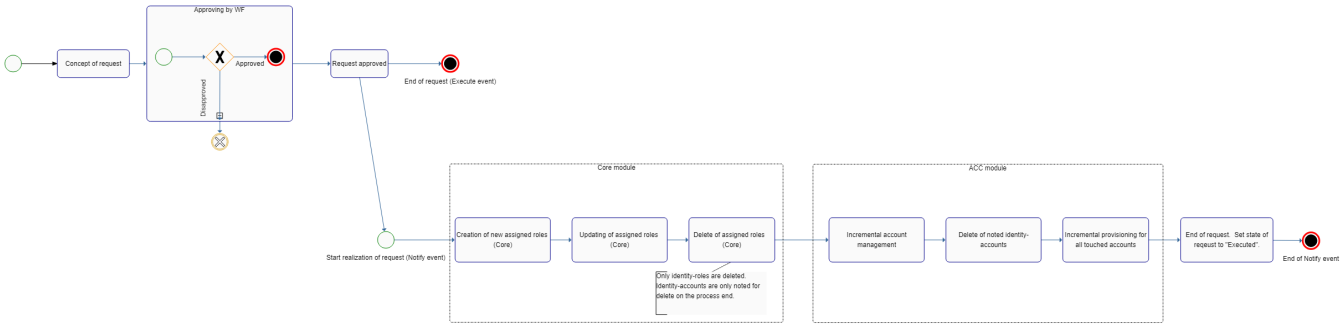
Changing user permissions

identity, permission, role, request, security, workflow

Introduction to the topic - read here.

Basic life cycle of the request for change a permissions



A manual change (via REST api) of permission, i.e. adding/editing/removing a link between an identity and a role is not possible. All permission changes must be made via the role request agenda (RoleRequest).

Role request agenda

This agenda contains all requests (wishes) for requested changes of authorized identities. The main idea is that all changes in identities' permission must go through this agenda. Therefore, it is not intended only for end users' requests but for automatic operations (synchronization, automatic roles, etc.) as well.

The standard scenario of a permission change is as follows:

1. The user request a change in his permission.
2. He presses the "**Change permission**" button in the "**Assigned roles**" tab.
3. A new request (in "**CONCEPT**" state) is created automatically. The detail of this new request is shown to the user.
4. All the assigned roles which the user has are displayed in the table of concepts "Currently assigned roles (including requested changes)". The user can make the changes over this table (see more in **Concept table**)
5. All the requested changes of permission are displayed in the table "**Requested permission changes**".
6. The user presses the button "**Make a request**".
7. If the request is not evaluated as duplicate, its execution is performed (the state is set to "**IN_PROGRESS**").

8. Subsequently, the event "**RoleRequestEventType.EXCECUTE**" is called out.
9. Ordinarily, this event is firstly captured by the processor "**role-request-approval-processor**", which starts the approval workflow (calls the method **IdmRoleRequestService.startApprovalProcess**). In the default state, the workflow with the definition "**approve-identity-change-permissions**" is started. The key of the definition which is to be started can be changed in the application configuration by adding the key "**idm.sec.core.processor.role-request-approval-processor.wf**", (the value will be the key of the requested definition).
10. After a successful approval, the workflow process calls out the event "**RoleRequestEventType.EXCECUTE**" again (it continues with the event with which it has been started).
11. Ordinarily, the event is then captured by the processor "**role-request-realization-processor**", which provides the realization of the event itself. The processor calls the method **IdmRoleRequestService.executeRequest(requestId)**, which performs the application of all role concepts which are in the "**APPROVED**" or "**CONCEPT**" states (the concept state is realized due to the situation when the realization processor is called out immediately after making the request, i.e. approval is disabled).
12. **The user is assigned roles according to the requested changes**.

If the user creates a new request but does not submit it, it will be displayed as "**Concept**" in the table "**Unfinished requests for permission change**" in the assigned roles tab. Thus they can return to an unfinished concept.

Concept table

The goal is to provide the user with a comfortable way of creating the requested permission changes (they can see all the change in one location, projected into their currently assigned roles).

