

打开CommentType类, 在`buildForm()`方法中, 我们使用`add()`方法添加一个表单行, 添加到message行后面, 表单行的名称就是属性名。我们添加`files`属性, 我们需要指定表单行的类型, `files`属性它是个集合属性。

回到浏览器, 我们搜索`symfony collectiontype`。打开`symfony.com`这个页面的文档, `CollectionType`通常用于在表单中渲染一个集合属性, 我们向下看, 这里有最基本的使用方法。

表单行的类型, 使用`CollectionType`, 可以通过`entry_type`来设置集合中单个对象它所使用的表单行类型, 然后可以使用`entry_options`来对单个表单行进行一些设置。

回到项目, 表单行类型我们输入`CollectionType`, 我们选择Form组件下的`CollectionType`。设置`entry_type`, 我们需要文件上传这里使用`FileType`, `FileType`也是表单组件提供一个类型。我们想在评论表单中动态的添加文件上传表单行, 可以设置`allow_add`为`true`。

```
#src/Form/CommentType.php
class CommentType extends AbstractType
{
    public function buildForm(FormBuilderInterface $builder, array $options)
    {
        $builder
            // ...
            ->add('files', CollectionType::class, [
                'entry_type' => FileType::class,
                'allow_add' => true,
            ])
            // ...
        ;
    }
}
```

回到浏览器, 打开博客首页, 打开第一篇文章。现在在表单中就添加了一个Files表单行, 但是Files并没有显示文件上传组件, 我们点击右键检查Files。在Files表单行中, 我们查看html代码, 我们发现有个`data-prototype`。`data-prototype`属性它的值就是一个文件上传表单的代码, 我们需要通过JS的方式将文件上传组件的代码动态的添加到表单行中。

首先我们需要在Files表单行中添加一个按钮, 通过按钮的点击事件, 我们动态的将代码追加到表单行中。新建一个标签页, 我们搜索`symfony form theme`。打开`symfony.com`文档, 我们往下看, 我们可以定一个模板文件。在模板文件中定义一个区块, 通过区块名覆盖的方式来自定义表单行模板。

回到项目, 在`templates`目录中我们新建一个目录叫`form`, 在`form`目录中, 我们新建一个模板文件, 模板文件的名称我们叫做`_comment_files_widget.html.twig`。在模板文件中, 我们定义一个区块。区块的名称, 我们参考文档首先是下划线, 然后是类名小写, 下划线, 然后是属性名称小写, 然后是下划线`widget`。

我们复制一下, 粘贴, 进行下修改, `_comment_files_widget`, 结束标签。我们要覆盖的是`CollectionType`类型的表单行模板。回到项目我们查看一下form组件源码, 打开`vendor`目录, 在`symfony`目录下, 我们打开`twig-bridge`, 打开`Resources`目录, 这里有个`Form`文件夹, `Form`文件夹下就是我们表单的模板文件。

我们使用了bootstrap 4的样式, 我们打开这个文件。在模板文件中, 表单行的模板都是使用表单行类型的名称, 加上下划线widget作为区块名。我们搜索一下collection这里没有找到。

我们查看引用(口误)模板, 在引用(口误)模板中我们搜索collection, 也没有找到。我们在form_div_layout文件中继续查找, 现在找到了collection_widget。我们复制区块中的代码, 回到我们自定义的模板粘贴。

block()方法会渲染我们的表单行, 我们在block()方法下面添加一个按钮button。button类型我们设置为按钮类型, 添加样式类, 我们需要处理按钮的点击事件, 为按钮添加一个类, 按钮名称我们叫做添加文件。

```
#templates/form/_comment_files_widget.html.twig

{% block _comment_files_widget %}
    {% if prototype is defined and not prototype.rendered %}
        {%- set attr = attr|merge({'data-prototype': form_row(prototype)
    }) -%}
    {% endif %}
    {{- block('form_widget') -}}

    <button type="button" class="btn btn-sm btn-outline-secondary js-add-
file-row-btn">添加文件</button>
{% endblock %}
```

回到浏览器, 我们刷新一下文章详情页, 我们自定义的模板并没有生效。回到项目, 我们需要修改一下twig组件的配置, 打开config目录, 找twig.yaml文件, 我们之前设置了form_themes, 我们还需要指定一下使用我们刚刚编写模板文件。在form目录下comment_files_widget.html.twig, 回到浏览器刷新。

```
#config/packages/twig.yaml
twig:
    default_path: '%kernel.project_dir%/templates'
    form_themes:
        - 'bootstrap_4_layout.html.twig'
        - 'form/_comment_files_widget.html.twig'
```

现在在Files表单行中就有一个添加文件按钮, 在下节课我们来为添加文件按钮添加事件处理。