


ez_render_field

- [Description](#)
- [Prototype and Arguments](#)
- [Override a field template block](#)
 - [Inline override](#)
 - [Inline override using current template](#)
 - [Global override](#)


Description

`ez_render_field()` is a Twig helper allowing to display a Content's Field value, taking advantage of the template block exposed by the FieldType used.

 Template blocks for built-in FieldTypes [reside in EzPublishCoreBundle](#).


Prototype and Arguments

```
ez_render_field( eZ\Publish\Core\Repository\Values\Content\Content content, string fieldDefinitionIdentifier [, hash params] )
```

Argument name	Type	Description
<code>content</code>	<code>eZ\Publish\Core\Repository\Values\Content\Content</code>	Content object the displayable field belongs to.
<code>fieldDefinitionIdentifier</code>	<code>string</code>	The identifier the Field is referenced by.
<code>params</code>	<code>hash</code>	<p>Hash of parameters that will be passed to the template block.</p> <p>By default you can pass 2 entries:</p> <ul style="list-style-type: none">• lang (to override the current language, must be a valid locale with xxx-YY format)• template (to override the template to use, see below)• attr (hash of HTML attributes you want to add to the inner markup)• parameters (arbitrary parameters to pass to the template block) <div> Some FieldTypes might expect specific entries under the parameters key, like the Map Location field type</div>

Override a field template block

In some cases, you may not want to use the built-in field template block as it might not fit your markup needs. In this case, you can choose to override the template block to use by specifying your own template. You can do this inline when calling `ez_render_field()`, or globally by prepending a field template to use by the helper.

 Your template block must comply to a regular FieldType template block, [as explained in the FieldType documentation](#).

Inline override

You can easily use the template you need by filling the **template** entry in the `params` argument.

```

{{ ez_render_field(
    content,
    'my_field_identifier',
    { 'template': 'AcmeTestBundle:fields:my_field_template.html.twig' }
) }}

```

The code above will load `my_field_template.html.twig` located in `AcmeTestBundle/Resources/views/fields/`.

Inline override using current template



Version compatibility

Inline override using current template is possible as of **5.2 / 2013.11**

If you want to override a specific field template only once (i.e. because your override would be only valid in your current template), you can specify the current template to be the source of the field block.

Inline override using current template

```

{% extends "MyBundle::pagelayout.html.twig" %}

{% block content %}
    {# Note that "tags" is a field using ezkeyword fieldType #}
    <div class="tags">{{ ez_render_field( content, "tags" , { "template": _self } ) }}</div>
{% endblock %}

{# Here begins the inline block for my ezkeyword field #}
{% block ezkeyword_field %}
    {% spaceless %}
        {% if field.value.values|length() > 0 %}
            <ul>
                {% for keyword in field.value.values %}
                    <li>{{ keyword }}</li>
                {% endfor %}
            </ul>
        {% endif %}
    {% endspaceless %}
{% endblock %}

```



Limitation

Using `_self` will only work if your current template is extending another one.

This is basically the same limitation than for [Symfony form themes](#).

Global override

In the case where you want to systematically reuse a field template instead of the default one, you can append it to the field templates list to use by `ez_render_field()`.

To make your template available, you must register it to the system.

ezpublish/config/ezpublish.yml

```
ezpublish:
  system:
    my_siteaccess:
      field_templates:
        -
          template: "AcmeTestBundle:fields:my_field_template.html.twig"
          # Priority is optional (default is 0). The higher it is, the
higher your template gets in the list.
          priority: 10
```



You can define these rules in a dedicated file instead of `ezpublish/config/ezpublish.yml`. [Read the cookbook recipe to learn more about it.](#)