

**Extend functionality  
with bundles**

# Service/parameter overriding

- Define service/parameter with the same name
- Only the latest definition is used

<https://github.com/oro-inc/platform/blob/3.1/src/Oro/Bundle/EntityBundle/Resources/config/services.yml#L2-L7>

*Demonstration code, e.g. in customization >>>*

# Service decoration

- Decorates original service
- Can inject original service into decorator
- Most of the time, the decorator should be declared private

<https://github.com/oro-inc/orocommerce/blob/3.1/src/Oro/Bundle/WebsiteSearchBundle/Resources/config/services.yml#L108-L114>

# Compiler passes

- Dynamic manipulations with services and parameters
- Allows to add custom logic during the compilation of DIC
- Examples
  - Add method call
  - Change class

# ~~Bundle inheritance~~

- Allows to override parts where bundle based paths are used (e.g. @OroUserBundle)
  - Routing
  - Controllers
  - Templates
- Each bundle may have only one parent bundle

[\[Documentation\]](#)

# Overriding at the application level

- Allows to override parts where bundle based paths are used (e.g. @OroUserBundle)
  - Routing
  - Controllers
  - Templates
- All custom files have to be at the src/Resources/<bundle> directory

[\[Documentation\]](#)

# Routes overriding

- Define routes with the same name
- The latest route wins
- Might be used to override controllers

# Template inheritance

- Uses `{% extends %}` tag
- Allows to redefine or extend blocks
- `parent()`

`{% extends 'OroUIBundle:actions:update.html.twig' %}`

<https://github.com/oroinc/platform/blob/3.1/src/Oro/Bundle/UserBundle/Resources/views/User/view.html.twig>



# Template placeholders

- New twig token parser
  - extends `\Twig_TokenParser`
- Uses `{% placeholder %}` tag
- Allows to inject required blocks at the defined positions (similar to events)

<https://github.com/oroinc/platform/blob/3.1/src/Oro/Bundle/UIBundle/Resources/views/actions/view.html.twig#L47-L54>

<https://github.com/oroinc/platform/blob/3.1/src/Oro/Bundle/DataAuditBundle/Resources/config/oro/placeholders.yml>

# Twig extensions

- Filters
  - `'label'|trans`
- Functions
  - `oro_currency_name(currency)`
- Tests
  - `value is defined`
- Operators
  - `value == value`
- Globals

<https://github.com/oroinc/platform/blob/3.1/src/Oro/Bundle/EntityBundle/Twig/EntityExtension.php>

# Form extensions

- Add features on top of existing types
  - Add specific feature to a single type
  - Add generic feature to several types
- Cases
  - Add new fields to forms
  - Process submitted data
  - Modify existing form behaviour

[\[Symfony documentation\]](#)

# Configuration files

- config/config.yml
- <bundle>/Resources/config/oro/app.yml
- php bin/console debug:config --help

<https://github.com/oro-inc/oro-commerce-application/blob/3.1/config/config.yml#L37-L43>

<https://github.com/oro-inc/oro-commerce/blob/3.1/src/Oro/Bundle/CMSBundle/Resources/config/oro/app.yml>

# Events

- Presented in all components
- Allow to change data or behaviour
- ORO Events, eg. Datagrid, Import, UI
  - `<bundle>/Event`
- Doctrine Events, eg. Lifecycle
- Symfony Events, eg. Form

<https://github.com/oroinc/platform/blob/3.1/src/Oro/Bundle/DataGridBundle/Datagrid/Builder.php#L112-L113>