Applied Mechanics and Materials *Online: 2013-10-15*
ISSN: 1662-7482, Vols. 433-435, pp 1603-1608
doi:10.4028/www.scientific.net/AMM.433-435.1603
© 2013 Trans Tech Publications, Switzerland

The Implementation of Rights Management of Network Teaching Platform with Role-Based Access Control

Guiying HAN [1,a] , Xizuo LI[2,b]

[1]School of Information and Communication, Dalian Nationalities University, Dalian, 16600,China

[2]School of Computer Science and Technology, Dalian Nationalities University, Dalian, 16600,China

[a]guiyinghan@126.com, [b]lixizuo@163.com

Keywords: Role-Based Access Control(RBAC), Network Teaching Platform, Rights Management

Abstract. For the system security risks of multiple administrators to manage the network teaching platform, role-based access control(RBAC) from the ThinkPHP framework is used to implement the rights management of the network teaching platform. Teachers or administrators access and can only access the related function modules or methods authorized by the super administrator. It facilitates the rights management of the network teaching platform and makes a clear division of works for the multiple administrators and more scientific management of the platform. Practice shows that RBAC-based rights management for the network teaching platform is simple, practical and has a good application value.

Introduction

With the constant improvement of network teaching platform, the user capacity is also increasing. Generally, the network teaching system involves many courses and a number of teachers or administrators, so only one administrator can not meet the needs of the development of the platform. But, multiple background administrators to manage the platform will also inevitably involve security issues and even teaching experiment or the fairness of subject contests have been affected. Teachers or administrators can access and can only access the relevant functional modules authorized by their own super administrator (but super administrator is not restricted, and their assigned permission is invalid, thus preventing the super administrator can not access the background due to operational errors and other causes). It not only facilitates the management of the administrator, while multiple administrators can have a clear division of works, so that the background management is more scientific. Therefore, in order to meet the development needs of network teaching platform, meanwhile taking into account the fairness and safety for the system, an role-based access control for the rights management of the network teaching system is very necessary.

All rights reserved. No part of contents of this paper may be reproduced or transmitted in any form or by any means without the written permission of Trans Tech Publications, www.ttp.net. (#88066385, ProQuest, USA-26/04/17,16:07:21)

The Process of Rights Management of Network Teaching Platform

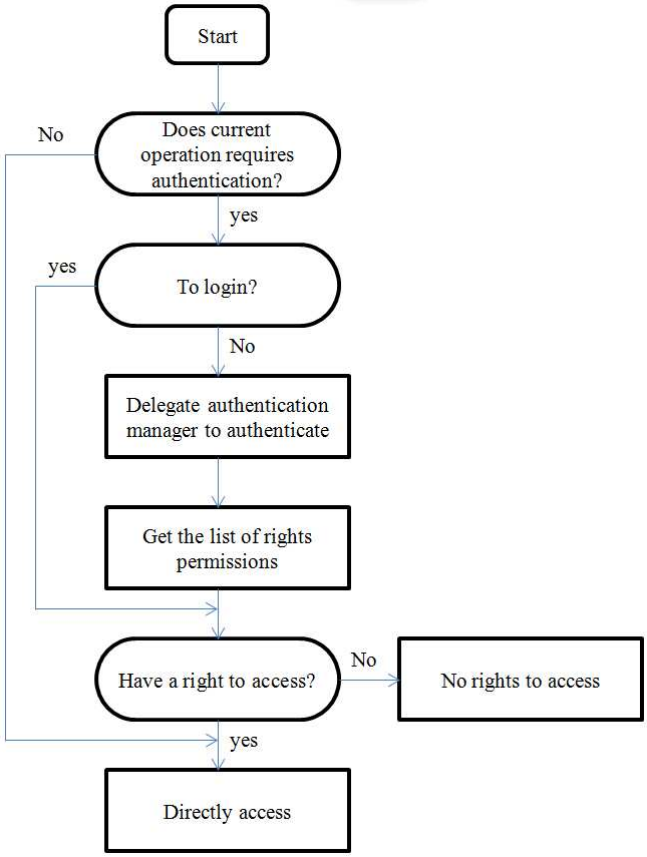

Fig. 1 The Process of Rights Management of Network Teaching Platform

Modular Design for Rights Management

The modular design for rights management is divided into two parts: The one is administrator module that is further divided into adding an administrator, viewing administrator's rights and deleting an administrator; the another is the "rights management" module that can be used to display a list of all administrators' rights and also edit them. Selecting an authorized administrator and entering the edit page, if the administrator has not been assigned to permissions, all the available modules and functions can be checked to allow access and click the submit button to complete the editing; if the administrator has been assigned for the privileges, all the modules and functions which has been authorized to access will automatically be checked and also you can continue to check or uncheck the checked module in order to increase or decrease the administrator's access privileges. The rights management module is not permitted to assign permissions to any other administrator.

