

# Introduction to Pääsuke and configuring roles

Version 0.4 (11 January 2024)

In this document, we describe the aspects that are common for both sides:

- 1) pulling mandates from external systems and displaying them in the central system and
- 2) storing mandates centrally and offering API to other systems for pulling them

The main topic is configuring roles.

## 1. Versions

Version	Date	Description and changes
0.1		Document split from "Description of X-Road services offered to Pääsuke"
0.2	11.12.2023	delegateCanEqualToRepresentee renamed into delegateMustEqualToRepresenteeOnAdd
0.3	22.12.2023	canSubDelegate (boolean) replaced with subDelegable (enum)
0.4	11.01.2024	Roles in the subDelegableBy list can now also withdraw sub-delegated mandates.  added chapter "2.3 Understanding differences between configuration parameters addableBy, withdrawableBy, waivableBy, subDelegableBy"

## 2. What is a role

The role is a group of privileges to be used in a self-service that can be granted to delegate by the representee. Role always belongs to a namespace.

Roles usually have descriptive code that indicates the profession that needs that role ("Andmeesitaja") or what the owner can do (like "TOLLIDEKL\_ESITAMINE").

Roles in Pääsuke always start with the namespace code followed by a colon (for example: "STAT\_ESTAT:Andmeesitaja", "EMTA:TOLLIDEKL\_ESITAMINE").

Let's say some institution, for example, AgencyABC, wants to declare a role "Job ad editor". This institution must first request a namespace from Pääsuke and usually, namespace reflects the name of the institution - that is "AGENCY\_ABC" in our case.

Since the role code in Pääsuke must begin with namespace, the new role will be "AGENCY\_ABC:Job.Ad.Editor")

Is solely the responsibility of that agency to:

- Decide what can be done by a person who has a mandate for that role
- Declare the code, title, and description of that role

Properties of a role:

- the role belongs to a namespace
  - the namespaces are assigned to agencies by Pääsuke. So an agency must agree with RIA before it can start to use a namespace.
- the role has a code (unique identifier, not shown out to end user) that is unique in that namespace
  - in Pääsuke the role codes always are prefixed with the namespace so all the roles are unique in Pääsuke
- the role must have a title in Estonian
  - it is recommended to always provide the translation of the role title in English and in Russian
- the role can have a description
  - if there is a description it must have an Estonian translation and may have translations in English, and Russian

### 2.1 Role code

When some object has a property "role" it refers to role code.

Role codes can contain any UTF-8 symbols (including colons and spaces although spaces are not recommended)

NB! The roles are everywhere prefixed with their namespace that is separated by a colon. When separating the namespace from a role it must be kept in mind that the role code can contain several colons.

The role code must be unique using a case-insensitive comparison.

## 2.2 RoleDefinition parameters

Note: all boolean parameters default to false.

