Doctrine的事件订阅器需要实现Doctrine提供的订阅器接口

```
class FileUploadSubscriber implements EventSubscriberInterface
{

    public function getSubscribedEvents(): array
    {
        return [
            Events::prePersist,
            Events::preUpdate,
            Events::postPersist,
            Events::postUpdate,
        ];
    }

    public function prePersist(LifecycleEventArgs $args): void
    {
        dd($args);
    }

    public function preUpdate(LifecycleEventArgs $args): void
    {
        dd($args);
    }

    public function postPersist(LifecycleEventArgs $args): void
    {
        dd($args);
    }

    public function postUpdate(LifecycleEventArgs $args): void
    {
        dd($args);
    }
}
```

把订阅器添加到容器，别忘了添加tag，然后修改`Extension`类，加载`doctrine.xml`文件

```
<?xml version="1.0" encoding="utf-8" ?>
<container xmlns="http://symfony.com/schema/dic/services"
           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
           xsi:schemaLocation="http://symfony.com/schema/dic/services
http://symfony.com/schema/dic/services/services-1.0.xsd">
    <services>
        <service
id="Teebb\UploadBundle\EventSubscriber\FileUploadSubscriber"
public="true">
            <tag name="doctrine.event_subscriber"/>
        </service>
        <service id="teebb.upload.event.file_upload_subscriber"

alias="Teebb\UploadBundle\EventSubscriber\FileUploadSubscriber"
                 public="true"/>
    </services>
</container>
```

试着上传一个文件，停在了第25行，`LifecycleEventArgs`可以添加判断，并使用Handler生成新的文件名：

```
class FileUploadSubscriber implements EventSubscriberInterface
{
    public function __construct(UploadHandler $uploadHandler)
    {
        $this->uploadHandler = $uploadHandler;
    }

    public function prePersist(LifecycleEventArgs $args): void
    {
        $object = $args->getEntity();
        if ($object instanceof SimpleFile)
        {
            $fileName = $this->uploadHandler->getFileName($object-
>getUploadedFile());
            $object->setName($fileName);
        }
    }
}
```

这里要修改xml文件添加构造参数

```
<?xml version="1.0" encoding="utf-8" ?>
<container xmlns="http://symfony.com/schema/dic/services"
           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
           xsi:schemaLocation="http://symfony.com/schema/dic/services
http://symfony.com/schema/dic/services/services-1.0.xsd">
    <services>
        <service
id="Teebb\UploadBundle\EventSubscriber\FileUploadSubscriber"
public="true">
            <argument type="service"
id="Teebb\UploadBundle\Handler\UploadHandler"/>
            <tag name="doctrine.event_subscriber"/>
        </service>
        <service id="teebb.upload.event.file_upload_subscriber"

alias="Teebb\UploadBundle\EventSubscriber\FileUploadSubscriber"
                 public="true"/>
    </services>
</container>
```

刷新重新提交表单，看到现在断在了第52行，在`postPersist`事件的方法里，这时可以使用Handler移动文件了。

```
class FileUploadSubscriber implements EventSubscriberInterface
{
    public function __construct(UploadHandler $uploadHandler)
    {
        $this->uploadHandler = $uploadHandler;
    }

    public function prePersist(LifecycleEventArgs $args): void
    {
        $object = $args->getEntity();
        if ($object instanceof SimpleFile)
        {
            $fileName = $this->uploadHandler->getFileName($object-
>getUploadedFile());
            $object->setName($fileName);
        }
    }

    public function postPersist(LifecycleEventArgs $args): void
    {
        $object = $args->getEntity();
        if ($object instanceof SimpleFile)
        {
            $this->uploadHandler->upload($object->getUploadedFile(),
$object->getName());
        }
    }
}
```

提取共用的代码，最终效果：

```
class FileUploadSubscriber implements EventSubscriberInterface
{
    /**
     * @var UploadHandler
     */
    private $uploadHandler;

    public function __construct(UploadHandler $uploadHandler)
    {
        $this->uploadHandler = $uploadHandler;
    }

    public function getSubscribedEvents(): array
    {
        return [
            Events::prePersist,
            Events::preUpdate,
            Events::postPersist,
            Events::postUpdate,
        ];
    }

    public function prePersist(LifecycleEventArgs $args): void
    {
        $object = $args->getEntity();
        $this->setFileObjectName($object);
    }

    public function preUpdate(PreUpdateEventArgs $args): void
    {
        $object = $args->getEntity();
        $this->setFileObjectName($object);
    }

    public function postPersist(LifecycleEventArgs $args): void
    {
        $object = $args->getEntity();
        $this->uploadFile($object);
    }

    public function postUpdate(LifecycleEventArgs $args): void
    {
        $object = $args->getEntity();
        $this->uploadFile($object);
    }

    private function setFileObjectName($object)
    {
        if ($object instanceof SimpleFile) {
            //这里要添加判断
            if ($uploadedFile = $object->getUploadedFile()) {
                $fileName = $this->uploadHandler-
>getFileName($uploadedFile);
                $object->setName($fileName);
            }
        }
    }

    private function uploadFile($object)
    {
        if ($object instanceof SimpleFile) {
            //这里要添加判断
            if ($uploadedFile = $object->getUploadedFile()) {
                $this->uploadHandler->upload($uploadedFile, $object-
>getName());
            }
        }
    }
}
```

现在如果数据库插入失败 文件就不会上传了。现在发现了另一个问题：代码里把`SimpleFile`类型写死了，我是我在课程里随意定义一个类，你的项目可能不叫这个名，还有一种情况，你的项目里，可能有多个文件类型，不同的文件类型存储在不同的位置甚至会存储到其他的云平台。 下节课：我们先来解决文件类型类名的问题。