

继续查看`{{ dump() }}`中变量的数据, 如果`data`变量不为null, 那么在模板中显示文件链接或者图片

```
{% block teebb_file_upload_widget -%}

    <{{ element|default('div') }} class="custom-file">
    {%- set type = type|default('file') -%}
    {%- set input_lang = 'en' -%}
    {% if app is defined and app.request is defined %}{%- set input_lang =
app.request.locale -%}{%- endif -%}
    {%- set attr = {lang: input_lang} | merge(attr) -%}
    {{- block('form_file_widget_simple') -}}
    {%- set label_attr = label_attr|merge({ class:
(label_attr.class|default('') ~ ' custom-file-label')|trim
})|filter((value, key) => key != 'id') -%}
    <label for="{{ form.vars.id }}" {% with { attr: label_attr } %}{{
block('attributes') }}{% endwhile %}>
        {%- if attr.placeholder is defined and attr.placeholder is not
none -%}
            {{- translation_domain is same as(false) ? attr.placeholder :
attr.placeholder|trans({}, translation_domain) -}}
            {%- endif -%}
        </label>
    </{{ element|default('div') }}>

    {# 这里只显示了文件名, 要显示可访问路径需要设置路径前缀, 可以为表单类设置一个
option, options['base_path'], 然后进行拼接。 #}
    {% if data %}
        {{ data.name }}
    {% endif %}
{% endblock %}
```

这里只显示了文件名, 要显示可访问路径需要设置路径前缀, 可以为表单类设置一个option, options['base\_path'], 然后进行拼接。

再修改CommentType类, 设置对应的entry\_options:

```
// CommentType
->add('files', CollectionType::class, [
//
    'entry_type' => FileType::class,
    'entry_type' => FileManagedType::class,
    'entry_options' => [
        'label' => false,
        'attr' => [
            'onchange' => 'fixFileInputName(this)'
        ],
        'constraints' => [
//
            new FileManaged([
//
                'image/*'
//
            ])
        ]
    ]
]);
```

```
        new FileConstraint([
            'image/*'
        ])
    ],
    'file_class' => SimpleFile::class,
    'base_path' => 'teebb_upload' . DIRECTORY_SEPARATOR,
//这里添加路径
],
```

然后覆盖父类 `buildView()` 方法, 把 `base_path` 设置到 `form.vars` 数组里:

```
public function buildView(FormView $view, FormInterface $form, array
$options)
{
    $view->vars['base_path'] = $options['base_path'];
}
```

如果想用 `img` 标签显示图片可以额外定义一个 option, `options['show_image'] = false`, boolean 类型的。为真时显示图片, 否则显示超链接。

额外定义一个 option, `image_attr` 用于修改图片显示的样式。

```
public function buildView(FormView $view, FormInterface $form, array
$options)
{
    $view->vars['base_path'] = $options['base_path'];
    $view->vars['show_image'] = $options['show_image'];
    $view->vars['image_attr'] = $options['image_attr'];
}
```

```
{% block teebb_file_upload_widget -%}
...
{% if data %}
    {% if show_image %}
        
    {% else %}
        <a href="{{ asset(form.vars.base_path ~ data.name) }}"
target="_blank">{{ data.originName }}</a>
    {% endif %}
{% endif %}
{% endblock %}
```

```
{% block image_attr %}
    {%- for attrname, attrvalue in image_attr -%}
        {{- " " -}}
        {%- if attrname in ['placeholder', 'title'] -%}
            {{- attrname }}="{{ translation_domain is same as(false) or
attrvalue is null ? attrvalue :
attrvalue|trans(attr_translation_parameters, translation_domain) }}"
        {%- elseif attrvalue is same as(true) -%}
            {{- attrname }}="{{ attrname }}"
        {%- elseif attrvalue is not same as(false) -%}
            {{- attrname }}="{{ attrvalue }}"
        {%- endif -%}
    {%- endfor -%}
{% endblock %}
```

我们自定义了表单的模板，并且在Teebblog项目中手动设置了表单的模板。如果有其他程序员使用了我们的Bundle还需要手动配置表单的模板，那就太不自动化了。我们可以添加代码让twig-bundle自动读取我们自己的表单模板。