

打开底部控制台，输入`symfony console make:form`，我们来创建一个自定义的表单类型。类型的名称，我们叫做`FileManagedType`。这里提示让我们指定一个类名，我们查看一下`FileManaged`类。`FileManaged`类的各个属性都是通过上传的文件来获取的。

我们需要一个文件上传的表单行，这里我们不能使用`FileManaged`类。打开控制台，直接回车就可以了。打开`FileManagedType`类，在`buildForm()`方法中，`$builder`的`add()`方法添加了一个表单行，我们自定义的`FileManagedType`需要在表单中生成一个表单上传框，我们可以覆盖一下父方法`getParent()`。

在`getParent()`方法中，我们让`FileManagedType`返回的类型是`FileType`，这样`FileManagedType`就是`FileType`的一个子类型。我们删除`buildForm()`方法中的代码。修改`CommentType`，将`entry_type`的类型设置为`FileManagedType`。

```
#src/Form/FileManagedType.php

class FileManagedType extends AbstractType
{
    public function buildForm(FormBuilderInterface $builder, array $options)
    {
    }

    public function configureOptions(OptionsResolver $resolver)
    {
        $resolver->setDefaults([
            // Configure your form options here
        ]);
    }

    public function getParent()
    {
        return FileType::class;
    }
}
```

回到浏览器再次访问详情页，点击添加文件，现在添加文件按钮，添加的表单行，和之前是一样的。

Symfony的表单系统提供了一些表单的事件，我们搜索`symfony form events`。打开`symfony.com`文档。根据表单的工作流程，Symfony的表单事件分为两类，第一类是用于表单数据的预填充，第二类是用于表单数据的提交。

我们先来看表单事件的预填充，当我们在Controller中创建一个Form对象后，`$this->createForm()`方法中，可以设置第二个参数`$data`，设置Data后，可以在表单中通过事件来对表单中的数据进行修改。

我们查看一下`PRE_SET_DATA`和`POST_SET_DATA`的说明，`PRE_SET_DATA`通常用于在预填充时修改数据，或者根据表单预填充的数据来动态的添加或者删除表单行。我们再来看`POST_SET_DATA`，`POST_SET_DATA`通常在表单的数据进行预填充后进行数据读取。

我们继续往下看, 当表单中的数据进行提交时, 有三个事件可以处理数据, 分别是`PRE_SUBMIT`, `SUBMIT`和`POST_SUBMIT`。根据字面意思, `PRE_SUBMIT`是在表单提交之前进行数据处理, `SUBMIT`是在表单提交时进行数据处理, `POST_SUBMIT`是在表单提交之后进行数据的处理。

我们查看一下文档说明, `PRE_SUBMIT`通常在数据提交到表单之前, 修改请求的数据, 或者在表单提交数据之前添加或者删除表单字段。`SUBMIT`事件可以从数据的规范化表示中更改数据, 我往上看流程图。当我们提交完数据之后, 数据会首先进行规范化。规范化, 就是将表单中的数据规范为一个数组, 然后再将数组的数据转换为对象, 转换为对象过程叫做反规范化。然后是`POST_SUBMIT`事件, `POST_SUBMIT`事件通常用于在表单提交后获取数据。

对于文件的上传, 我们可以在`SUBMIT`事件中进行处理, 获取表单提交的文件对象, 然后转换为`FileManaged`对象。

在下节课我们将来完成这段代码。