

Duplicate role

role, duplicate, processor, state

Role is duplicated by RoleDuplicateBulkAction bulk action. This action propagate DUPLICATE event type with content:

- original source - contains the selected original role
- content - contains duplicated role

Creating duplicate role is then propagated into registered (and enabled) entity event processors. Custom processors can be registered by the different module (or product processor can be disabled and overridden, see below).

Processors can register form attributes into the bulk action form - distinct form attributes (by code, the first attribute wins by processor's order) are rendered and used as input for the bulk action.



Form attributes registered by processor can be localized in the custom module. Each attribute can have different module - module is preset automatically by the processor's module.



If role is duplicated into different environment and role with the same base code already exists there, then new role on the target environment is not created, but updated by the source (selected) role. E.g. when some basic role attribute is changed or some automatic role is added ⇒ this changes can be "synchronized" into target role repetitively. **Roles are paired between environments by the base code** ⇒ if base code is changed, then new duplicate will be created.

Processors

Implemented processors in the product sorted by order of the processing:

DuplicateRolePrepareProcessor

@since 9.5.0

- Event content: IdmRoleDto
- Event type: DUPLICATE
- Default order: **-1000**

Prepares role's basic properties.



Register custom processor after this processor's order, if some role basic property has to be overridden (or filled by different business logic).

```
## Enable / disable  
idm.sec.core.processor.core-duplicate-role-prepare-processor.enabled=true
```

DuplicateRoleSaveProcessor

@since 9.5.0

- Event content: IdmRoleDto
- Event type: DUPLICATE
- Default order: **0**

Here is the role persisted into database.



Register custom processor after this processor's order, if some related entities has to be duplicated (e.g. guarantees).

```
## Enable / disable  
idm.sec.core.processor.core-duplicate-role-save-processor.enabled=true
```

DuplicateRoleFormAttributeProcessor

@since 9.5.0

- Event content: IdmRoleDto
- Event type: DUPLICATE
- Default order: **50**

Duplicate role form attributes - parameters for the identity (~assigned) roles. Parameters are created for the target role or updated - extended attribute code is used for pairing.

Parameters provided to the bulk action form:

- **Duplicate role form attributes** - if role form attributes will be duplicated.

Configuration properties:

```
## Enable / disable  
idm.sec.core.processor.core-duplicate-role-form-attribute-processor.enabled=true
```

DuplicateRoleCompositionProcessor

@since 9.5.0

- Event content: IdmRoleDto
- Event type: DUPLICATE

- Default order: **100**

Duplicate configured role composition (sub roles by business role definition) and duplicate sub roles recursively. If the same environment is selected, the only role composition is created - existing sub roles are used. If the different environment (~target environment) is used, then sub roles with the same environment as original are duplicated recursively into target environment.

Parameters provided to the bulk action form:

- **Duplicate sub roles (by business role definition)** - if business role configuration will be duplicated (recursively).

Overidable methods (can be used for on the projects, e.g. example below):

- `duplicateRecursively` - Returns true, when role should be cloned recursively - can be overridden, if some role hasn't be cloned recursively, if doesn't exist on the target environment before.
- `includeComposition` - Returns true, when role composition should be included in the target role - can be overridden, if some role hasn't be cloned recursively, if doesn't have the same environment etc.

Configuration properties:

```
## Enable / disable
idm.sec.core.processor.core-duplicate-role-composition-
processor.enabled=true
```

Custom processor example

Core processor can be disabled and overridden by processor implemented in custom module, if behavior of the core processor has to be changed.

```
/**
 * Project specific processor for duplicate role composition.
 */
@Component(CustomDuplicateRoleCompositionProcessor.PROCESSOR_NAME)
@Description("Duplicate role - composition and recursion.")
public class CustomDuplicateRoleCompositionProcessor extends
DuplicateRoleCompositionProcessor {

    public static final String PROCESSOR_NAME = "custom-duplicate-role-
composition-processor";

    @Override
    public String getName() {
        return PROCESSOR_NAME;
    }

    /**
     * Returns true, when role should be cloned recursively
```

```
* - it's not cloned, if application sub role doesn't exist on the
target environment before.
*
* @param event processed event
* @param originalSubRole original sub role
* @param targetSubRole duplicate sub role. {@code null} if target role
has to be created.
* @return
*/
@Override
public boolean duplicateRecursively(EntityEvent<IdmRoleDto> event,
IdmRoleDto originalSubRole, IdmRoleDto targetSubRole) {
    return (targetSubRole != null && targetSubRole.getId() != null) ||
originalSubRole.getChildrenCount() > 0;
}

/**
 * Returns true, when role composition should be included in the target
role
 * - it's not included, when sub role doesn't have the same environment
 *
 * @param event processed event
 * @param composition source composition
 * @return
 */
@Override
public boolean includeComposition(EntityEvent<IdmRoleDto> event,
IdmRoleCompositionDto composition) {
    IdmRoleDto subRole = DtoUtils.getEmbedded(composition,
IdmRoleComposition_.sub);
    //
    return Objects.equals(event.getOriginalSource().getEnvironment(),
subRole.getEnvironment());
}
}
```

