

# Entities and their data

# Entities

- PHP classes and objects
- Used to map data by ORM
- Have related repositories
- Define metadata using Doctrine annotations
- Might contain data storage logic
- Might contain lifecycle callbacks (events)
- Usually stored at <bundle>/Entity

<https://github.com/oroinc/platform/blob/3.1/src/Oro/Bundle/UserBundle/Entity/Group.php>

# Entity repositories

- Used to find and retrieve entities via ORM
- Might contain custom methods
- Usually stored at <bundle>/Entity/Repository
- Has to be requested from entity manager
- Default repository is  
Doctrine\ORM\EntityRepository
- LoadClassMetadataEventArgs
  - Useful for customizations

[\[Example\]](#)

# Quiz

- `$this`  
    `->getDoctrine()->getRepository(Config::class)`  
    `->findAll();`
- What will be the result of such a call?

*Demonstration >>>*

# Entity managers

- Used to manage ORM data (including entities and repositories)
- One entity manager per DB connection
- Has to be requested from ManagerRegistry

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

# Avoid using default manager

- `$this`  
`->getDoctrine()->getRepository(Config::class)`  
`->findAll();`
- The class 'Oro\Bundle\ConfigBundle\Entity\Config' was not found in the chain configured namespaces

*Demonstration >>>*

# Get entity using ManagerRegistry

```
/** @var \Doctrine\Common\Persistence\ManagerRegistry $registry */  
$registry = $this->container->get('doctrine');  
$user = $registry->getManagerForClass('OroUserBundle:User')  
->getRepository('OroUserBundle:User')  
->find(1);
```

# Get entity using DoctrineHelper

```
/** @var \Oro\Bundle\EntityBundle\ORM\DoctrineHelper $doctrineHelper */  
  
$doctrineHelper = $this->container->get('oro_entity.doctrine_helper');  
  
$user = $doctrineHelper->getEntityRepository('OroUserBundle:User')  
  
->find(1);
```



# Entity configuration

- Stores metadata related to entities and entity fields
  - Eg. attachments, auditable (field)
- Data is stored into fields grouped into scopes
  - Resources/config/oro/entity\_config.yml
- Defined using Oro annotations @Config and @ConfigField
- Developer may add custom scopes and fields

<https://github.com/oro-inc/crm/blob/3.1/src/Oro/Bundle/ContactBundle/Entity/Contact.php>

# Entity configuration: CLI commands

- `bin/console oro:entity-config:update`
  - Updates configuration data for entities.
- `bin/console oro:entity-config:cache:clear`
  - Clears the entity config cache.
- `bin/console oro:entity-config:cache:warmup`
  - Warms up the entity config cache.

# Entity configuration: UI

- Attachments

- Contact entity - view page
- Enable attachments
- Update schema + (optional) show DB
- Add attachment to Contact

- Auditable

- Edit firstName
- Show change history
- Disable auditable
- Show change history

# Extended entities

- Allow to add custom fields and relations to existing entities
- Intermediate layer with custom data is represented by a class in `<bundle>/Model/`
- Real autogenerated classes are stored at `var/cache/<env>/oro_entities/Extend/Entity`

[\[Video presentation\]](#)

# Extended entities: Code

- Examples
  - Add field to an extended entity
  - Add relation to an extended entity
- Extend extension

# Extended entities: UI

- Contact - UI
  - Create field products (one-to-many to Product)
  - Edit any Contact -> add products
- Contact - code
  - cache

<https://github.com/oroinc/crm/blob/3.1/src/Oro/Bundle/ContactBundle/Entity/Contact.php>

# Custom entities

- Allow to add completely new entities with custom fields and relations
- Custom entities work the same way as extended entities
- Simple CRUD out of the box

# Custom entities: UI

- Create new entity
- Show CRUD
- Create relation Contact -> New entity (many-to-one)
- Assign to Contact
- Show view, grid, inline edit
- Show cache
- Show DB



# Migrations

# Bundle installation

- Tool for DB structure and data migrations
  - Only “up” method (Migration interface)
- Bundle version
  - OroPlatformBundle v1\_1
- Installer
- Migrations
- Data fixtures
- Demo data fixtures

*Demonstration >>>*

# Bundle version

**v1\_0**

Installer v1_0
Migration v1_0

**v1\_1**

Installer v1_1	
Migration v1_0	Migration v1_1

**v1\_2**

Installer v1_2		
Migration v1_0	Migration v1_1	Migration v1_2

# Installer

- Contains version (preferably the latest one)
- Set up DB schema for a bundle
- Might contain queries and extensions
- No interaction with ORM
- Migrations/Schema/<bundle>Installer.php
  - MigrationsLoader
  - MigrationExecutor

