



RBAC 示例操作演示

ThinkPHP 2.0 更新日期：2010-09-28
中文WEB应用开发框架

2010 年

上海顶想信息科技有限公司

目录

RBAC 示例操作演示	3
理论介绍	3
准备工作	3
感受一下	5
角色管理	7
节点管理	12

1 RBAC 示例操作演示

1.1 理论介绍

基于角色的访问控制模型：

基于角色的访问控制模型（RBAC Model，Role-based Access Model）：RBAC 模型的基本思想是将访问许可权分配给一定的角色，用户通过饰演不同的角色获得角色所拥有的访问许可权。这是因为在很多实际应用中，用户并不是可以访问的客体信息资源的所有者（这些信息属于企业或公司），这样的话，访问控制应该基于员工的职务而不是基于员工在哪个组或是谁信息的所有者，即访问控制是由各个用户在部门中所担任的角色来确定的，例如：一个学校可以有教工、老师、学生和其他管理人员等角色。

1.2 准备工作

下载最新的 TP 版本

网站下载：<http://www.thinkphp.cn/Down/download/131>

SVN 下载：<http://thinkphp.googlecode.com/svn/trunk/>

导入示例数据库

导入前可以先查看一下数据库文件 Examples\examples.sql（如图 1），sql 中没有创建数据库的语句，所以在导入 sql 文件之前，先自行创建一个名为`demo`的数据库，字符集选择

utf8_general_ci。

```
16  /*!40101 SET NAMES utf8 */;  
17  
18  --  
19  -- 数据库: `demo`  
20  --
```

图 1

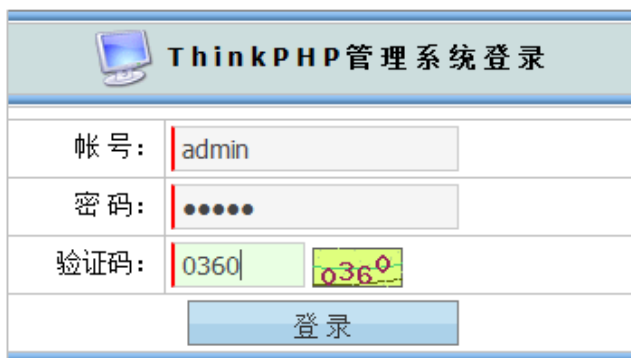
说明：

如果你的数据库名以及用户名密码不同，请重新修改一下/config.php 配置文件，当然对于已经生产

RUNTIME 目录的示例，需要删除一下缓存文件的。

1.3 感受一下

可以同时开多个不同浏览器，对不同的角色进行登录查看一下。默认情况下，只有 admin 才能进入管理后台，其他的账号均没有权限进入。



The image shows a login form for the 'ThinkPHP 管理系统' (ThinkPHP Management System). The form has a title bar with a computer icon and the text 'ThinkPHP 管理系统 登录'. Below the title bar, there are three input fields: '帐号:' (Username) with the value 'admin', '密码:' (Password) with masked dots, and '验证码:' (Captcha) with the value '0360'. To the right of the captcha input is a small image of the captcha '0360'. At the bottom of the form is a blue button labeled '登录' (Login).

管理员：admin/admin 领导：leader/leader 员工：member/member 演示：demo/demo

图 2

我们以超级管理员 admin 进入后台以后，可以进行任何的操作。头部有“后台首页”和“应用中心”两个选项按钮；其中“应用中心”又包括“数据管理”、“节点管理”、“角色管理”和“后台用户”四个功能模块。

我们先在“后台用户”中，把用户的昵称改一下，以区别“角色”里面的角色名（如图 3）。



图 3

1.4 角色管理

默认状态下我们三个角色分组，分别是：演示组、员工组、领导组。当然我们也可以添加新的组别，或者修改组名；这些操作不是很难，就不赘述了，这里还是着重讲一下如何来授权。



图 4

下面比如我们要对用户“张三”来授权访问我们的后台；RBAC 是针对一种角色来授权，所以我们要先将用户添加到某个“角色组”中，再来对该角色授权，最后拥有这个“角色”的“用户”，就拥有了该“角色”的权限。

A. 用户列表

点击“用户列表”，添加“张三”到当前组中，这里我们将后台用户“张三”，添加到“演示组”中，勾选好以后，点击“保存”按钮即可。

组用户列表 [返 回]