- The input parameter of the concept table is a list of all currently assigned roles of the user (they are not coloured).
- If the user adds new permission (role + date of validity), it will be displayed in the table a new lines (in green colour).
- If the user removes permission, it will not disappear from the table, its state will only change (in red colour)
- If the user edits permission (currently, only the date of validity can be changed), the permission will be displayed in orange colour. The tooltip over the edited cell will display the original value.
- All edits can be removed from the concept table (changes, cancellation of removal, removal of newly-added permissions).

The request can be submitted only when the request is in the "**CONCEPT**", "**DUPLICATED**" or "**EXCEPTION**" states

Changing permission without approval

There are two ways of how to make a change in permission without approval.

Disabling the approval processor

As mentioned above, after submitting a request, an event is called out which is captured by processor "**role-request-approval-processor**". If we disable this processor, all submitted requests will be realized immediately without any approval.

By disabling the approval processor, all submitted requests will be realized immediately for all users in the system regardless of their permissions.

Request attribute "Execute immediately"

A request for permission change contains attribute "**executeImmediately**". If this attribute is set to "true", the request will be executed immediately after submitting it. A submission without approval can be executed only by users having permission "**ROLEREQUEST_EXECUTEIMMEDIATELY**". If an event marked in this way is executed by a user without this permission, an exception will be called out.

Systematically, it is possible to execute a submission of an event without approval even without having the required permission. Method "**IdmRoleRequestService.startRequestInternal**" serves this purpose. This method is not available through the REST interface.

Duplicate request

Since version 9.6.0, the check for duplicate requests has been removed for performance!

During the process of submitting a request, it is verified if there is another duplicate request of the new request (in state "**IN_PROGRESS**", "**APPROVED**"). If a duplicate request is found, the executed request is labelled as duplicated (state "**DUPLICATED**") and its attribute "**duplicatedToRequest**" is filled with the identifier of the duplicate request (a record is made in the request log). The submission (execution) of the request is thus suspended.

A duplicate request is such a request with equivalent:

- applicants.
- individual role concepts (identical role type, identical validity from/to, identical link

IdentityRole).
- a request note (the user can define their request only in this note).

A duplicate request can be submitted (executed) repeatedly. Due to this, duplicate requests are found again.

Request removal

A request can be removed completely only in "**CONCEPT**" state. If it is in "**EXECUTED**" state, every attempt to remove it ends in exception "**ROLE_REQUEST_EXECUTED_CANNOT_DELETE**". If it is in "**APPROVED, IN_PROGRESS, EXCEPTION, DUPLICATED**" states, the request is set to "**CANCELED**" state. If a process is tied to the request, the process is terminated (subsequently a record about this is made in the request log).

Request agenda for administrators

A situation when the end user creates a request for permission change from their profile has been described above. For administrators, it is convenient to use the "**Role requests**" agenda, which provides an overview of all requests in the system. All requests in all states are displayed in this agenda (including those already executed).

The agenda allows a direct request submission from a list of requests. The agenda also allows **creating a new request**, where the administrator must first choose the user who the request is being created for.

In addition, the request detail in this agenda contains (compared to the end user request detail) a possibility of ticking that the request is not to be approved. The request detail also contains a **log**. This log contains all crucial information which occur during the life cycle of the request (errors, duplicates, request cancellation due to integrity, executed operations, etc.).

Integrity on remove contract or role

When role or contract (using in the some concept) is deleted and workflow process for that concept is not ended, the must be that process terminated. We cannot use standard 'cancel' of subprocess, because this operation does not triggered the main process.

This cause the main process will be **frozen**, because from his view canceled subprocess never ended.

For prevent this situation is concept's subprocess not canceled, but **disapproved** (on delete role or contract).

It means if current process task has decision with ID '**disapprove**', then is used. When disapprove decision is not found, the standard cancel of process is called.

