Description of X-Road services offered to Pääsuke

Pääsuke API version 0.9.5.

1. Introduction

This document is accompanied by OpenAPI definitions:

https://app.swaggerhub.com/apis/aasaru/x-road-services-consumed-by-paasuke/<version number>

and these definitions refer to a common domain definition:
https://app.swaggerhub.com/domains/aasaru/paasuke-common-types/<version number excluding the minor version>

1.1 Versions

Version	Date	Description and changes
0.2.0		First public draft with the services that are offered by parties who keep mandates on their side and want to publish that info to Pääsuke
0.2.1		Added chapter "3. X-road services offered by Pääsukese to query mandates that are stored in Pääsuke"
0.2.2		Added chapters "4. X-road services to modify mandates that are stored in Pääsuke" and "5. Integration with Pääsuke without implementing any X-road services"
0.3.1		Improved terminology and introduced different types of namespaces (parent, child, standalone and external). Header parameters changed back to query parameters (except "If-Modified-Since") Added chapter 3. "X-road services offered by Pääsukese to query mandates that are stored in Pääsuke" together with new endpont getNamespaces Person type INDIVIDUAL changed to NATURAL_PERSON. Smaller adjustments and parameters.
0.4.0		Shortened "namespace" to "ns" everywhere. Added ns path parameter to all methods that change state. New endpoints addMandateToDelegate, addMandateSubdelegate, removeAllMandatesFromDelegate Added chapter 1.2. Preface to this document.
0.4.1		Changed the Translations object ("2letterLangCode":"translation") changed parent_namespace parentNamespace Person type LEGAL_ENTITY changed to LEGAL_PERSON (so it matches better with NATURAL_PERSON)
0.5.0	19.01.2023	Person IdentityCode changed into person identifier. Added description of responses to queries. Added description of custom data types used inside the API Added description of how sub-delegating a mandate takes place.
0.5.1	20.01.2023	validityPeriod, editMandate, Authorization
0.5.2	24.01.2023	Added mandatevalidityPeriodLimit

0.6.0	31.01.2023	Changed Person.identifier - now using URI-s instead of "internal:", "email:" prefixes. RoleMetaData moved into RoleDefinition and removed state field from it. Added roleDefiniton fields Role.deletableBy, Role.representeeType (this has an additional enum value GOVERNMENT_PERSON). The output of "/roles" query changed so that it returns an array of the following: namespace: "{namespaceCode}", roles: <array of all the roles in namespace> Added description Problem data type that is returned in case of any errors. Removed support for editing a mandate through xRoadPutEdit link. Removed validityPeriodLimit.
0.6.1	08.02.2023	Role codes now always start with namespace + ":" Role codes can contain any UTF-8 characters (including spaces). Added role parameter "deletableByDelegate". Added roles that are checked when Pääsuke is used through MISP2 portal.
0.7.0	14.02.2023	Role codes now always start with the namespace code followed by a colon. Changes to links: • xRoadDeleteMandate delete • xRoadPutEdit update • xRoadPostSubDelegate addSubDelegate • uiExternalView origin • uiExternalEdit - removed The internal names of the methods were changed: • "removeMandate" was renamed to deleteMandate. • "addMandateSubDelegate" "addSubDelegate" Added new Person type UNKNOWN. Removed pagination (attributes "limit" and "skip" for GET queries). Changed URL path "/ns/" (namespace) to "/nss/" (namespaces) Added role definition attribute "deletableByDelegate".
0.7.1	16.02.2023	a separate x-road request is made for adding each mandate (rather than grouping several mandates into a single request)
0.7.2	17.02.2023	Removed "update" link that was meant for updating the record. It might be added back in future versions of API. Added Person.type OTHER as a value. Added RoleDefinition.visible
0.7.3	21.02.2023	Common types (including request and response types) moved to a different domain definition. Improved the documentation within the OpenAPI definitions.
0.7.4	8.03.2023	To RoleDefinition added fields canAssignIfHasRoleAndOneOf, canDeleteIfHasRoleAndOneOf. Added the global restriction that the sub-delegate can only be a natural person and never a legal person. The mandate can include a field "subDelegatorIdentifier" to indicate the identifier of the person who added this relation via sub-delegation.
0.7.5	6.04.2023	Set 100 to be the maximum number of Mandate objects in a single MandateTriplet (chapter 3.4).
0.7.6	19.04.2023	Add new filter parameter "delegate" to getRepresenteeDelegatesWithMandates. To better facilitate eesti.ee RR partner use-case the role definition parameters canAssignIfHasRoleAndOneOf and canDeleteIfHasRoleAndOneOf were removed and replaced with new parameters assignableOnlyIfRepresenteeHasRoleIn and delegateCanEqualToRepresentee (see chapter 3.3.2)
0.8.0	8.05.2023	Added new parameter document to payloads of addMandate and addSubDelegate. This contains information about the digitally signed asice container.
0.8.1	11.05.2023	RoleDefinition.visible changed from optional to compulsory.