<https://github.com/oroinc/orocommerce/blob/3.1/src/Oro/Bundle/ProductBundle/Migrations/Schema/OroProductBundleInstaller.php>

# Migrations

- Adjust existing DB schema
- Might contain queries and extensions
- No interaction with ORM
- Migrations/Schema/<version>/<file>.php
- OrderedMigrationInterface
- preQueries and postQueries
- bin/console oro:migration:load

[https://github.com/oroinc/orocommerce/blob/3.1/src/Oro/Bundle/ProductBundle/Migrations/Schema/v1\\_9/MakeVisibleDefaultProductAttributes.php](https://github.com/oroinc/orocommerce/blob/3.1/src/Oro/Bundle/ProductBundle/Migrations/Schema/v1_9/MakeVisibleDefaultProductAttributes.php)

# Data fixtures

- Doctrine DataFixtures
- No version (usually)
- Interaction with ORM
- Types
  - Dependent, ordered, versioned
- Migrations/Data/ORM/<file>.php
- bin/console oro:migration:data:load

<https://github.com/oroinc/orocommerce/blob/3.1/src/Oro/Bundle/ProductBundle/Migrations/Data/ORM/LoadProductUnitData.php>

# Demo data fixtures

- Load demo entities
  - Useful for demo with the client
  - Show the features with generated data
- Migrations/Data/Demo/ORM/<file>.php
- `php bin/console oro:migration:data:load --fixtures-type=demo`

<https://github.com/oroinc/orocommerce/blob/3.1/src/Oro/Bundle/ProductBundle/Migrations/Data/Demo/ORM/LoadProductDemoData.php>

# MigrationBundle

- <https://github.com/oro-inc/platform/tree/3.1/src/Oro/Bundle/MigrationBundle>
- <https://oro-inc.com/orocrm/doc/current/dev-guide/entities/data-fixtures>



# Datagrid

# Datagrid

- Table oriented representation of *some data* from *some datasource*
- Several data sources (including ORM)
- Builds ORM query based on configuration
- Applies filtering, sorting and pagination
- Supports inline editing
- Allows to execute single and mass actions

*Demonstration UI >>>*

# Configuration of datagrid

- ORM query - select, from, where, groupBy
- Columns - label, frontend type
- Filters - type, data name
- Sorters - data name, default
- Properties
- Single actions and mass actions
- Resources/config/oro/datagrids.yml

<https://github.com/oro-inc/platform/blob/3.1/src/Oro/Bundle/UserBundle/Resources/config/oro/datagrids.yml>

# Customization of datagrid

- Merge grid configuration - be careful
- Grid extends grid
- Grid events
  - Builder - preBuild, buildBefore, buildAfter
  - Datasource - resultBefore, resultAfter
- Grid extensions

<https://github.com/oro-inc/orocommerce/blob/3.1/src/Oro/Bundle/CatalogBundle/EventListener/DatagridListener.php#L50-L54>

<https://github.com/oro-inc/platform/blob/3.1/src/Oro/Bundle/EntityPaginationBundle/Datagrid/EntityPaginationExtension.php>

# Datagrid - exercise

- Disable firstName filter from Contacts grid
- You can use the easiest way

PS. You don't need to clear the cache :)

# Entity index page

- Route without parameters
- Contains datagrid and actions for all entities
- Uses index template

<https://github.com/oroinc/platform/blob/3.1/src/Oro/Bundle/UserBundle/Controller/GroupController.php#L71-L76>

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

# Entity CRUD

- Create
- Read
- Update
- Delete

# Read: Code

- Get specific entity from a repository
- `EntityRepository::find*`
- Read entities are stored in `UnitOfWork`

<https://github.com/oroinc/crm/blob/3.1/src/Oro/Bundle/DemoDataBundle/Migrations/Data/Demo/ORM/LoadGroupData.php#L67>



# Read: Controller

- Route with entity ID
- Entity is passed as an argument
- Uses view template

<https://github.com/oroinc/platform/blob/3.1/src/Oro/Bundle/UserBundle/Controller/UserController.php#L42-L48>

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

# Create: Code

- Create entity object
- EntityManager::persist
- EntityManager::flush
- Created entities are stored in UnitOfWork

<https://github.com/oroinc/crm/blob/3.1/src/Oro/Bundle/ContactBundle/Migrations/Data/ORM/LoadSourceData.php>

# Create: Controller

- Route without parameters
- Create new entity object and submit form
- Usually uses update template