REST interface

The creation of a request via a REST interface has the following steps:

1. **Creation of the request**
2. **Creation of the role concepts** (requested permission changes)
3. **Execution of the request** (submitting the request)

Creation of the request

A request can be created via method **POST** on endpoint "**/api/v1/role-requests/**".

Example of a request:

```
{
  "applicant" : "846164b4-8272-46a7-ba69-0bfdad652aff",
  "requestedByType" : "MANUALLY",
  "conceptRoles" : [],
  "executeImmediately" : false,
  "description" : "Please check and approve the permission change"
}
```

Creation of the role concepts

A role concept can be created via method **POST** on endpoint "**/api/v1/concept-role-requests/**".

Example of a request (creation of a concept for assigning a new role):

```
{
 "roleRequest":"740d7dc3-d90b-40cd-80f9-80cf36180c80",
 "identityContract":"7fdf1128-ce2d-4d34-ba99-821a544ad753",
 "role":"6ac36596-4f68-4282-88de-685e654199d6",
 "identityRole":null,
 "roleTreeNode":null,
 "validFrom":null,
 "validTill":"2017-07-31",
 "operation":"ADD"
}
```

Execution of the request

Execution of the request can be done via method **PUT** on endpoint "**/api/v1/role-requests/{requestId}/start**", where **requestId** is the UUID of the created request.

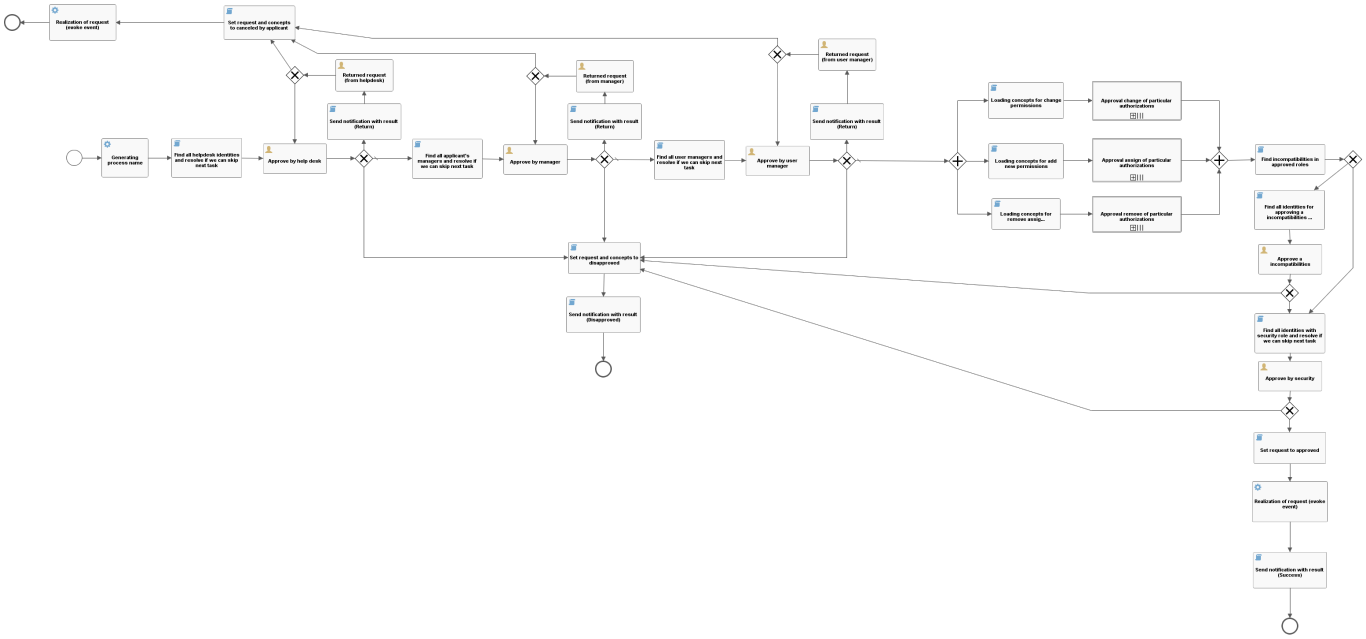
During the approval process, the requested changes can be changed (e.g. the manager disagrees with assigning of the role and removes it from the request). For easy backtracking of how the original request looked like, it is converted to JSON format at its start and saved to the request.

