

搜索`symfony voter`，查看Voter文档。在文档中有一段是关于访问策略的描述，我们查看一下。

这里说我们可以修改访问策略，通常情况下只有一个Voter会执行`vote()`方法，剩下的Voter会弃权，`supports()`方法会返回`false`，停止投票。但是在理论上我们可以使用多个Voter对一个操作或者对象进行投票。这里有一段描述，假设我们有一个Voter用来检查当前的用户是不是我们网站的会员，然后有另一个Voter来检查当前的用户年龄是否大于18岁。这样就有了两个Voter来对同一个用户进行权限的检查，对于多个Voter处理同一个操作或者对象时，有几种可以使用的策略。

首先是第一种，这也是默认的策略，只要有一个Voter通过了授权，那么就可以进行访问。我们查看一下第二个策略，查看一下翻译，**共识协议**，这个意思就是说少数服从多数，如果有大多数Voter通过授权的话，那么就可以进行访问。当然如果通过授权的Voter数量和反对的Voter数量相等的话，我们就需要添加一个额外的配置。然后是第三种策略，全场一致，只有所有的Voter都投的同意票，那么才有访问权限。最后一个是`priority`，我们的Voter类它是一个服务类，我们可以为服务类设置优先级，最高优先级的Voter执行的结果就是最终的授权结果。当然还有另外一种情况，就是所有的Voter都弃权。在这种情况下，我们需要添加一个额外的配置。

我们再往下看，在`security.yaml`配置文件中。我们可以通过这段代码来进行配置的修改，回到项目双击`shift`键，我们搜索`AccessDecisionManager`类。前面课程我们讲到通过`decide()`方法来验证当前的授权。根据不同的策略名称会执行对应的`decide`方法，我们查看一下类中的方法，这里有个`decideAffirmative()`。

我们查看一下代码，首先它会对所有的Voter进行遍历，只要有一个Voter通过了授权，那么当前的方法就直接返回，否则的话就返回`false`。

我们再往下看，我们来看第二种情况，就是大多数同意，首先它定义了两个变量。一个是通过授权的变量，一个是反对授权的变量。它们都是一个整数类型，用来统计数量。然后遍历所有的Voter，将Voter的结果进行统计。如果同意大于反对，那么就返回真，否则的话就返回`false`。

我们再往下看这种情况就是全场一致，所有的Voter都同意授权的话，才会返回真，这里对所有的Voter进行遍历，只要有一个Voter反对授权，那么就直接返回`false`。

最后一个是根据优先级进行判断，`votes`集合，会根据Voter的优先级进行排序。根据最高优先级的Voter的授权结果进行返回，如果同意授权就返回`true`，否则的话就返回`false`。

这就是Voter的使用策略。回到项目，我们打开管理端登录一下超级管理员。点击新增文章，我们已经为文章添加了作者属性，表单中也显示了作者行。这里我们需要修改一下，我们在添加文章时自动的将当前用户设置为文章的作者。

在下节课，我们使用EasyAdminBundle提供的事件来添加文章的作者。