当前组: 演示组

<input type="checkbox"/>	admin 管理员
<input checked="" type="checkbox"/>	demo 张三
<input type="checkbox"/>	member 李四
<input type="checkbox"/>	leader 王五

全选 反选 全否 保存

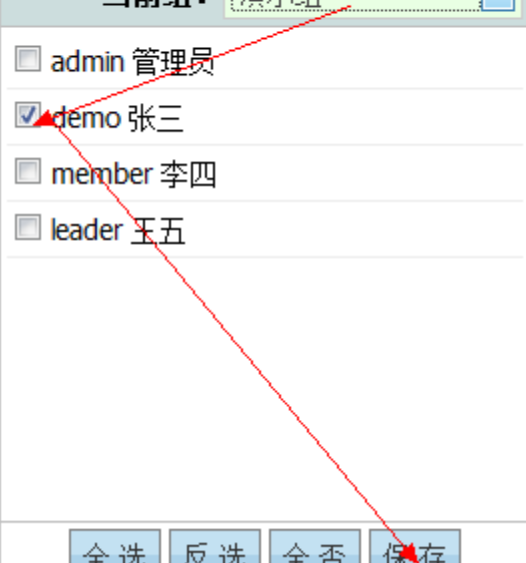


图 5

现在后台用户“张三”拥有了“演示组”的角色，也就拥有了“演示组”所具有的一些默认的权限；现在用“张三”的账号来登录一下后台（我们这里是 demo），已经可以正常访问后台了。

下面图 6、图 7、图 8 就是我们“演示组”所具有的默认权限。



图 6



图 7

这里我们默认就拥有了“数据管理”模块的权限，所以“张三”登录后，可以在左侧看到“数据管理”这个功能模块的操作链接。

应用授权 模块授权 操作授权	
当前组:	演示组
当前应用:	Rbac后台管理
当前模块:	公共模块
<input checked="" type="checkbox"/>	查看
<input checked="" type="checkbox"/>	列表
<input type="checkbox"/>	恢复
<input type="checkbox"/>	禁用
<input type="checkbox"/>	删除
<input type="checkbox"/>	更新
<input type="checkbox"/>	编辑
<input type="checkbox"/>	写入

图 8

如果我们想对该角色组增加一些权限，拥有更多的操作，我们便可以如下操作：

1. 对演示组增加“恢复”，“禁用”，“新增或写入”，“编辑”、“更新”等操作；

如图 8，在“操作授权”下，勾选相应的权限选项列表，保存即可。

2. 增加“演示组”具有模块“后台用户”管理的功能

如图 7，在“模块授权下”，勾选“后台用户”选项，保存即可。

此时我们再用“张三”账号进行登录，发现左侧已经多了一个功能模块，如图 9；也具有了上面所有的操作权限，比如“新增”和“编辑”等，“删除”还是没有权限的，这是正确的，因为我们没有分配“演示组”具有该权限。



图 9

注意：角色组的权限发生改变后，后台用户要退出后再登录，新的权限才能生效，因为权限列表是存储在\$_SESSION 中的；

其他的一些“用户组”以及“后台用户”，模块的权限以及各模块下的具体操作的权限分配，操作步骤同上，大家都可以试一试。

注意：RBAC 的授权时针对“角色”（即上面提到的“用户组”），而非具体的某一个“后台用户”，同一个“后台用户”，可以同时拥有多个“角色”的权限；只要在“角色组”的“用户列表”中，选择相应的用户即可。

我们也可以换一条思路（其实本人认为这样操作更好）：

1. 先把各个“角色”建立好
2. 然后对各个“角色”进行权限的划分
3. 最后再在各个“角色”中，“拖”些“人”进来；

然后某些人就拥有了好几种角色，即拥有了多个角色所拥有的权限了；而有些“平民百姓”可能只能看看特定的模块特定的页面了。

1.5 节点管理

“数据管理”和“后台用户”的操作，没什么特别的；加上上面对“角色管理”的讲解，基本上 RBAC 你已经掌握了 60%了。

现在我们就拿下最后的 40%，加油！

了解一下

这里的 Rbac (如图 10) 对应于该示例的目录名，以及入口文件中的 APP_NAME，如果你的管理后台的“项目名”或者“项目分组”为“Admin”，则要将此名称改成对应的即可。