The approval process

If the request-permission-change-without-approval mode is not used, process "approve-identity-change-permissions" will be started. This process ensures the approval of the request as a whole and its base implementation is composed of these parts:

1. **Process name generating** (the applicant's name will also be stated in the name)
2. **Approval by the helpdesk department**.
3. **Approval by the manager of the applicant**.
4. **Approval by the user administration department**.
5. **Execution of subprocesses for each requested role**.
6. **Approval of role incompatibilities**
7. **Approval by the security department**.
8. **Sending the notification**.
9. **Realization of the request** - the realization itself is not carried out by the process, but by the service for managing requests for permission change.

The input of the proces is the event which started the approval process. This event contains the request itself (IdmRoleRequestDto). At the end of the process the event is called out again (performs the realization).



Approval by the helpdesk department

- The approving task will be assigned to all users with role **Helpdesk**.
- The role can be changed in the application configuration "**idm.sec.core.wf.approval.helpdesk.role**", the default setting is **Helpdesk**.
- The approval round can be enabled or disabled in the application configuration under key "**idm.sec.core.wf.approval.helpdesk.enabled**".

"Helpdesk" approval is disabled by default

.

Approval by the manager

* The approving task will be assigned to all users evaluated as the managers of the applicant. The manager is defined based on the industrial relations of the applicant.

- The approval round can be enabled or disabled in the application configuration under key "**idm.sec.core.wf.approval.manager.enabled**".

Approval by the manager is disabled by default

.

Approval by the user administration department

- The approving task will be assigned to all users with role **Usermanager**.
- The role can be changed in the application configuration

"**idm.sec.core.wf.approval.usermanager.role**", the default setting is **Usermanager**.
- The approval round can be enabled or disabled in the application configuration under key "**idm.sec.core.wf.approval.usermanager.enabled**".

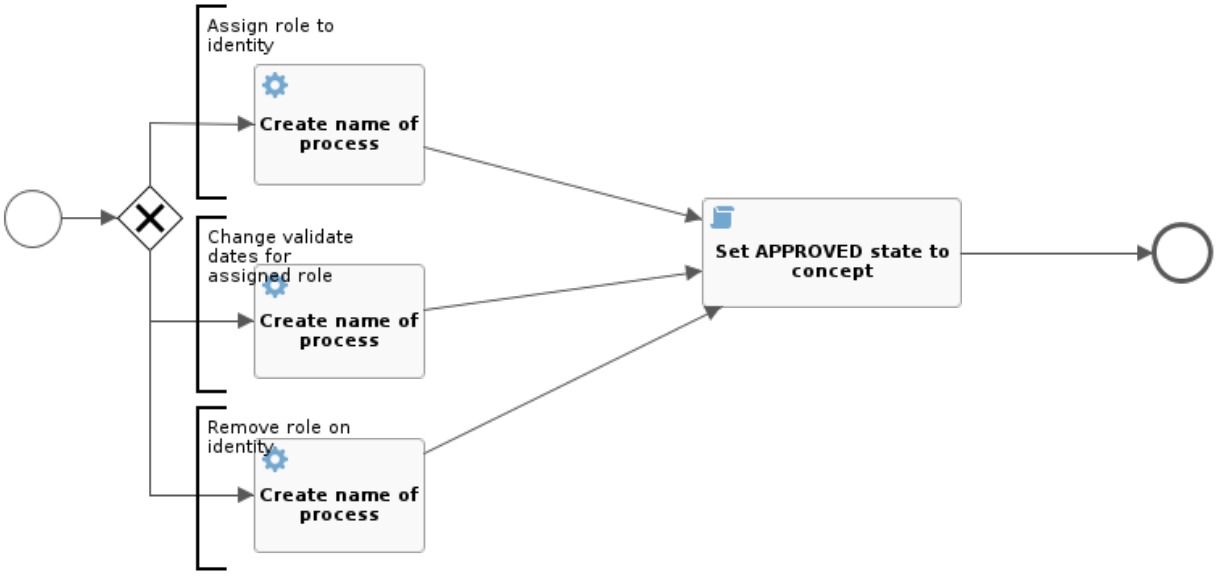
Approval by the user administration department is disabled by default.

