

Products and Master Catalog

Products

Product

- Types - simple, configurable
- Images
 - AttachmentBundle, LiipImagineBundle
- Measure units
- Attributes
- Families
- OroProductBundle:Product [\[link\]](#)

[\[User guide\]](#)

Demonstration UI >>>

Configurable product

- Product with multiple custom variations (e.g. size, color, material, etc.)
- Group several simple products (configurable product variants)
- Prerequisites - Attributes, Product family, Product variants
- Matrix Grid form

[Documentation]

Demonstration UI >>>

Localizations

- Process of translating and adapting a product for a specific country
- Language
- Formatting
- Hierarchy

[Documentation]

Demonstration >>>

Localizable values

- String or text/html
- Default value
- Fallback to parent or default value
- OroLocaleBunde:LocalizedFallbackValue
- Product name example
- LocalizationHelper to get value

[Documentation]

Demonstration UI + DB >>>

Product Units

- Precision
- Conversion rate
- Sell
 - unless selected, the unit is considered to be an internal
- Single unit mode
 - limits the product unit to just one default value
- OroProductBundle:ProductUnit
- OroProductBundle:ProductUnitPrecision

Demonstration UI >>>

Product Images

- AttachmentBundle, LiipImagineBundle
- Types
 - Main image
 - Listing image
 - Additional images for gallery
- Watermarks
- Configuration
- OroProductBundle:ProductImage
- OroProductBundle:ProductImageType

Demonstration UI >>>

Product Families

- Represent product groups
- Define set of product attributes
- Attributes split into attribute groups
- Used during the product creation
- OroEntityConfigBundle:AttributeFamily
- OroEntityConfigBundle:AttributeGroup

[User guide]

Demonstration UI >>>

Product Attributes

- Represent product parameters
- By adding the product attributes only to the product families they fit, you can limit the product data to the necessary characteristics

TVs	T-shirts
name	
vendor	
resolution	size
matrix	color

Product Attributes

- Extended fields
- Frontend options
 - Searchable
 - Filterable
 - Sortable

[\[Documentation\]](#)

[\[User guide\]](#)

Demonstration UI + search >>>

Master Catalog

Master Catalog

- Represents main Category tree structure
- Product can be assigned to only one Category
- Allows to configure
 - Default product options
 - Visibility
 - SEO
- Can be used to build a menu [\[link\]](#) [\[link\]](#)
- [Gedmo nested tree](#)
- [OroCatalogBundle:Category](#)

Demonstration UI >>>

Category and Product SEO

- Meta title, meta keywords, meta description
- Rendered as meta tags `[link]`
- Searchable `[code]`
- Extended fields `[code]`
- OroSEOBundle

Demonstration UI + code + DB (website-search) >>>

Category and Product visibility

- Values: Hidden / Visible (plus fallbacks)
- Levels: Customer / Customer Group / Website (product only, EE only) / All
- Config values
- Fallbacks
- Resolved visibility
 - Cache & Calculation
- OroVisibilityBundle

Demonstration UI >>>

Website Search

- Sometimes referred as storefront/frontend index type
- Allows to use search on storefront
- Data is stored by websites
- Supports multiple entities (Product is used)
- Supports both sync and async reindexation
- Engine data collection is event based
 - Data is collected via listeners in batches
- Configuration in website_search.yml

Website Search

- Placeholders - “specific scope”
 - WEBSITE_ID
 - LOCALIZATION_ID
 - CUSTOMER_ID, and more
 - Developers may create custom placeholders - [example](#)
- oro:website-search:reindex
- ORM and Elasticsearch (EE only)
- [OroWebsiteSearchBundle](#)

[\[Admin guide\]](#)

Frontend Products datagrid

- Product search results at OroCommerce storefront
- Website Search based configuration
- Ways to customize
 - add data to website index
 - ex. manually
 - ex. product attributes (filterable)
 - mix data from other source example

Best Practices

- Add (or import) all your product attributes with all flags (searchable, filterable, sortable) at the very beginning to avoid useless search reindexation
- If you're planning to use single unit mode then it has to be enabled from the very beginning
- Product attribute has to be added to product family prior to its usage
- Changing of a product family for a product is highly unrecommended

Best Practices

- New fields can be added to products frontend grid either by adding them to search index and then adding them to the grid (select and column/property are required) or using result after event (column/property is required)