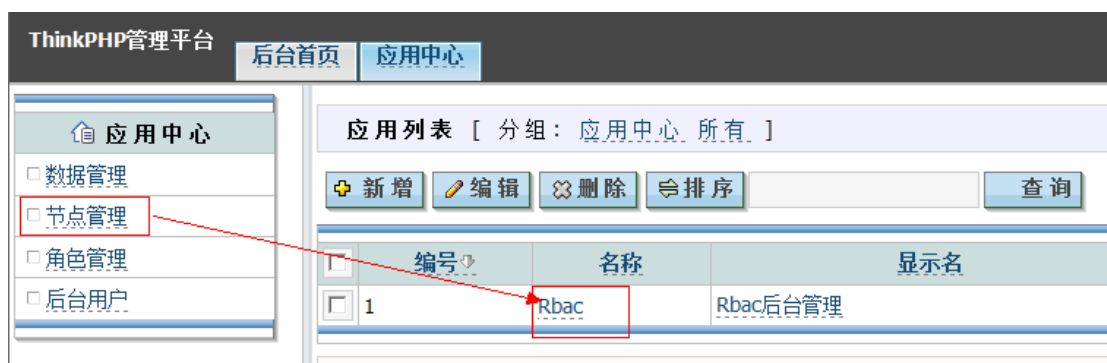


图 10

图 11，是我本地的修改测试。后面的讲解我还是会继续以 Rbac 来做演示。



图 11

点击 Rbac，如图 10 的箭头所指，进入各个模块的页面，如图 12；

[R b a c] 模块列表 [分组：应用中心 所有]				
新增 编辑 删除 排序 查询				
<input type="checkbox"/>	编号	名称	显示名	
<input type="checkbox"/>	69	Form	数据管理	应用中心
<input type="checkbox"/>	40	Index	默认模块	未分组
<input type="checkbox"/>	30	Public	公共模块	未分组
<input type="checkbox"/>	7	User	后台用户	应用中心
<input type="checkbox"/>	6	Role	角色管理	应用中心
<input type="checkbox"/>	2	Node	节点管理	应用中心

图 12

点击各个模块的名称，我们会发现有的下面还会有一些具体的操作列表，如图 13；而有的模块下面却没有任何的操作；

[P u b l i c] 操作列表 [分组：应用中心 所有]			
新增 编辑 删除 排序 查询			
<input type="checkbox"/>	编号	名称	显示名
<input type="checkbox"/>	49	read	查看
<input type="checkbox"/>	39	index	列表
<input type="checkbox"/>	37	resume	恢复
<input type="checkbox"/>	36	forbid	禁用
<input type="checkbox"/>	35	foreverdelete	删除
<input type="checkbox"/>	34	update	更新
<input type="checkbox"/>	33	edit	编辑
<input type="checkbox"/>	32	insert	写入
<input type="checkbox"/>	31	add	新增

图 13

打开项目文件/Rbac/Lib/CommonAction.class.php，该文件就是公共模块 Public 了，其下包括的操作列表都是对应于该文件代码中的各个方法名。而其他的 Action 文件，例如 FormAction 类就是继承于 CommonAction 类，所以上面对于公共模块的操作的授权，只在公共模块的“操作授权”里面作相应的勾选即可。

新任务

比如现在我们要在“数据管理”中，新增数据的时候，希望可以上传附件。然后只有领导组具有上传操作权限，演示组是不具备该权限的。我们可以如下操作：

1. 在 FormAction.class.php 文件中写好相应的模板显示方法和上传处理方法。具体实现

方法这里省略；可参考相应的示例。如图 14；

```
// 这里是上传页面显示
function upload_file() {
    $this->display();
}
// 这里是处理上传的方法
function upload_file_op() {
    //
}
```

图 14

2. 在对应的模板/Rbac/Tpl/default/Form/目录中，对相应的模板进行制作和更改。下图

是我再 add.html 中新增的一个按钮，再制作一个附件上传的页面，取名 upload_file.html。

添加数据

[[返回列表](#)]<

标题:

附件:

上传

内容:

代码

🔍

✖️

↶

↷

🖨️

✂️

📄

📁

T

💻

☰

☳

☵

☶

☱

☲

I 标题

A+ 字体

A- 大小

字

字

@

@

@

S

🔄

📑

🖼️

🔗

🏠

✎

图 15

3. 新增节点：在节点管理中，点击 Rbac->Form 模块，然后进入 Form 模块的操作列表；
- 如图 16；此时我们需要增加两个节点（对应于上面上传附件的两个方法或两个操作）。填写好对应的表单，保存即可，如图 17；新增好以后，再来查看 Form 模块的操作列表，如图 18 为正常。