Property	Mandatory	Type	Precondition(s)	Description
code	yes	string		Unique identifier of the role. The prefix until the first colon is called the role namespace. Max length 4000 characters.
title	yes	Translation (see chapter 3.7)		Role title in different languages.
delegateType	yes	enum [ LEGAL_PERSON, NATURAL_PERSON ]		Type of persons this role can be assigned to. Setting delegateType only to LEGAL_PERSON is meant to be used for machine-to-machine roles. For example EMTA has a role "Käibedeklaratsioon (KMD) andmete saatmine masin-masin liidese vahendusel (code XT_MM_KMD)".  See Chapter 2.2.3 for more info.
representeeIdentifierIn	no	list of strings (up to 10 identifiers are allowed)		New mandates with this role code can only be created for representees in this list. This parameter only has an effect if it is not null and not empty.
representeeType	yes	enum [NATURAL_PERSON, LEGAL_PERSON]		Type of representees who can add a mandate with this role
GOVERNMENT_PERSON	no	list of strings		New mandates with this role code can only be created for representees in this list. This parameter only has an effect if it is not null and not empty
addableBy	no	list of strings		In order to add a mandate with this role, the user representing the representee must have a valid mandate with a role in this list. If the value is empty or null, this role cannot be assigned from Pääsuke.  <b>Note about natural persons</b>  If the role is configured so that representeeType has the value NATURAL_PERSON in it, then add "NAT_REPRIGHT:SOLEREP" to this list - this value indicates that a natural person (as a representee) is allowed to add a mandate with this role if the natural person adding the mandate has the right to represent oneself (kui volituse andja teovõime ei ole piiratud).  <b>Assigning a role through Admin Portal (MISP2)</b>  Pääsuke allows adding roles through its MISP2 based admin portal (this is a different portal than <a href="#">eesti.ee</a> ). If a role must be addable from the admin portal then addableBy list must include the value "ADMIN_PORTAL:ADMIN_USER".  Before 0.9.0 this was named assignableBy  If the role is addable also depends on parameter 'addableOnlyIfRepresenteeHasRoleIn' (if it is specified)
addableOnlyIfRepresenteeHasRoleIn	no	list of strings	addableBy has at least one item	Defining this list is used in rare cases where in order to assign the role, the <u>representee</u> also must have at least one mandate with a role in this list. For example to add some PRIA role the user representing the representee must have a role in the addableBy list and the <u>representee</u> must have the role "PRIA:PRIA.customer"  Before 0.9.0 this was named assignableOnlyIfRepresenteeHasRoleIn
addingMustBeSigned	no	boolean	addableBy has at least one item	If this is set as true then the user adding a mandate with this role needs to digitally sign it.
delegateMustEqualToRepresenteeOnAdd	no	boolean	addableBy contains at least one role that starts with "PAASUKE_ADMIN".	If this is set to true then only the administrator can create the initial mandate and the legal person can sub-delegate it further (to its employees). This parameter is for use-cases where not anyone has the right to use the service but administrator adds the initial mandate that can be further sub-delegated. For such mandates the subDelegable must be set to YES.
description	no	Translation (see chapter 3.7)		Role description in different languages. If the description is provided it must have at least the description translation in Estonian.
hidden	no	boolean		Mandates with hidden roles are not shown in Pääsuke UI. A hidden role is a method to add extra information about the person. For example, we could create a role AA with property addableBy=BB, hidden=false. Now we can create hidden role BB and we can add a mandate with the role BB to persons who are allowed to add role AA. False by default. If hidden is set to true then all of the role properties are ignored.  See Chapter 2.2.3 for more info about hidden roles. If hidden is set to true then all of the following properties are ignored.
validityPeriodFromNotInFuture	no	boolean		If true then Pääsuke UI restricts setting validityPeriod.from into the future when adding a new mandate. If adding a sub-delegate is allowed then this check is also applied when adding a sub-delegate.
validityPeriodThroughMustBeUndefined	no	boolean		If true then Pääsuke UI forces setting validityPeriod.through to infinity. If adding a sub-delegate is allowed then this check is also applied when adding a sub-delegate.

subDelegable	yes	enum [YES, NO, ASK, LEGAL_PERSON_YES__NATURAL_PERSON_ASK, LEGAL_PERSON_YES__NATURAL_PERSON_NO]		<p>System checks this value during adding a mandate.</p> <p>"NO" - a mandate with this role can not be added with the right to further sub-delegate it.</p> <p>"YES" - the system forces the user to add new mandates with this role with the right to further sub-delegate them.</p> <p>"ASK" - the user can choose during adding if further sub-delegating is allowed or not.</p> <p>"LEGAL_PERSON_YES__NATURAL_PERSON_ASK" - If delegate is a legal person then system forces the user to allow further sub-delegating the mandate. If delegate is a natural person then the user can choose if further sub-delegating the mandate is allowed.</p> <p>"LEGAL_PERSON_YES__NATURAL_PERSON_NO" - If delegate is a legal person then system forces the user to allow further sub-delegating the mandate. If delegate is a natural person then the user cannot add the mandate with the right to further sub-delegate it.</p>
subDelegableType	no	enum [ LEGAL_PERSON, NATURAL_PERSON ]	value of subDelegable is not "NO"	Type of persons this role can be sub-delegated to. This should have a value if canSubDelegable is set to true and normally (except in rare corner cases) the value will be NATURAL_PERSON.
subDelegableBy	no	list of strings	value of subDelegable is not "NO"	In order to add a sub-delegate for a mandate with this role or to withdraw a sub-delegated mandate the user representing the <u>delegate</u> of this mandate must have a valid mandate with a role in this list. This should have a value if subDelegable != 'NO'.
subDelegatingMustBeSigned	no	boolean	value of subDelegable is not "NO"	If this is set as true then the user is adding a sub-delegate for a mandate with this role then the user needs to digitally sign it.
waivableBy	no	list of strings		<p>Waiving means deleting a mandate from the delegate side (volitusest loobumine).</p> <p>The user representing the <u>delegate</u> must have a valid mandate with a role in the list to waive this mandate from the delegate side. If this is set to null or to an empty list then this mandate cannot be waived.</p> <p><b>Some examples</b></p> <p>Adding "BR_REPRIGHT:SOLEREP" to this list indicates that if a legal person is a delegate (or sub-delegate) then any person who has a sole representation right for this delegate is allowed to waive a mandate with this role.</p> <p>Adding "NAT_REPRIGHT:SOLEREP" to this list indicates that if a natural person is a delegate (or sub-delegate) then that person is allowed to waive a mandate with this role.</p>
waivingMustBeSigned	no	boolean	waivableBy has at least one value	If this is set true then the delegate has to digitally sign when the delegate (volituse saaja) wants to waive the mandate (volitusest loobumine).
withdrawableBy	no	list of strings		<p>Withdrawing means deleting a mandate from the representee side (volituse tagasivõtmine).</p> <p>The user representing the <u>representee</u> must have a mandate with a role in the list to withdraw this mandate from the representee side.</p>
withdrawalMustBeSigned	no	boolean	withdrawableBy or addableBy has a value	If this is set true then the representee (or the person representing the representee) has to digitally sign when he wants to withdraw the mandate (volituse tagasivõtmine).

