

我们点击右键检查一下按钮，当我们点击按钮时，首先会获取表单行的DOM元素，然后再获取DOM元素的 `data-prototype` 的值。就是文件上传的原型代码，再将原型代码中的 `__name__` 进行一下替换，替换为文件上传行的序号，再追加到表单行中。为了更好的获取到表单行，我们可以在表单行中添加一个样式类。

回到项目，打开 `CommentType` 类，在 `files` 表单行中我们添加一个 `attr` 设置，我们添加一个样式类，样式类的名称我们叫做 `input-row-wrapper`。

回到浏览器刷新，现在表单行中添加了一个样式类，我们可以先获取 `button` 按钮的父级元素，再通过父级元素来查找 `div` 元素，再通过 `div` 元素获取原型代码，对代码进行处理。

回到项目我们打开 `app.js` 文件，在 `ready()` 方法中，我们输入 `$('#button.js-add-file-row-btn')`，然后监听 `click` 事件，添加方法，我们为方法添加一个参数 `element`。为了节省时间，我这里就直接粘贴写好的代码了。

```
#assets/app.js
// ...

$('#button.js-add-file-row-btn').on('click', function (element) {
    //查询input-wrapper DOM
    var fileInputWrapper = $(element.target).closest('fieldset.form-group').find('div.input-row-wrapper');
    //获取当前input行计数
    var inputCount = fileInputWrapper.children().length;
    //获取input prototype并进行序号修改
    var inputCode = fileInputWrapper.data('prototype');
    inputCode = inputCode.replace(/__name__/g, inputCount);

    fileInputWrapper.append(inputCode);
})

// ...
```

首先我们获取按钮的父级元素，再查找 `.input-row-wrapper div` 元素，然后计算当前 `div` 元素它内部的文件上传表单行的数量，然后获取 `prototype` 原型代码。再对原型代码中的 `__name__` 字符串进行一下序号的替换，最后将替换过的代码追加到 `div` 元素中。

我们打开控制台，确保资源开发服务器已经启用，并且刚刚编写的代码已经构建成功了，回到浏览器我们刷新。当我们点击添加文件时，会自动的在 `Files` 表单行中添加一个文件上传表单行。但是表单行的标签显示的是乱码，我们可以让表单行不显示标签，回到 `CommentType`，我们设置 `entry_options`，`label` 设置我们设置为 `false`，回到浏览器刷新。

添加文件，现在就可以了，我们选择一个文件，`bootstrap 4` 版本，选择文件后有个显示 `bug`，回到 `app.js` 文件，我们直接粘贴修复 `bug` 的代码。

```
#assets/app.js
// ...

window.fixFileInputName = function (element) {
```

```
let fileName = $(element).val().split('\\').pop();
$(element).next('.custom-file-label').html(fileName);
}

//...
```

回到浏览器，点击添加文件，当我们选择要上传的文件后，我们希望文件上传框的`onchange`事件能够调用这个方法，回到`CommentType`类，我们在`entry_options`设置中，为文件上传框添加一个属性，属性名称叫做`onchange`，直接调用我们刚刚编写的方法，参数就传入`this`。

```
#src/Form/CommentType.php
class CommentType extends AbstractType
{
    public function buildForm(FormBuilderInterface $builder, array
$options)
    {
        $builder
            // ...
            ->add('files', CollectionType::class, [
                'entry_type' => FileType::class,
                'entry_options' => [
                    'label' => false,
                    'attr' => [
                        'onchange' => 'fixFileInputName(this)'
                    ]
                ],
                'allow_add' => true,
                'attr' => [
                    'class' => 'input-row-wrapper'
                ]
            ])
            // ...
        ;
    }
}
```

回到浏览器刷新，我们点击添加文件按钮，上传一个文件，现在这个bug就解除了。我们补充一下表单，点击提交，现在出错了。`files`属性它是`FileManaged`对象的集合，但是我们上传后的文件Symfony帮我们封装成`UploadedFile`对象，两个对象不一致，所以表单提交出错了。

在下节课我们来解决这个问题。