# Agenda of universal requests

universal, request

## How it works?

When the approval mode is turned on, it is not possible to edit the object over the standard **REST** interface (for example '**/api/v1/roles**'), but you must use the REST for universal requests (for example'**/api/v1/requests/{request ID }/roles**').

In the approval mode, **it is not possible to use the standard REST** interface. This restriction only applies to the **REST interface**. Editing through a given **service is fully functional**.

From a **UI** perspective, the situation is similar. If the approval mode is enabled, it is not possible to edit the object on a standard URL. This means that even if the user has the right to edit, the object will be **read-only** (details will be read only, edit buttons will not be available, bulk operations will not be available). **An object can only be edited after you have moved to a specific request URL**.

**One of the main objectives of universal applications was to achieve the reuse of the UI components**. This means that if a user creates a request (which goes to a completely different URL), they should not visually see "no" change. They should feel that by simply creating a request, the form just switched to editing mode.

Example of a URL role and the same role edit role within the request:

* **/role/**{role ID}**/detail** * **/requests/**{request ID}**/role/**{role ID}**/detail**

### Creation of a request

If you want to go to the above mentioned URLs, **you will first need to create the request**. This can be generated by calling the **method on the REST request interface of the object**. This is a **POST** method where a single entry is the **object for which we want to create a request**. This input object may not exist in the database (for a situation where we want to create a new object within the request).

**Example** creating a **request** for a new **role**:

```
curl 'http://localhost:8080/idm-backend/api/v1/requests/roles' -H 'content-type: application/hal+json;charset=UTF-8' --data-binary '{"name":"NewRole","code":"NewRole"}'
```

### Creation of request items

If we already have a request, we can start making individual **changes**. As described above, individual **REST** request calls are "redirected" to the REST request interface of that object type. Each partial change (called REST interface) creates a request item (**IdmRequestItem**). This item includes, in particular, ownership of the owner, that is, the link to the object being edited.

Additionally, the request entry contains a **complete object** in the form of how to get from the interface. This object is used to apply changes when applying for approval. The object is saved in **JSON** format. This format has been chosen with respect to backward compatibility. There is a lower probability of any problems with a change in the structure of the target object. **JSON** format allows us to perform additional transformations (to ensure compatibility) against binary serialization of the whole object.

The **REST** interface is represented by the controller, for example, the role is the **standard** controller **IdmRoleController** and the **request** controller is **IdmRequestRoleController**.

## How to enable the possibility of requesting a specific object?

Requesting mode is controlled for all requestable objects by **IdmRoleDto** (for now)!

Requesting mode can be enabled for every supported object by property in the application configuration:

```
idm.pub.core.request.<requestable object>.enabled=true
```

, where **<requestable object>** is the name of requestable object (DTO).

For example **approving for role** (IdmRoleDto) can be enable by this property:

```
idm.pub.core.request.idm-role.enabled=true
```

## Supported requestable objects

All supported objects must implemented interface `eu.bcvsolutions.idm.core.api.domain.Requestable`.

- IdmRoleDto - **Roles**
- IdmRoleCompositionDto - **Business roles**
- IdmRoleGuaranteeDto - **Guarantee defined by identity**
- IdmRoleGuaranteeRoleDto - **Guarantee defined by other role**
- IdmAuthorizationPolicyDto - **Premissions**
- IdmRoleCatalogueRoleDto - **Relations on the catalogue**
- SysRoleSystemDto - **Linking the system to a role**
- SysRoleSystemAttributeDto - **Overloaded system attribute** (on the role)
- IdmFormValueDto - **Extended attributes**

# Workflow processes

Each requestable type of object may have its own approval process. The approval process can be defined in the application configuration:

```
idm.sec.core.request.{object type}.wf={code of WF process}
```

**Example for IdmRole**:

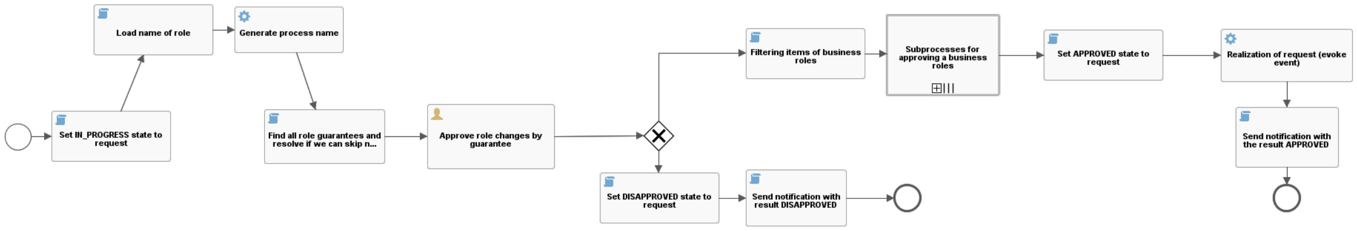
```
idm.sec.core.request.idm-role.wf=request-idm-role
```

If none workflow process is defined, then '**request-idm-role**' is using as default.

