

输入`symfony console list make`，来查看所有可用的make命令。这里有个`make:subscriber`命令行，这个命令行可以帮助我们快速的创建订阅器。拷贝命令行，粘贴命令行，订阅器的名称我们叫作`AfterCommentSubmitSubscriber`。这里让我们指定一个需要订阅的事件，我们订阅之前创建的事件类。复制类名，粘贴回车。我们查看一下命令行创建的订阅器类。

订阅器类实现了订阅器接口，然后在`getSubscribedEvents()`方法中，它返回了一个数组，数组中我们创建的事件类，它指向了一个方法，我们将在这个方法中来接收事件，并且处理数据。

我们想使用订阅器过滤掉评论中的所有敏感字，在config目录下，我们打开services.yaml文件。我们定义一些敏感字，在parameters键下，我们定义一些敏感字`strip_words`。我们需要定义一个数组，数组每行之前添加一个横线，代表一个数组项。我们过滤hello，过滤world，过滤abc。

```
#config/services.yaml

parameters:
    # ...
    strip_words:
        - 'hello'
        - 'world'
        - 'abc'
```

回到订阅器，我们先来获取评论对象。`$comment`等于`$event->getComment()`；然后获取评论对象的消息，`$message`等于`$comment->getMessage()`，我们想用星号(*)来替换消息中的敏感词，首先我们要获取刚刚设置的所有敏感词。

在构造方法中，我们通过依赖注入的方式来获取ParameterBag对象。参数类型，我们选择`ParameterBagInterface`。鼠标点击参数，按着alt键点击回车，我们来初始化属性。这样我们可以直接在类中使用`$parameterBag`对象了。首先我们来获取所有的敏感字`get('strip_words')`。

我们可以添加一个方法来处理消息中的所有敏感字，方法名称叫`strip_words()`，需要传入两个参数。第一个参数是消息的正文，第二个参数是所有的敏感字。我们循环遍历所有的敏感字。如果消息中有敏感字，我们进行替换。`str_replace()`方法来进行替换，第一个参数我们传入`$word`，第二个参数传入替换的字符我们传入三个星号，第三个参数我们传入`$message`。我们需要使用到返回值。返回值我们定义为`$message`，这样循环就会替换所有的敏感词。最后我们返回替换过的消息。

回到事件处理方法，我们先过滤所有的敏感词，第一个参数传入`$message`，第二个参数传入`$words`。我们将它的返回值重新设置到`$comment`对象中，现在我们的订阅器类就已经编写好了。

```
#src/EventSubscriber/AfterCommentSubmitSubscriber.php

class AfterCommentSubmitSubscriber implements EventSubscriberInterface
{
    /**
     * @var ParameterBagInterface
     */
    private ParameterBagInterface $parameterBag;

    public function __construct(ParameterBagInterface $parameterBag)
```

```
{
    $this->parameterBag = $parameterBag;
}

public function onAfterCommentSubmitEvent(AfterCommentSubmitEvent
$event)
{
    $words = $this->parameterBag->get('strip_words');

    $comment = $event->getComment();
    $message = $comment->getMessage();

    $comment->setMessage($this->stripWords($message, $words));
}

public static function getSubscribedEvents()
{
    return [
        AfterCommentSubmitEvent::class => 'onAfterCommentSubmitEvent',
    ];
}

private function stripWords(string $message, array $words)
{
    foreach ($words as $word){
        $message = str_replace($word, ' *** ', $message);
    }

    return $message;
}
}
```

回到浏览器我们来提交一个评论，我们在评论中间添加一些数据，点击提交。查看评论列表，我们提交的评论已经成功的过滤了敏感字。回到代码，我们添加的订阅器类并没有像监听器一样添加一些配置，这是因为订阅器实现了订阅器接口。

我们打开services.yaml文件，这里`autoconfigure`设置为`true`，Symfony会自动的为我们的订阅器进行配置，所以就不用再编写配置了。

下节课，我们再更多的了解一下事件系统。