图 16

新增应用 [返回列表]

应用名:

upload_file

显示名:

上传附件

分组:

应用中心

状态:

启用

描述:

显示附件上传的页面

保存

清空

图 17

[Form] 操作列表 [分组: 应用中心 所有]

+ 新增

编辑

删除

排序

查询

<input type="checkbox"/>	编号	名称	显示
<input type="checkbox"/>	84	upload_file_op	上传附件处理
<input type="checkbox"/>	83	upload_file	上传附件

图 18

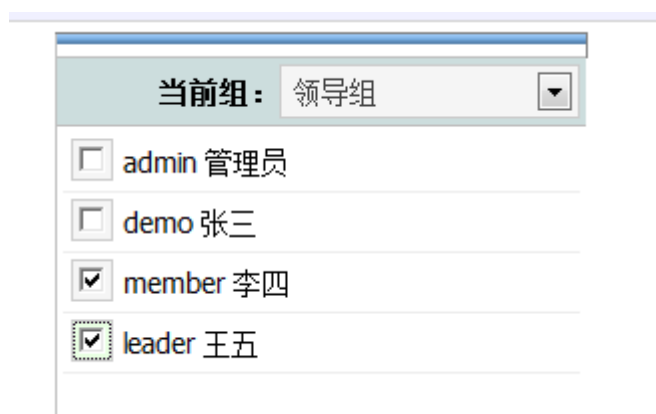
4. 分配权限：进入“角色管理”，先对“领导组”进行授权，如图 19；



应用授权 模块授权 操作授权	
当前组:	领导组
当前应用:	Rbac后台管理
当前模块:	数据管理
<input checked="" type="checkbox"/>	上传附件
<input checked="" type="checkbox"/>	上传附件处理

图 19

5. 拉人：点击“用户列表”，在领导组中，勾选“后台用户”账号，保存即可，如图 20；



当前组: 领导组	
<input type="checkbox"/>	admin 管理员
<input type="checkbox"/>	demo 张三
<input checked="" type="checkbox"/>	member 李四
<input checked="" type="checkbox"/>	leader 王五

图 20

6. 测试：现在我们分别用账号 leader 和 demo 来测试一下。正常的授权操作后，leader 有权限，而 demo 没有权限。

模块的授权

最复杂的“操作授权”已经被我们搞定了，现在剩下“模块的授权”。操作步骤基本同上。

并且没有具体到某一个模块下的具体方法的授权操作，针对整个模块功能操作的一个授权。

我们可以如下操作：

1. 创建功能模块文件 `XyzAction.class.php`，写入相应的方法，创建对应的模板文件；
2. 新增节点：“节点管理”->“新增模块”；如图 21；退出，重新登录，发现左侧“应用中心”多了一个模块链接，如图 22；



新增 模块 [返回列表]

模块名:

显示名:

分组:

状态:

描述:

图 21



图 22

3. 授权：模块授权；如图 23

应用授权 | 模块授权 | 操作授权

当前组: 员工组

当前应用: Rbac后台管理

☒ 默认模块

☒ 公共模块

☒ 数据管理

☐ 后台用户

☐ 角色管理

☐ 节点管理

☒ Xyz模块

图 23

4. 拉人：点击“用户列表”，在员工组中（对应上面的授权操作，这里只是方便测试），勾选“后台用户”账号，保存即可，如图 23；

当前组: 员工组

☐ admin 管理员

☐ demo 张三

☒ member 李四

☐ leader 王五

图 24

5. 测试：现在我们分别用演示组账号 demo 和 员工组的账号 member 来测试一下。成功的授权操作后，member 李四有权限，而 demo 张三左侧都不会显示“Xyz 模块”的链接，显然是没有权限的。

现在我们已拿下了 Rbac 的操作了.....K.O！赶快去项目中试试吧~~

ThinkPHP 文档小组 2010-9-28

[HTTP://ThinkPHP.CN](http://ThinkPHP.CN) | WEB 应用开发最佳实践框架

大道至简，开发由我 WE CAN DO IT , JUST THINK