworld.php 文件。在这个文件中首先通过 block_base 基类创建 block_helloworld 类, 然后为这个类创建一个初始化函数 function init(), 在此函数中设置块标题和块版本两个值, 如下所示。

```
<?php
class block_helloworld extends block_base
{
    function init() {
        $this->title = get_string('helloworld',
'block_helloworld');
        $this->version = 2010080700;
    } //function init
```

其次, 设置 block 显示内容, 通过设置 get_content() 函数里 \$this->content;

的值来设置 block 的显示内容, 如下所示。

```
function get_content() {
    if ($this->content !== NULL) {
        return $this->content;
    }
    $this->content = new stdClass;
    $this->content->text = 'Hello World!';
    return $this->content;
} //function get_content
} //class block_helloworld
?>
```

这时, helloworld 块已经创建成功。

● 添加语言文件

同 Moodle 中的其他插件一样, block 可以用自己的语言文件来管理其显示的字符串。这些语言文件包含在一个专门的 lang 文件夹内。Moodle 使用其语言文件夹层次结构。这种设计考虑了子语言包对父语言包的继承, 子语言中未定义的任何字符串, 都将会从父语言包中提取。

在 helloworld 文件夹中建立 lang 文件夹, 再在 lang 文件夹中建立 en_utf8、zh_cn_utf8 两个子文件夹, 在每个子文件夹中创建 block_helloworld.

php 文件, Moodle 存储所有的字符串在一个哈希数组内称为 \$string。在 en_utf8 的 block_helloworld.php 文件中定义字符串的值, 如下所示。

```
<?PHP
$string['helloworld'] = 'Hello World';
?>
```

在 zh_cn_utf8 的 block_helloworld.php 文件中定义字符串的值, 如下所示。

```
<?PHP
$string['helloworld'] = '你好世界!';
?>
```

● 添加权限设置

Moodle 的用户操作管理分为五个级别的权限: 网站管理员操作权限, 课程管理员操作权限, 教师操作权限, 学生操作权限和访客操作权限。下面为新建的 helloworld 块设置相应的权限。

在 helloworld 文件夹中创建 db 文件夹, 在 db 文件夹中创建一个名为 access.php 的文件, 该文件用于定义权限。下面是一个简单的阅读权限定义的示例。

```
<?php
$string['helloworld_capabilities'] = array(
    'block/helloworld:view' => array(
```

```
'captype' => 'read',
'contextlevel' => CONTEXT_SYSTEM,
'legacy' => array(
'guest' => CAP_PREVENT,
'student' => CAP_PREVENT,
'teacher' => CAP_PREVENT,
'editingteacher' => CAP_PREVENT,
'coursecreator' => CAP_PREVENT,
'admin' => CAP_ALLOW
)
);
?>
```

通过该设置后,因为 admin 设置为“CAP_ALLOW”,所以只有管理员能看到本 block,以其他角色登录时显示的都是“没有权限”的信息。

● 开发一个实用的 block

在这里创建一个“给任课教师发送信息”的 block,创建、语言文件、权限设置与 helloworld 板块相同,只需改一下名称即可。本部分只介绍增加内容。

一个班级可能有多个任课教师,不同课程联系的教师可能不同,因此首先要给板块提供“配置”功能。添加方法“instance_allow_config()”到板块类中。

改动后,当在任何课程中打开编辑功能,就可以使 Moodle 在板块的首部显示一个“配置”按钮,但是如果单击

“配置”按钮,会得到一个提示,告知板块的配置功能不能正确执行。这个提示是因为虽然要求 Moodle 提供配置功能,但并没有告诉它需要什么样的配置功能。要再创建一个文件 /block/contactteacher/config_instance.html(文件必须完全按照此命名),复制下面的代码到文件中,并保存。

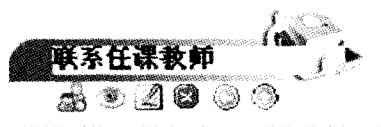
```
<?php print_string('configcontent', 'block_
contactteacher'); ?>
<tr valign="top">
<td align="right">
<?php print_string('config_
title', 'block_course_results') ?>
</td>
<td>
<input name="blocktitle"
type="text" size="24" value="<?php
p(empty($this->
config->blocktitle) ? "" : $this->config->
blocktitle)
?>" />
</td>
</tr>
<input type="submit" value="<?php
print_string('savechanges') ?>" />
<?php use_html_editor(); ?>
```

现在,板块可以配置标题内容了。如果想让它功能再强一些:实现在一

个课程中可以添加多个这种类型的板块(因为每个班有多门学科),需要再将“instance_allow_multiple()”方法添加到板块类中。

这样设置后,Moodle 就能在一门课程中添加任意数目个 contactteacher 的功能板块。注意:若板块自身允许在同一个页面中有多个实例,管理员可以再设置是否允许这种行为。在“网站管理”板块中的“插件”下的“板块”页中,即可设置是否允许板块在同一页面中有多个实例。而且,如果“instance_allow_multiple()”方法被添加到板块中,在前面已经添加到板块类中的“instance_allow_config()”方法,就不再有意义了。Moodle 认为若一个板块允许在同一页面中有多个实例,这些实例一定是要求可配置的(在同一页面中,有多个完全一样的板块实例是没有意义的),因此会自动提供“配置”按钮。所以,我们现在在板块类中移除“instance_allow_config()”方法,不会有任何影响。

现在,一个简单的 contactteacher 板块已经定义好了(如下图),当然还可以添加更多的方法,这里不再详述。^e



勇
范
君
乔
方
军
于

参考文献

- [1] Jonathan Moore, Michael Churchward. Moodle 1.9 Extension Development [M]. Published by Packt Publishing Ltd, 2010.
- [2] Moodle developer documentation [DB/OL]. http://docs.moodle.org/en/Development:Blocks/Appendix_B.
- [3] 叶海松, 季隽. Moodle 的二次开发与设计 [J]. 电化教育研究, 2007(4).
- [4] 刘平. Moodle 系统下模块开发初探 [J]. 无锡.