# 授权

授权是指验证用户是否允许做某件事的过程。Yii提供两种授权方法： 存取控制过滤器（ACF）和基于角色的存取控制（RBAC）。

# 存取控制过滤器

存取控制过滤器（ACF）是一种通过 [yii\filters\AccessControl](https://www.yiichina.com/doc/api/2.0/yii-filters-accesscontrol) 类来实现的简单授权方法， 非常适用于仅需要简单的存取控制的应用。

正如其名称所指，ACF 是一种动作过滤器 [filter](https://www.yiichina.com/doc/guide/2.0/structure-filters)，可在控制器或者模块中使用。当一个用户请求一个动作时， ACF会检查 [access rules](https://www.yiichina.com/doc/api/2.0/yii-filters-accesscontrol" \l "$rules-detail) 列表，判断该用户是否允许执 行所请求的动作。

下述代码展示如何在 site 控制器中使用 ACF：



上面的代码中 ACF 以行为 (behavior) 的形式附加到 site 控制器。 这就是很典型的使用行动过滤器的方法。

基本上只能指定哪个方法是登录用户访问，哪个是未登录用户访问。

再加上matchCallback的匿名函数可以实现自定义的验证逻辑。

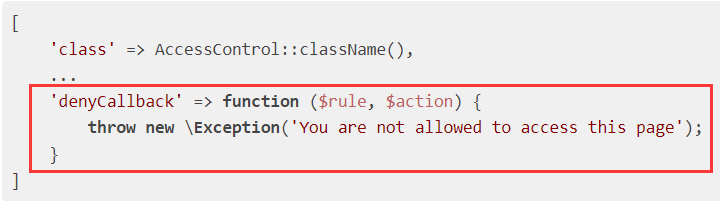
整体非常简单。

## 未获取授权的处理

当 ACF 判定一个用户没有获得执行当前动作的授权时，它的默认处理是：

* 如果该用户是访客，将调用 [yii\web\User::loginRequired()](https://www.yiichina.com/doc/api/2.0/yii-web-user" \l "loginRequired()-detail) 将用户的浏览器重定向到登录页面。
* 如果该用户是已认证用户，将抛出一个 [yii\web\ForbiddenHttpException](https://www.yiichina.com/doc/api/2.0/yii-web-forbiddenhttpexception) 异常。

你可以通过配置 [yii\filters\AccessControl::$denyCallback](https://www.yiichina.com/doc/api/2.0/yii-filters-accesscontrol" \l "$denyCallback-detail) 属性定制该行为：



## [Access rules](https://www.yiichina.com/doc/api/2.0/yii-filters-accessrule) 选项

* [allow](https://www.yiichina.com/doc/api/2.0/yii-filters-accessrule" \l "$allow-detail)： 指定该规则是 "允许" 还是 "拒绝" 。（译者注：true是允许，false是拒绝）
* [actions](https://www.yiichina.com/doc/api/2.0/yii-filters-accessrule" \l "$actions-detail)：指定该规则用于匹配哪些动作。 它的值应该是动作方法的ID数组。匹配比较是大小写敏感的。如果该选项为空，或者不使用该选项， 意味着当前规则适用于所有的动作。
* [controllers](https://www.yiichina.com/doc/api/2.0/yii-filters-accessrule" \l "$controllers-detail)：指定该规则用于匹配哪些控制器。 它的值应为控制器ID数组。匹配比较是大小写敏感的。如果该选项为空，或者不使用该选项， 则意味着当前规则适用于所有的动作。（译者注：这个选项一般是在控制器的自定义父类中使用才有意义）
* [roles](https://www.yiichina.com/doc/api/2.0/yii-filters-accessrule" \l "$roles-detail)：指定该规则用于匹配哪些用户角色。 系统自带两个特殊的角色，通过 [yii\web\User::$isGuest](https://www.yiichina.com/doc/api/2.0/yii-web-user" \l "$isGuest-detail) 来判断：
  + ?： 用于匹配访客用户 （未经认证）
  + @： 用于匹配已认证用户

使用其他角色名时，将触发调用 [yii\web\User::can()](https://www.yiichina.com/doc/api/2.0/yii-web-user" \l "can()-detail)，这时要求 RBAC 的支持 （在下一节中阐述）。 如果该选项为空或者不使用该选项，意味着该规则适用于所有角色。

* [roleParams](https://www.yiichina.com/doc/api/2.0/yii-filters-accessrule" \l "$roleParams-detail)：指定将传递给 [yii\web\User::can()](https://www.yiichina.com/doc/api/2.0/yii-web-user" \l "can()-detail) 的参数。 请参阅下面描述RBAC规则的部分，了解如何使用它。 如果此选项为空或未设置，则不传递任何参数。——不用YII自带的RBAC则此参数不传。
* [ips](https://www.yiichina.com/doc/api/2.0/yii-filters-accessrule" \l "$ips-detail)：指定该规则用于匹配哪些 [客户端IP地址](https://www.yiichina.com/doc/api/2.0/yii-web-request" \l "$userIP-detail) 。 IP 地址可在其末尾包含通配符 \* 以匹配一批前缀相同的IP地址。 例如，192.168.\* 匹配所有 192.168. 段的IP地址。 如果该选项为空或者不使用该选项，意味着该规则适用于所有地址。—— 一般不传。
* [verbs](https://www.yiichina.com/doc/api/2.0/yii-filters-accessrule" \l "$verbs-detail)：指定该规则用于匹配哪种请求方法（例如GET，POST）。 这里的匹配大小写不敏感。
* [matchCallback](https://www.yiichina.com/doc/api/2.0/yii-filters-accessrule" \l "$matchCallback-detail)：指定一个PHP回调函数用于 判定该规则是否满足条件。（译者注：此处的回调函数是匿名函数）——匿名函数用来写自定义规则的代码。

## matchCallback 选项

以下例子展示了如何使用 matchCallback 选项， 可使你设计任意的访问权限检查逻辑：



# 基于角色的存取控制 （RBAC）

这是YII自带的RBAC，表设计跟常规的不一样。

主要是权限表和角色表是共用一个表，然后多了一个****auth\_rule（****权限规则表****）。****

****具体用法见文章 Yii2的RBAC实现原理详解：****

<https://blog.csdn.net/phpduang/article/details/81587496>

另外，yii的rbac没有支持对应的后台前端代码，需要自己找对应插件或者自己实现。

如果要添加权限、角色等表的数据，就要根据yii的规则写代码再去命令行执行命令插入对应数据，或者直接写SQL插入。

本质上只是在组件上进行了一定的支持，并不能直接拿来就用。