

第十二课 Yii2 权限管理 RBAC

主讲老师: Smister

1、前言

Rbac 的主要核心文件是放在 yii2-demo\vendor\yiisoft\yii2\rbac 文件夹中，其中也包含了数据库表的创建信息，放在 migrations 文件夹中，可以进行导入数据库。Rbac 分为 4 种类别，分别为 Permission、Role、User、Rule，一个 User 可以有多个 Role，一个 Role 有多个 Permission，Rule 则是辅助 Permission 和 Role 的辅助验证。

2、权限数据表之间的关系

yii2-demo\vendor\yiisoft\yii2\rbac\migrations\schema-mysql.sql 一共有 4 张表，分别为 auth_assignment, auth_item_child, auth_item, auth_rule。

auth_item：存放 Role 和 Permission，区别为 type 字段，1 为 Role，2 为 Permission

auth_item_child：Role 拥有的 Permission，一个 Role 多个 Permission

auth_assignment：用户拥有的 Role，一个用户对应多个 Role

auth_rule：单独存放 Rule，跟 auth_item 的 rule_name 对应

3、配置 Rbac 权限的 config/web.php

```
<?php
    return [
        //..
        'components' => [
            'authManager' => [
                'class' => 'yii\rbac\DbManager',
            ]
        ],
        //..
    ];
?>
```

4、Permission 操作

```
<?php
    $auth = \Yii::$app->authManager; //直接通过 Yii::$app 调用 Components

    //添加 Permission
    //createPermission('permission 的名称');
    $perm = $auth->createPermission('test-add');
    $perm->description = 'test add operate ..';
    $auth->add($perm);
```

```
//读取一个 Permission
//getPermission('Permission 的名称');
$onePerm = $auth -> getPermission('test-add');
```

```
//读取全部 Permission
$allPerm = $auth -> getPermissions();
```

//这里需要注意一点，因为 auth_item 表跟 auth_item_child, auth_assignment 之间是关联，只要这里的数据变了关联的表之间的数据也是会自动变化的

```
//更新一个 Permission
$permName = 'test-add';
$perm = $auth -> getPermission($permName);
$perm -> description = 'i am a new description';
$auth->update($permName, $perm);
```

```
//删除一个 Permission
$perm = $auth -> getPermission('test-add');
$auth->remove($perm);
```

```
//删除所有所有 Permission
$auth -> removeAllPermissions();
```

?>

5、Role 操作

```
<?php
$auth = \Yii::$app->authManager;

//添加一个 Role
//createRole('role 的名称')
$role = $auth->createRole('super');
$role -> description = 'super manager';
$auth -> add($role);

//读取一个 Role
//getRole('Role 的名称')
$oneRole = $auth->getRole('super');

//读取全部 Role
$allRole = $auth->getRoles();
```

```

//更新 Role
$roleName = 'super';
$oneRole = $auth->getRole($roleName);
$oneRole -> description = 'update Role';
$auth -> update($roleName , $oneRole);

//删除 Role 的时候 , auth_item_child , auth_assignment 的数据也会被删除
//删除一个 Role
$oneRole = $auth->getRole('super');
$auth->remove($oneRole);

//删除所有 Role
$auth->removeAllRoles();

//将一个 Permission 添加到 Role 角色里面进去
$oneRole = $auth->getRole('super'); //读取一个角色
$onePermission = $auth->getPermission('test-add'); //读取一个权限
$auth->addChild($oneRole , $onePermission);

//读取 Role 里面的所有 Permission
$roleName = 'super';
$auth->getPermissionsByRole($roleName);

//判断 Role 里面是否有 Permission
$oneRole = $auth->getRole('super'); //读取一个角色
$onePermission = $auth->getPermission('test-add'); //读取一个权限
$auth->hasChild($oneRole , $onePermission);

//将一个 Permission 从 Role 角色中移除出来
$oneRole = $auth->getRole('super'); //读取一个角色
$onePermission = $auth->getPermission('test-add'); //读取一个权限
$auth->removeChild($oneRole , $onePermission);

//将 Role 里面 Permission 全部移除
$oneRole = $auth->getRole('super');
$auth->removeChildren($oneRole);
?>

```

6、User 操作

```

<?php
$auth = \Yii::$app->authManger;
//给 id 为 1 的用户 , 赋予 super 的角色
$oneRole = $auth->getRole('super');
$auth->assign( $oneRole , 1 );

```

```

//将 super 的角色从 id 为 1 的用户移除
$oneRole = $auth->getRole('super');
$auth->revoke($oneRole , 1);

//移除 id 为 1 用户的所有角色
$auth->revokeAll(1);

//读取 id 为 1 用户所拥有的 Permission
$auth->getPermissionsByUser(1);

//检查 id 为 1 的用户是否有 Permission 的 test-add
$auth->checkAccess(1 , 'test-add');

?>

```

7、Rule 操作

Rule 其实就是 Permission 或者 Role 的附加验证功能，例如你可以修改文章，但是只能更改自己的文章。这个时候需要附加的 Rule 进行验证

① 写一个 Rule 规则

首先在 yii2-demo 新建一个 rbac 的文件夹，然后新建一个 TestRule.php 文件。

```

<?php
namespace app\rbac;
class TestRule extends \yii\rbac\Rule{
    public $name = 'testRule'; //name 是必须的 ,也是 Rule 的名称

    //这里规定 3 个参数
    // $user_id 用户的 id
    // $item Role 或者 Permission
    // $params 传进来的数据
    //如果文章不属于自己的则没有权限修改
    public function execute($user_id , $item , $params ){
        return isset($params['article']) ? $params['article']['uid'] == $user_id : false;
    }
}

?>

```

②将 Rule 添加到 Permission 中

```

<?php
$auth = \Yii::$app->authManager;

//添加一个 Rule
$testRule = new \app\rbac\TestRule();
$auth->add($testRule);

```

```
//读取一个 Rule
$ruleName='testRule';
$auth->getRule($ruleName);

//读取所有 Rule
$auth->getRules();

//给一个 Permission 添加一个 Rule
//这里需要 rule_name 就直接更新掉或者移除掉整个 Rule
$onePerm = $auth->getPermission('test-add');
$onePerm->rule_name = 'testRule';
$auth->update($onePerm->name , $onePerm);

//验证 test-add 规则
$auth -> checkAccess(1 , 'test-add' , ['article' => ['uid' => 1]]);

//删除一个 Rule
$ruleName='testRule';
$oneRule = $auth->getRule($ruleName);
$auth->remove($oneRule);

//删除所有 Rule
$auth->removeAllRules();
```

?>

更多请参阅:

<http://www.yiiframework.com/doc-2.0/guide-security-authorization.html>

<http://www.yiiframework.com/doc-2.0/yii-rbac-dbmanager.html>

感谢收看本次教程!

<http://www.smsiter.com>