<https://github.com/oroinc/platform/blob/3.1/src/Oro/Bundle/UserBundle/Controller/UserController.php#L124-L129>

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

# Update: Code

- Get entity
- Change entity
- EntityManager::flush

<https://github.com/oroinc/platform/blob/3.1/src/Oro/Bundle/UserBundle/Migrations/Data/ORM/UpdateUserEntitiesWithOrganization.php#L39-L56>

# Update: Controller

- Route with entity ID
- Entity is passed as argument
- Uses update template

<https://github.com/oroinc/platform/blob/3.1/src/Oro/Bundle/UserBundle/Controller/UserController.php#L146-L149>

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

# Delete: Code

- Get entity
- EntityManager::remove
- EntityManager::flush
- Deleted entities are removed from  
UnitOfWork

<https://github.com/oroinc/crm/blob/3.1/src/Oro/Bundle/ContactBundle/Form/Handler/ContactEmailHandler.php#L111-L113>

# Delete: Controller

- No need to have it
- Actions added automatically
  - or via API, e.g. oro\_api\_delete\_contact
  - [More info]

# Validation

- Validates entity data
- Can be assigned to fields or whole entity
- Developer might add custom constraints
- Triggered automatically on form submission
- Can be triggered using validator service
- Resources/config/validation.yml

<https://github.com/oro-inc/crm/blob/2.6/src/Oro/Bundle/ContactBundle/Resources/config/validation.yml>



# Menu

- Uses KnpMenuBundle
- ORO: NavigationBundle
- Represented by a tree structure
- Label
- Route
- Position
- Resources/config/oro/navigation.yml

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

# Customization of menu

- UI
- Config - Resources/config/oro/navigation.yml
- ConfigureMenuEvent
- Menu extension
  - oro\_navigation.menu\_extension
- Menu builder
  - oro\_menu.builder
- + page titles

<https://github.com/oroinc/platform/blob/3.1/src/Oro/Bundle/NavigationBundle/Resources/config/services.yml>

# Menu - exercise

- Add via navigation.yml new menu item
- Label: 'My custom contacts'
- Position: just before 'Contacts'
- Route: oro\_contact\_index

# Search

- Creates search index for entities and allows to perform search using this index
- Specify fields and relations used to search
- Allows to add custom/modify data
  - `oro_search.prepare_entity_map`
- Allows to build custom queries
  - [example](#)
- `Resources/config/oro/search.yml`
  - [example](#)

# Search demo

- UI - quick search
- API - simple + advanced
  - `search=Debra from=oro_contact`
  - `select firstName from oro_contact where firstName ~ 'Debra'`
- DB - structure
- `bin/console oro:search:reindex`
- QueryBuilder
  - `Oro\Bundle\SecurityBundle\Search\AclHelper::apply`

# Search docs

- <https://github.com/oro-inc/platform/tree/3.1/src/Oro/Bundle/SearchBundle>
- [Search Component Architecture](#)
- [Dealing with the Search Index](#)
- [Configuration file structure](#)

# Autocomplete

- Uses search index as a data source by default
- Can be customized by developer
- Used by form type to select entities
  - Form type example
  - Autocomplete handler example

*Demonstration >>>*

# Name provider

- Direct text representation of entity object
- DQL string to get text representation from DB
- Used all over the system
- Developer can add custom name providers

<https://github.com/oroinc/crm/blob/3.1/src/Oro/Bundle/ContactBundle/Provider/ContactEntityNameProvider.php>



# Name provider demo

- UI - contact
- ContactEntityNameProvider
- Twig: entity|oro\_format\_name

<https://github.com/oroinc/platform/blob/3.1/src/Oro/Bundle/LocaleBundle/Resources/doc/reference/name-formatting.md>

<https://oroinc.com/orocrm/doc/current/admin-guide/localization>

# Entity aliases

- Simple and elegant way of referring to entities
- Text representation of entity class
- Have singular and plural forms
- Used all over the system
  - API, dictionaries, pass entity class via URL
- Resources/config/oro/entity.yml

[https://github.com/oroinc/platform/blob/3.1/src/Oro/Bundle/EntityBundle/Resources/doc/entity\\_aliases.md](https://github.com/oroinc/platform/blob/3.1/src/Oro/Bundle/EntityBundle/Resources/doc/entity_aliases.md)

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

# Virtual fields and relations

- Additional fields or relations
- Can be used only to view data
- Good for reports or API
- `Resources/config/oro/entity.yml`

<https://github.com/oro-inc/crm/blob/3.1/src/Oro/Bundle/ContactBundle/Resources/config/oro/entity.yml#L2-L38>