

打开PostCrudController类, 我们来调整一下字段, 我们不需要在表单中显示作者。这里修改一下, 我们使用onlyOnIndex()方法, 只让作者在首页显示。

我们往上看, 我们查看PostCrudController类的父类, 查看类中的方法。当我们在创建一篇新的文章时, 会调用Controller类中的new()方法。查看new()方法, 在new()方法中首先发送BeforeCrudActionEvent事件, 然后继续往下, 这里有处理表单提交的流程。我们继续往下看, 在表单处理流程中, 有个BeforeEntityPersistedEvent, 这个事件发生在对象持久化之前, 我们可以创建监听器或者订阅器处理这个事件。

在事件处理中, 我们为文章对象设置作者属性。打开控制台, 输入symfony console make:subscriber, 类名我们叫做PostSetAuthorSubscriber。我们复制事件的类名, 粘贴。

打开对应的订阅器类, 我们查看一下事件类的源码。在事件类中有个属性, 这个属性就是我们要持久化的Entity对象实例。回到订阅器类, 定义一个变量等于\$event->getInstance()。

我们进行一下条件判断, 如果\$instance变量是Post类的一个实例。我们为\$instance变量设置作者属性, 可以通过依赖注入的方法注入Security对象, 通过Security对象来获取当前的用户。这里输入\$this->security->getUser(), 这样我们的订阅器就已经写好了。

```
#src/EventSubscriber/PostSetAuthorSubscriber.php

class PostSetAuthorSubscriber implements EventSubscriberInterface
{
    /**
     * @var Security
     */
    private Security $security;

    public function __construct(Security $security)
    {
        $this->security = $security;
    }

    public function
onBeforeEntityPersistedEvent(BeforeEntityPersistedEvent $event)
    {
        $instance = $event->getEntityInstance();
        if ($instance instanceof Post){
            $instance->setAuthor($this->security->getUser());
        }
    }

    public static function getSubscribedEvents()
    {
        return [
            BeforeEntityPersistedEvent::class =>
'onBeforeEntityPersistedEvent',
        ];
    }
}
```

回到浏览器刷新页面，我们当前的用户是admin用户。我们新增一篇文章Hello world。点击添加。我们看到手动添加的文章，作者就是我们当前的用户，我们看到当前的管理端有很多英文。比如说菜单项和列表的表头。

在底部的工具栏，我们看翻译这里，一共缺失了14条翻译项。在下节课，我们来对管理端进行一下国际化翻译。