## Workflow process for roles

The basic approval process where changes on the role are approved by the guarantors of the role in request. If there are no guarantors or a new role, then those who have the role defined in the variable '**core.wf.approval.role-change.role**' are the approvers. If this variable does not have any role, '**superAdminRole**' is returned. The result of the approval process is sent to the applicant for changes via email notification. Sending of these notifications can be disabled by setting the corresponding notification topic to inactive state. Topic `core:approveRoleDefinitionChange` for approved requests and `core:disapproveRoleDefinitionChange` for disapproved ones.

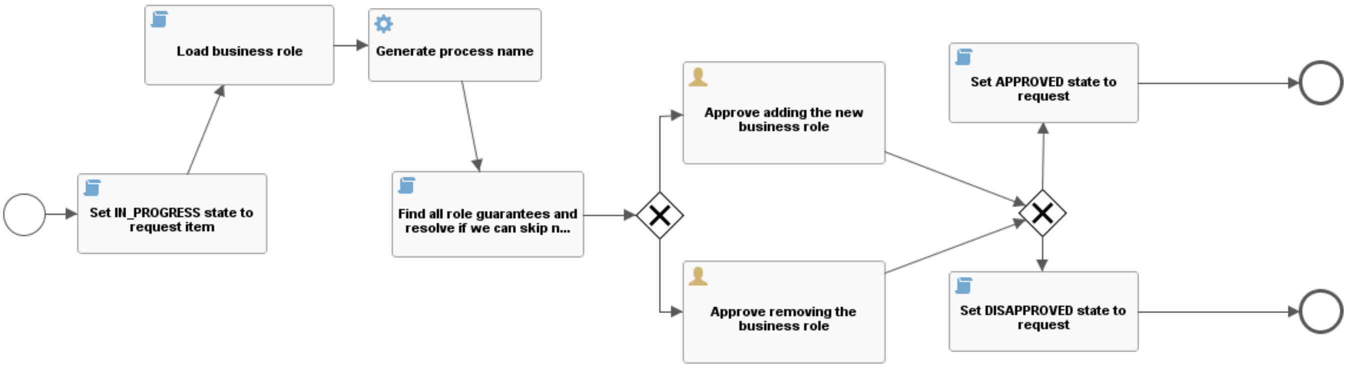
```
core.wf.approval.role-change.role=superAdminRole
```



Set IN_PROGRESS state to request
Load name of role
Generate process name
Find all role guarantees and resolve if we can skip n...
Approve role changes by guarantee
Set DISAPPROVED state to request
Send notification with result DISAPPROVED
Filtering items of business roles
Subprocesses for approving a business roles
Set APPROVED state to request
Realization of request (evoke event)
Send notification with the result APPROVED

### Workflow process for business roles

If the request contains a item with **business role** change (**IdmRoloComposition**), then a separate workflow process (**request-idm-role-composition**) is started for each item. This process aims to verify whether the guarantors of the target role agree with the change. This means that if we put **B** role to role **A**, guarantor of the role B must agree with this assignment. The same applies to the removal of the relation. **If the target role has no guarantors**, then the change is considered **approved**.


Set IN_PROGRESS state to request item
Load business role
Generate process name
Find all role guarantees and resolve if we can skip n...
Approve adding the new business role
Approve removing the business role
Set APPROVED state to request
Set DISAPPROVED state to request

### Configurable role guarantor type for role change request

role, request, guarantee, type

By default a role is approved by any guarantor defined for that role. **If you need to restrict approval to only guarantors with a specific type**, then you can use the configuration item **idm.sec.core.request.idm-role.approval.guarantee-type**, where the value specifies the **type of guarantor**.

If the value is not defined or item does **not exist**, then **all guarantors are used for approval**, regardless of what type they have defined. The described behavior is the same for guarantors defined by **identity** or **role**.

**By default**, this configuration item (**idm.sec.core.request.idm-role.approval.guarantee-type**) is empty. This means that approval will be run with all guarantors (regardless of their type).

# Limitations

Enabling of the request mode is controlled only by **IdmRole** now.

Changes in the request preview are highlighted only on tables. Type of changes are not show on the object **details** or on **EAVs**!