

# Templating with Twig

# Overview

- easy syntax
- inheritance
- blocks
- autoescaping
- sandboxing
- performance
- different sources support
- etc.

```
{#  comments are
    not rendered
#}

{{ variable }}

{{ 'simple text' }}
simple text as well

{% set foo = 'bar' %}

{% if foo is 'bar' %}
true
{% endif %}

{% for user in users %}
    * {{ user.name }}
{% else %}
    No users have been found.
{% endfor %}
```

# Templates

- Advantages
  - More readable than php
  - Compiled to php to increase performance
- Disadvantages
  - Allowed to do almost everything you can do in PHP

# Extending

- Doesn't support multiple inheritance
- It's real PHP extends

# Blocks

- A way to allow override part of the template
- Placeholders and replacements
- Can be nested

# Compilation

- Templates compiled to PHP Classes
- Blocks compiled to Templates class methods
- Easy to search through compiled templates

# Global Variables

```
# app/config/config.yml
twig:
  # ...
  globals:
    website_name: My Fancy website
    website_slogan: '%app.website_slogan%'
    user_provider: '@AppBundle\Service\UserProvider'
```

# Functions and Filters

- Filters - modify or format one of the arguments
  - FilterExtension age
- Function - returns some result
  - FilterExtension oro\_format\_filename
- Tests
  - definition, function, old implementation



# Twig extensions

- Twig\_ExtensionInterface
- Official Twig Extensions
- Performance issue

# Macros

Twig extensions written in Twig

- to reuse HTML blocks

Difference from PHP functions:

- "|default" filter used for default variables
- arguments are always optional.
- extra arguments goes to "varargs" variable as an array

Example:

[definition](#), [import](#), [usage](#)

# Include

Render template inside another one

- [example](#)

# Embed

- Include & extend at the same time
- In cache new PHP class will be created for every embed with overridden blocks
- [example](#)

# Sandbox

When you don't trust content editors.

- Regulated with SecurityPolicy
- We have unrelated 2 sandboxes in ORO:
  - Email templates
  - Content blocks
- Not easy to extend

# Assets management

- Asset
- Assetic
- *Webpack*

# Overriding Templates at ORO

- `/templates` will work only when the template was referenced right
- `src/Resources/{BUNDLE_NAME}/views/{PATH}/TO/TEMPLATE.html.twig` will always work
- Bundle inheritance is deprecated
- Layout always requires layout update YAML file to override the template

# Debugging

- `lint:twig`
- `debug:twig`
- Find in compiled code by template names
- Dump

```
{{ dump(variable) }}  
{% dump(variable) %}  
  
# Silver bullet !!!  
{% dump(_context) %}  
# or  
{% dump(_context|keys) %}
```