0.9.0	23.05.2023	deleteMandate method was renamed to editMandat and changed from DELETE to PUT (that takes a request body)
		
		In Chapter 1.3. split deleting a mandate into two separate terms:
		
		1. mandate withdrawal
		2. mandate waiving.
		
		In Chapter 3.3.2 changed RoleDefinition:
		
		• added
		  ◦ addingMustBeSigned
		  ◦ withdrawalMustBeSigned
		  ◦ waivingMustBeSigned
		  ◦ subDelegableBy (list of strings)
		• added new fields and deprecated some old ones:
		  ◦ assignableBy (now deprecated) replaced with addableBy
		  ◦ assignableOnlyIfRepresenteeHasRoleIn (now deprecated) replaced with addableOnlyIfRepresenteeHasRoleIn
		  ◦ deletableBy (now deprecated) now renamed to withdrawableBy
		• changed
		  ◦ visible:false (now deprecated) changed to hidden:true
		  deletableByDelegate(boolean, now deprecated) changed to waivableBy (list of strings)
		• following RoleDefinition attributes are now compulsory
		  ◦ representeeType
		  ◦ delegateType

0.9.1	26.05.2023	Removed "/nss/{ns}" from the beginning of add a sub-delegate link and from the beginning of the edit mandate link
0.9.2	13.06.2023	• Added RoleDefinition.validityPeriodFromNotInFuture, RoleDefinition.validityPeriodThroughMustBeUndefined. • When mandates are queried and there are no mandates to return then an empty list should be returned (instead of returning HTTP status code 404).
0.9.3	19.06.2023	RequestBody of PUT changed from EditMandate to DeleteMandate to ease implementation. waivableBy, withdrawableBy, subDelegableBy - these no longer default to addableBy when unset but the values must be defined for each parameter. Removed deprecated RoleDefinition parameters (assignableBy, assignableOnlyIfRepresenteeHasRoleIn, deletableBy, visible, deletableByDelegate.
0.9.4	4.09.2023	Added subDelegatingMustBeSigned to RoleDefinition. Support for following RoleDefinition attributes lifted from 1.1. to 2.0: canSubDelegate, subDelegateType, subDelegableBy, validityPeriodFromNotInFuture and validityPeriodThroughMustBeUndefined. Removed RoleDefinition attribute modified.
0.9.5	22.09.2023	Action "DELETE" under editMandate which is used to end the validity of the mandate has been split into two parts: • action="DELETE_WITHDRAW" (if the mandate is being withdrawn by the representee) • action="DELETE_WAIVE" (if the mandate is being waived by the delegate) See Chapter 2.5 for more info.

1.2 Preface

1.2.1 Use cases of Pääsuke that this document is covering

Pääsuke displays the mandates that are stored in external e-services (at Tax and Customs Board for example). If the external e-service supports it, Pääsuke offers the following additional operations besides displaying mandates:

- deleting a mandate
- adding a new mandate
- adding a sub-delegate for a mandate

In order to resolve these use cases Pääsuke has created a standard for the queries that e-services must offer to Pääsuke over x-road.

This document is the description of that standard.