DuplicateRoleAutomaticByTreeProcessor

@since 9.5.0

- Event content: IdmRoleDto
- Event type: DUPLICATE
- Default order: **200**

Duplicate configured automatic roles by tree structure. Automatic roles are duplicated recursively, if composition is duplicated recursively (see DuplicateRoleCompositionProcessor above).

Parameters provided to the bulk action form:

- **Duplicate automatic roles** - if automatic roles will be duplicated (both by tree structure and

attribute).

Configuration properties:

```
## Enable / disable
idm.sec.core.processor.core-duplicate-role-automatic-by-tree-processor.enabled=true
```

DuplicateRoleAutomaticByAttributeProcessor

@since 9.5.0

- Event content: IdmRoleDto
- Event type: DUPLICATE
- Default order: **300**

Duplicate configured automatic roles by attribute. Automatic roles are duplicated recursively, if composition is duplicated recursively (see DuplicateRoleCompositionProcessor above).

Parameters provided to the bulk action form:

- **Duplicate automatic roles** - if automatic roles will be duplicated (both by tree structure and attribute).

Configuration properties:

```
## Enable / disable
idm.sec.core.processor.core-duplicate-role-automatic-by-attribute-processor.enabled=true
```

Example processor

Processors can be registered to process event with type DUPLICATE and IdmRoleDto content.

```
/**
 * Duplicate role - example processor, just create log
 */
@Enabled(ExampleModuleDescriptor.MODULE_ID)
@Component(DuplicateRoleLogProcessor.PROCESSOR_NAME)
@Description("Duplicate role - composition and recursion.")
public class DuplicateRoleLogProcessor
    extends CoreEventProcessor<IdmRoleDto>
    implements RoleProcessor {

    private static final org.slf4j.Logger LOG =
org.slf4j.LoggerFactory.getLogger(DuplicateRoleLogProcessor.class);
    //
    public static final String PROCESSOR_NAME = "example-duplicate-role-log-
```

```
processor";
    public static final String PARAMETER_INCLUDE_LOG = "include-log";

    public DuplicateRoleLogProcessor() {
        super(RoleEventType.DUPLICATE);
    }

    @Override
    public String getName() {
        return PROCESSOR_NAME;
    }

    /**
     * Adds form attribute - if log will be created - into bulk action form.
     */
    @Override
    public List<IdmFormAttributeDto> getFormAttributes() {
        IdmFormAttributeDto include = new IdmFormAttributeDto(
            PARAMETER_INCLUDE_LOG,
            "Log duplicated roles",
            PersistentType.BOOLEAN);
        include.setDefaultValue(Boolean.TRUE.toString());
        //
        return Lists.newArrayList(include);
    }

    @Override
    public boolean conditional(EntityEvent<IdmRoleDto> event) {
        return super.conditional(event)
            && getBooleanProperty(PARAMETER_INCLUDE_LOG,
event.getProperties());
    }

    @Override
    public EventResult<IdmRoleDto> process(EntityEvent<IdmRoleDto> event) {
        IdmRoleDto duplicate = event.getContent();
        IdmRoleDto originalSource = event.getOriginalSource();
        //
        LOG.info("Duplicate role [{}] from environment [{}] to [{}]",
originalSource.getBaseCode(), originalSource.getEnvironment(),
duplicate.getEnvironment());
        //
        return new DefaultEventResult<>(event, this);
    }

    @Override
    public int getOrder() {
        return 10000; // on the end
    }
}
```

Entity state usage

When some role composition or automatic role (~related entity) is removed from the source role and role is duplicated into different environment repetitively (⇒ update), then all removed related entities are removed at the end of the bulk action to prevent some account on target system (acc) will be removed and created again. When bulk action is processed (by processors) and some related entity has to be removed, then entity state `deleted` is created for this entity only. All entities with the `deleted` state are removed at end. States are marked with `transactionId`, which creates envelope for all states created by one bulk action ⇒ only states for the current bulk action are processed.

From:

<https://wiki.czechidm.com/> - **CzechIdM Identity Manager**

Permanent link:

<https://wiki.czechidm.com/devel/documentation/roles/dev/duplicate-role>

Last update: **2019/03/18 08:50**