.

Execution of approving subprocesses

Every constituent role being requested (change or assignment) can be approved separately. Due to that, a separate subprocess is always run for every role.

If the role is evaluated as not requiring approval, default process "**change-role-without-approve**" is run. This process only generates the name (it is saved in the history of workflow processes) and changes the state of the role concept to "**APPROVED**".



Setting of the approval process to a role

The approval process (for assigning or changing an ssigned role) is set to the role by setting the relevant priority. The role's priority can have values (0,1,2,3,4), where every priority can be assigned a different approval process.

The specific priority assignment and the type of approval process is defined in the application configuration:

```
idm.sec.core.wf.role.approval.{priority}={wf}
```

,where {priority} is the **priority number** and {wf} is the key of the process workflow.

There is only one process approving removal of assigned roles for the whole application (all priorities). Again, it can be defined in the application configuration under key "**idm.sec.core.wf.role.approval.remove**". The default process is "**approve-remove-role-by-manager**"

To make removing of a role approved, it is necessary that the role item "**Approve role removal**" is ticked.

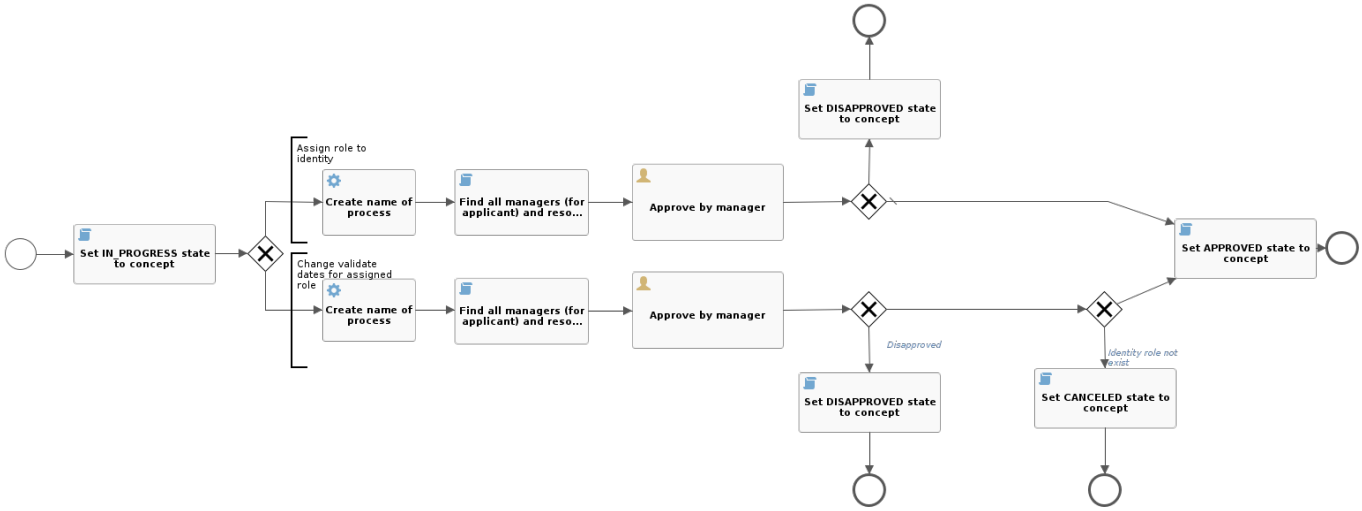
The default setting of priorities and approval processes is as follows:

- **No priority (0)**:

No approval takes place. The process "**change-role-without-approve**" is run (see above). This process is used, even if no workflow is configured (~is empty) for a priority.