Database Design

For the access control of the modules and functions of the network teaching platform, 5 tables are designed, that includes the user table used to store background administrator related information(shown as table 1), the node table used to store all modules and functions of the module of the platform(shown as table 2), the role table used to store roles (or user group) related information(shown as table 3), the role and user table used to store the relations between roles (or user group) and the corresponding users(shown as table 4), the permissions table used to store all the resources that roles can access(shown as Table 5).

www.manaraa.com

Table 1: Users table

Fields	Types	Attributes	Is Null	Default	Extra	Annotation
id	smallint(6)	UNSIGNED	No		auto_increment	Number
account	varchar(64)		No			User name
password	char(32)		No			Password
name	varchar(50)		No	NULL		nickname

Table 2: Nodes table

Fields	Types	Attributes	Is Null	Default	Extra	Annotation
id	smallint(6)	UNSIGNED	No		auto_increment	Number
name	varchar(20)		No			Node name
title	varchar(50)		Yes	NULL		Node topic
status	tinyint(1)		Yes	NULL		State
remark	varchar(255)		Yes	NULL		Tag
sort	smallint(6)	UNSIGNED	Yes	NULL		Classification
pid	smallint(6)	UNSIGNED	No			Parent node
level	tinyint(1)	UNSIGNED	No			Level

Table 3: Roles table

Fields	Types	Attributes	Is Null	Default	Extra	Annotation
id	smallint(6)	UNSIGNED	No		auto_increment	Number
name	varchar(20)		No			Role name
pid	smallint(6)		Yes	NULL		Parent ID
status	tinyint(1)	UNSIGNED	Yes	NULL		State
remark	varchar(255)		Yes	NULL		Tag

Table 4: Roles and users table

Fields	Types	Attributes	Is Null	Default	Annotation
role_id	mediumint(9)	UNSIGNED	Yes	NULL	Role ID
user_id	char(32)		Yes	NULL	User ID

Table 5: Permissions table

Fields	Types	Attributes	Is Null	Default	Annotation
role_id	smallint(6)	UNSIGNED	No		Role ID
node_id	smallint(6)	UNSIGNED	No		Node ID
level	tinyint(1)		No		Level
module	varchar(50)		Yes	NULL	Module
pid	int(11)		No		Parent ID

The Implement of Function Modules

The system is built using RBAC class from the ThinkPHP framework to implement rights management of the platform. The main functions of the RBAC class and its implementations are as follows:

www.manaraa.com

authenticate() is used to return the user's information by introducing the query user's conditions and user table. The specific codes are as follows:

```
static public function authenticate($map,$model='') {
    if(empty($model)) $model = C('USER_AUTH_MODEL');
    //Using the given map for certification
    return M($model)->where($map)->find();
}
```

saveAccessList () is used to set the $_SESSION ['_ACCESS_LIST'] values, which contain all the nodes that corresponds to the authorized rights for the user groups. Specific codes are as follows:

```
static function saveAccessList($authId=null) {
    if(null==$authId)
    $authId = $_SESSION[C('USER_AUTH_KEY')];
    // If the normal privileges mode is used, the current user's access permissions list is saved.
    // Open all permissions for administrators
   if(C('USER_AUTH_TYPE')!=2&& !$_SESSION[C('ADMIN_AUTH_KEY')] )
        $_SESSION['_ACCESS_LIST']=    RBAC::getAccessList($authId);
    return ;
}
```

checkAccess () is used to detect whether authentication is required for the current module and operations, and return bool type. Specific codes are as follows:

```
static function checkAccess() {
    // If the project requires authentication, and the current module requires authentication,
certification for permission is conducted.
    if( C('USER_AUTH_ON') ){
        $_module = array();
        $_action = array();
        if("" != C('REQUIRE_AUTH_MODULE')) {
            // The module is required to authenticated
            $_module['yes'] =      explode(',',strtoupper(C('REQUIRE_AUTH_MODULE')));
        }else {
            //The module is not required to authenticated
            $_module['no'] = explode(',',strtoupper(C('NOT_AUTH_MODULE')));
        }
        //To check whether authentication is required for the current module
        if((!empty($_module['no'])
&& !in_array(strtoupper(MODULE_NAME),$_module['no'])) || (!empty($_module['yes']) &&
in_array(strtoupper(MODULE_NAME),$_module['yes']))) {
            if("" != C('REQUIRE_AUTH_ACTION')) {
                // The operation is required to authenticated
                $_action['yes'] =    explode(',',strtoupper(C('REQUIRE_AUTH_ACTION')));
            }else {
                //The operation is not required to authenticated
                $_action['no'] =     explode(',',strtoupper(C('NOT_AUTH_ACTION')));
            }
            //To check whether authentication is required for the current operation
            if((!empty($_action['no'])&& !in_array(strtoupper(ACTION_NAME),$_action['no'])) ||
```