Once the external e-service has implemented some or all of the queries in the standard then Pääsuke can be configured to call these queries.

1.2.2 Other use cases that are also resolved by Pääsuke but not covered by this document

1.2.2.1 External system can use Pääsuke as a mandate system instead of implementing its own logic

e-services can use Pääsuke to store the mandates centrally in eesti.ee and use the user interface of eesti.ee to manage them (give out new or delete old ones).

Pääsuke offers x-road services to these e-services to query mandates from Pääsuke when a person tries to authorize himself in that service. Pääsuke also offers a list of persons (like management board members, procurers, etc) defined in the Business Registry as a response to this query.

1.2.2.2 External system can use Tara GovSSO as authorization provider

Currently, for most Estonian government e-services the authentication service is provided by Tara. For any e-service using Tara, there is no need to implement x-road services as Tara provides signed proof to e-services about the authenticated person.

GovSSO is Tara with SSO and it provides single sign-on functionality on top of Tara. It would be technically possible for GovSSO to offer additional UI flows for authenticated users to select a person to be represented in the upcoming session. This way GovSSO would provide the selected person as part of the OpenID connect flow together with details of the authenticated person.

If the user later wants to switch represented person to a different person then that would be possible as the e-service would anyway have to keep the session alive with GovSSO. To switch are representee the e-service would have to send the user's browser back to GovSSO for that and the user would return with details of the selected representee that would be signed by GovSSO.

This integration pattern is currently seeking interested parties. Please connect with Pääsuke team if you would be interested in using that flow.

1.2.3 Principles to follow when implementing the queries

1.2.3.1 Do not send null values

Not recommended	Recommended

`"title": {`
`  "et": "Tere",`
`  "en": "Hello",`
`  "ru": null`
`}`
	`"title": {`
	`  "et": "Tere",`
	`  "en": "Hello"`
	`}`

1.2.3.2 Pääsuke is not working with historical data

When implementing the queries:

- only return mandates that are currently valid or will become into effect in the future.
- do not include mandates that have been active in the past and have been deleted

1.2.3.3 Each e-Service can decide its own level of support

There are the following options for any e-service:

The required part is to offer queries for Pääsuke to query and display the mandates. This means the e-service must implement two queries for pulling the mandates and one query for Pääsuke to pull the role definitions.

Additionally, any or all of the following can be supported by the e-service

1. Deleting mandates
2. Adding new mandates
3. Adding a sub-delegate for a mandate

1.3 Terminology

- **Pääsuke** - central access rights management system hosted in eesti.ee
- **RIA** - Information System Authority (Riigi Infosüsteemi Amet), agency that develops and runs eesti.ee and Pääsuke
- **institution** - some party who has a self-service system that either queries mandates from Pääsuke and/or has mandates declared in the system and publishes them in Pääsuke.
- **representee** - a person (private or legal) who has given a mandate to a delegate to be represented by that delegate (or its sub-delegates)
- **delegate** - a person (private or legal), a representee has given the mandate to represent itself. Delegate normally always has the right to represent oneself (except if the person doesn't possess active legal capacity - in Estonian "piiratud teovõime")
- **namespace** - a group of roles that are maintained by a single institution.
- **privilege** - individual right to perform some action in e-service
- **role** - a group of privileges to be used in an e-service that can be granted to delegate by the representee. Role always belongs to a namespace. Read more from the chapter What is a role
- **mandate** - a role that is given to a delegate by some representee. Mandates can have a start date and end date, and some mandates can be sub-delegated (in Estonian "*edasi delegeerima*").
- **mandate withdrawal** - if the representee (or a person representing the representee) deletes the mandate that has been given to some delegate (in Estonian *"volituse tagasivõtmine"*)
- **mandate waiving** - if the delegate (or a person representing that delegate) deletes a mandate that has been given to the delegate (in Estonian *"volitusest loobumine"*)

1.3.1 What is a role

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

More info about roles is in chapter 3.3.

1.4 Notes

1.4.1 Prototype

Pääsuke offers its prototype publicly - it is available here: https://paasuke.github.io/proto/. The screenshots used in this document are taken from the prototype.