- **Trivial priority (1)**:

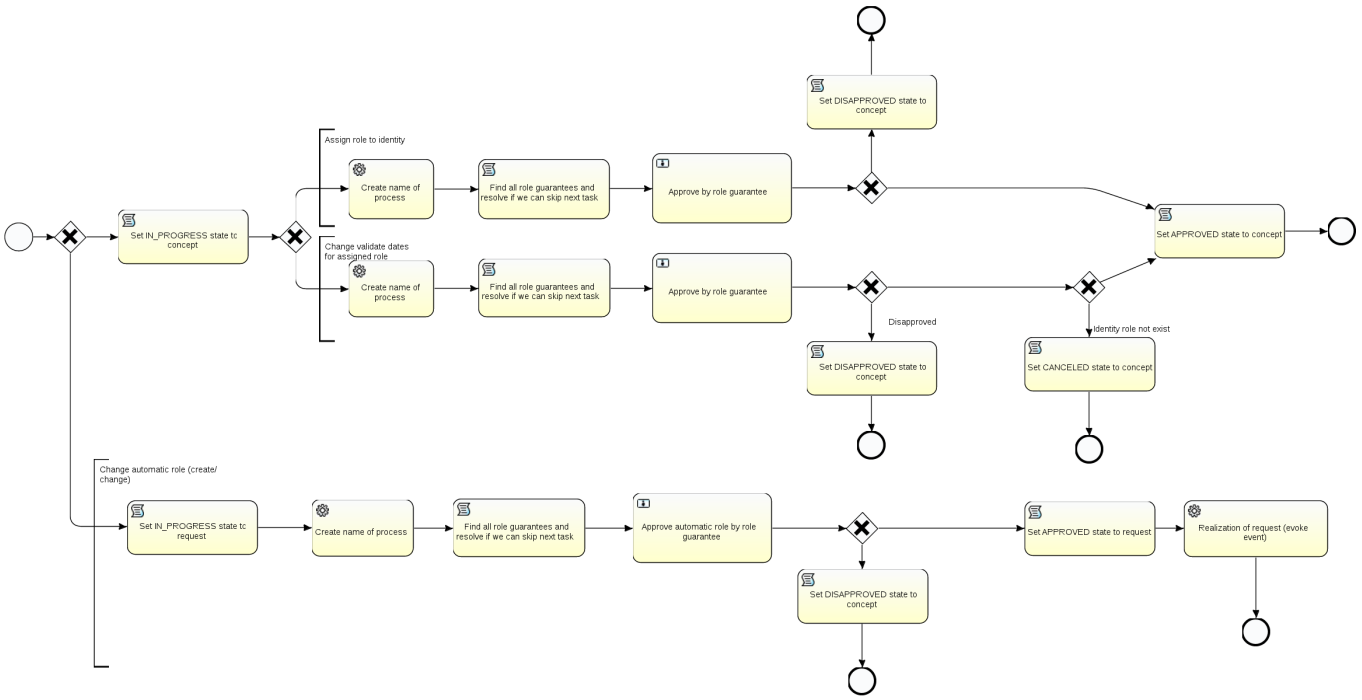
Approval by the manager of the applicant "**approve-role-by-manager**".



- **Low priority (2)**:

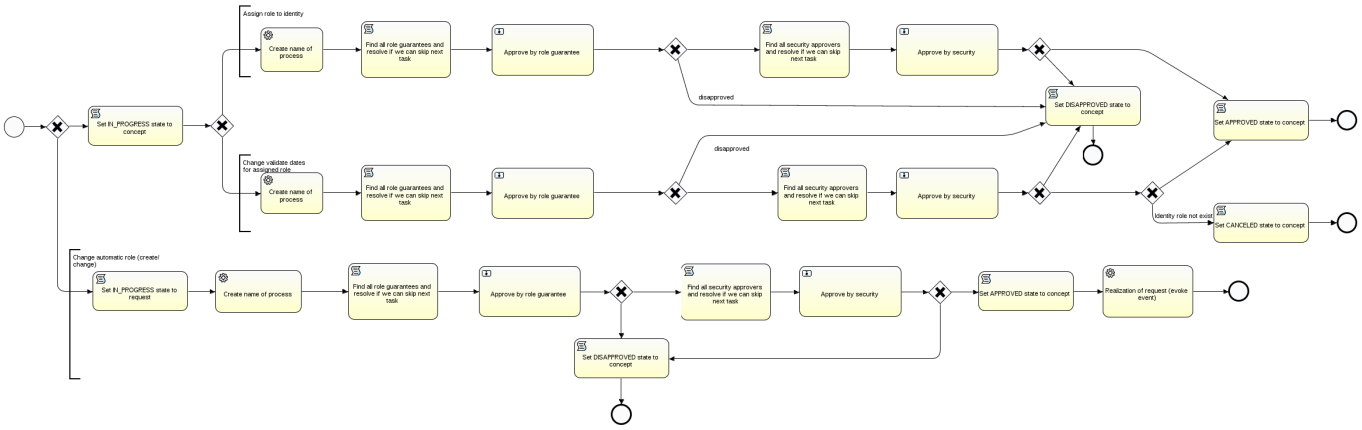
Approval by the guarantee of the role "**approve-role-by-guarantee**".

This process supports approving for change automatic role.



* **High priority (3)**: Approval by the role guarantee and by the security department "**approve-role-by-guarantee-security**".

This process supports approving for change automatic role.



Approval by the security department

- The approving task will be assigned to all users with role **Security**.
- The role can be changed in the application configuration "**idm.sec.core.wf.approval.security.role**", the default setting is **Security**.
- The approval round can be enabled or disabled in the application configuration under key "**idm.sec.core.wf.approval.security.enabled**".


Approval by the security department is disabled by default

.

Approval of role incompatibilities

- First is checking, if exists in the current request some incompatibilities of roles. Incompatibilities are searche only for new added roles (not disapproved) in the request. If none incompatibilites are found, then is this user task skiped.
- The approving task will be assigned to all users with role **Incompatibility**.
- The role can be changed in the application configuration "**idm.sec.core.wf.approval.incompatibility.role**", the default setting is **Incompatibility**.
- The approval round can be enabled or disabled in the application configuration under key "**idm.sec.core.wf.approval.incompatibility.enabled**".


Approval of role incompatibilities is enabled by default

.

Automatic skipping of approval processes

Automatic skipping of approval processes is performed if the request **realizator** (the one who really submitted it) is the same user as the **currently logged-in** user and, at the same time, is among the **candidates** who can approve this task. In that case, the task is skipped with the same result as if it was approved manually.

Approve, disprove task by task detail

To task detail is possible to get from url in notification (email and etd.), after click to url in notofication you will be redirected to task detail. After aprove, disaprove or go back you will be redirected to all your tasks. When is reopend already denied or approved task, user recive information about nonexisting or solved task.

Beware if url with task is copy by ctrl+c and paste directly to web browesr by ctrl+v, user will be redirected to the page on which he was (for example google.com, blank page, etc.)

Sending notifications

Sending notifications is by default enabled just to notify user just about his change of permissions. Sending notifications can be modified with two global boolean variables, which specify sending notifications.

- '**idm.sec.core.wf.notification.applicant.enabled**' by default is false and it configure sending

notifications to user, whose permissions will be modified. But if it is user, who requested change role to his own permissions, he is implementer too. So if this variable is false and the other is true, user is still implementer and will receive notification. But if both are true, user will get just 1 notification.
- '**idm.sec.core.wf.notification.implementer.enabled**' by default is true and it configure sending notification to user, who made request on behalf another user. For example, when manager request role for employee in his department and wants to know the result.

Security

Authorization policies are implemented fo role requests agenda. Concept role requests are secured by assigned role request internally. Only identity with authority `ROLEREQUEST_ADMIN` can read all concepts without filter by role request through the `IdmConceptRoleRequestController`.