www.manaraa.com

```
(!empty($_action['yes']) &&in_array(strtoupper(ACTION_NAME),$_action['yes'])) {
                    return true;
                }else {
                    return false;
                }
            }else {
                return false;
            }
        }
        return false;
    }
```

checkLogin () is used to detect whether to login. Specific codes are as follows:

```
    static public function checkLogin() {
        //To check whether authentication is required for the current operation
        if(RBAC::checkAccess()) {
            // To check the identification number of certification
            if(!$_SESSION[C('USER_AUTH_KEY')]) {
                if(C('GUEST_AUTH_ON')) {
                    //To open anonymous authorized access
                    if(!isset($_SESSION['_ACCESS_LIST']))
                        //To save anonymous access rights
                        RBAC::saveAccessList(C('GUEST_AUTH_ID'));
                }else{
                    // To ban anonymous access rights and jump to authentication gateway
                    redirect(PHP_FILE.C('USER_AUTH_GATEWAY'));
                }
            }
        }
        return true;
    }
```

accessDecision() is used to detect whether the ['current project'] ['current module'] ['current operation'] exists in the $ _SESSION ['_ACCESS_LIST'] array. If there is, which means you have permissions, otherwise it returns flase.

getAccessList() is used to return the permissions list $_SESSION['_ACCESS_LIST'] by querying the permissions table.

The implement of the process for the rights management is as follows:

First, the project names, all the module names and all the functions of the corresponding module are stored as the nodes into the nodes table. When a new administrator is created, the administrator's information is stored into the uses table. Meanwhile, the same role name (user group name)with the administrator's user name is created in the roles table, and the newly created administrator's ID and the role's ID are added into the table of the roles and users in order to make them correspond to each other. All the resources are stored in the permissions table that the roles(user groups) can access. Apparently, the newly added administrator must have the right to access the project, so an item of the administrator and the project is needed to insert into the permissions table in order to ensure that administrators can access the background.

www.manaraa.com

When the super administrator assigns permissions to an administrator, all the accessible modules are selected and then these modules and corresponding functions are found and inserted into the permissions table. Thus, the class RBAC under the framework of ThinkPHP can read the permissions table and decide the rights to access for the administrators. Finally, the rights to access to the modules and functions are assigned reasonably to administrators and the restriction of access for the administrators is implemented.

Conclusions

For the network teaching platform, if multiple administrators manages the system at the same time, it may result in security problems and unfair competition issues. In this paper, the RBAC from ThinkPHP framework is used to implement rights management for the network teaching platform. Thus, a teacher or administrator can access and can only access their own relevant functional modules authorized by super administrator, which not only facilitates the management of the administrators, meanwhile multiple administrators can have a clear division of works, so that the system management is more scientific. Practice shows that the use of the RBAC for the rights management of the platform is simple, practical and has a good application value.

Acknowledgements

This work was financially supported by the Independent Scientific Research Foundation for Central Universities(0913-130471) and 2012 Higher Education Undergraduate Teaching Reform Project of Liaoning Province of China(0406).

References

[1]Q.W.Yang, F.Hong, M.X.Yang, X.Zhu: Journal of Software, Vol.17, No.8(2006), pp.1804−1810(in Chinese)

[2]F.Lu,Z.K.Yang: Chinese Education Informatization, 2007.02, pp.49-51(in Chinese)

[3]L,Zhou, L.Pan: Journal of ShangHai JiaoTong University, Vol. 44 No. 9(2010), pp.1192-1196(in Chinese)

[4]H.T. Zhang, Z.F.Liu, Y.Li, J.H.Wang, W.T.Yang: Microcomputer Information, Vol.22 No.9-3(2006),pp.29-3,96(in Chinese)

[5]Baidu Encyclopedia: http://baike.baidu.com/view/908071.htm

www.manaraa.com

Reproduced with permission of copyright owner.
Further reproduction prohibited without permission.

www.manaraa.com