## 2.3 Understanding differences between configuration parameters addableBy, withdrawableBy, waivableBy, subDelegableBy

### 2.3.1. Parameters that check representee side

The parameters "addableBy" and "withdrawableBy" check that the person performing the action has mandate under representee.

Let's look at the following example. We have the following person that is logged into eesti.ee/volitused

SESSION	Authenticated person	EE60001019906
	Represented person	EE10391131 (Väikefirma OÜ)

This person is the board member (BR\_REPRIGHT:JUHL\_SOLEREP) in this company so in the system there exists following right of representation.

MANDATE #1	representee	delegate	role
	EE10391131	EE60001019906	BR_REPRIGHT:JUHL_SOLEREP

In the system the following role has been described:

ROLE	code	NS:SUPERACCOUNTANT
	addableBy	BR_REPRIGHT:JUHL_SOLEREP, BR_REPRIGHT:PROK_SOLEREP

The person logged in can add this role since we can form a chain Authenticated person has valid mandate under representee with a role in the addableBy list.

## 2.3.2. Parameters that check the delegate side

The parameters "subDelegableBy" and "waivableBy" check that the person performing the action has mandate under delegate.

Let's look at the following example. We have the following person that is logged into [eesti.ee/volitused](https://eesti.ee/volitused)

SESSION	Authenticated person	EE60001019906
	Represented person	EE23456789 (Raamatupidajad OÜ)

This person is the board member (BR\_REPRIGHT:JUHL\_SOLEREP) in this company so in the system there exists following right of representation.

MANDATE #2	representee	delegate	role
	EE23456789	EE60001019906	BR_REPRIGHT:JUHL_SOLEREP

Väikefirma OÜ (EE10391131) has given the following mandate to the bookkeeping company:

MANDATE #3	representee	delegate	role
	EE10391131	EE23456789	NS:ACCOUNTANT

In the system the role "NS:ACCOUNTANT" has been described:

ROLE	code	NS:ACCOUNTANT
	waivableBy	BR_REPRIGHT:JUHL_SOLEREP, BR_REPRIGHT:PROK_SOLEREP

The person logged in would be able to waive mandate #3 (on the behalf of Raamatupidamisfirma OÜ) because role "NS:ACCOUNTANT" has role "BR\_REPRIGHT:JUHL\_SOLEREP" in the waivableBy list and the person logged in has this role (that means the user is the delegate in mandate #2) under the delegate of the mandate #3.

## 2.3.3. Withdrawing a mandate

Let's continue looking at the following example

Väikefirma OÜ (EE10391131) has given the following mandate to the bookkeeping company (EE23456789):

MANDATE #3	representee	delegate	role	canSubDelegate	subDelegatorIdentifier
	EE10391131	EE23456789	NS:ACCOUNTANT	true	null

And bookkeeping company has further sub-delegated to its employee EE49414160303:

MANDATE #4	representee	delegate	role	canSubDelegate	subDelegatorIdentifier
	EE23456789	EE49414160303	NS:ACCOUNTANT	false	EE23456789

Now person representing the Bookkeeping company can withdraw this sub-delegated mandate if that person has a role in subDelegableBy list under bookkeeping company (that is the original delegate or subDelegator) because subDelegableBy checks the delegate side of the mandate.

Person representing Väikefirma can withdraw this sub-delegated mandate if that person has a role in the withdrawableBy list under Väikefirma (because withdrawableBy checks the representee side)

# 3. Configuring who can perform some action

## 3.1 Natural persons representing themselves

If some action can be performed by a natural person representing oneself then this is expressed using "NATURAL\_PERSONS:SELFREP".

For example is natural person is allowed to waive a mandate that has been given to him, then the configuration must have:

```
waivableBy: [  
  "NATURAL_PERSONS:SELFREP"  
]
```