The prototype is to illustrate how different mandates would be displayed to the user and what assigning a role looks like and how it looks like to withdraw /waive a mandate or add a sub-delegate to a mandate.

1.4.2 OpenAPI definitions

OpenAPI definitions (verify that the version in the link is up to date):

- https://app.swaggerhub.com/apis/aasaru/paasuhalduse-x-tee-teenused/<version>#/Offered%20to%20P%C3%A4%C3%A4suke
- https://app.swaggerhub.com/apis/aasaru/paasuhalduse-x-tee-teenused/<version>#/Offered%20to%20P%C3%A4%C3%A4suke%20(additional)

1.4.3 Mock service

In order to better illustrate what kind of responses are expected from external parties implementing these queries a mock service has been developed.

It is possible to run the mock locally using Docker or Java (see Running the mock service locally).

The mock is also accessible over ee-dev X-Road (see chapter 1.4.5).

The same mock plays different parties (EMTA and STAT for example):

- If you want the mock to act like EMTA mock then set the value of "X-Road-Id" header to something that starts with "EMTA"
- If you want the mock to act like a mock for STAT subsystem ESTAT then set the value of "X-Road-Id" header to something that starts with "STAT"
- In the future, the mock will also serve other agencies like "PRIA", "MAJ" etc

1.4.4 Running the mock service locally

Instructions can be found here: https://github.com/e-gov/PH/tree/main/ph-xroad-api-mock

1.4.5 Playing with the mock service over X-Road

A mock service has been set up in ee-dev X-Road that mimics the expected behavior of a system providing such services.

You can send requests against that service to better understand how the service has to work.

Throughout this document, the X-Road-specific headers (headers beginning with X-Road-...) are removed from the examples of HTTP requests.

If you want to test the queries then you always need to add the following x-road headers and the accept parameter:

```
curl \
-H "accept: application/json" \
-H "X-Road-Client: ee-dev/GOV/70001234/generic-consumer"\
-H "X-Road-UserId: EE39912310123" \
-H "X-Road-Id: EMTA_08544bbd2f41473800309d16bd81c64c0f54193d84b53f8ad22aacdf5e" \
-X GET "https://security-server/r1/ee-dev/GOV/70006317/volitused-mock/volitused-estat/v1/roles"
```

You need to make the following replacements:

- ee-dev/GOV/70001234/generic-consumer - replace with your own details. RIA needs to grant access to this x-road client.
- replace https://security-server with your security server IP/DNS.
- You need to set X-Road-UserId to your own personal id code.

1.4.6 Exporting the OpenApi definitions from SwaggerHUB

The definitions of the API are split between two files - the common data types are defined in another URL.

This is how you can download all the definitions as a single file (this is called 'Resolved').

1. Open definitions https://app.swaggerhub.com/apis/aasaru/x-road-services-consumed-by-paasuke/
2. Check that the correct version is open. If needed change the version from the drop-down menu
3. Do not copy-paste the definitions (as this would end up with an unresolved copy)
4. Choose Export Download API either pick "JSON Resolved" or "YAML Resolved".

2. Standard X-Road services that are consumed by Pääsuke

These services are used to show all the mandates from a central system. This way:

- any representee has visibility all over the Estonian e-services of the mandates that are currently valid.
- any delegate has information about all the mandates assigned to him by different representees.

Pääsuke uses the following services to query systems that among other things store mandates. These services are called standard services. Although data providers are different (Statistics Estonia, Estonian Tax and Customs Board, etc) these systems have all agreed to use the same query and data format.

This data that is pulled is displayed in Pääsuke UI (that is going to reside under eesti.ee)

2.1. Clarifications

2.1.1 Historical data is not returned

The services only return mandates that are currently valid or will become valid in the future. Records that are no longer valid are not available through this API.

2.1.2 X-road headers

If an actual person is making requests in Pääsuke then Pääsuke always adds headers:

- X-Road-UserId - identifier of the user currently logged in to Pääsuke (for example EE50001029996)
- X-Road-Represented-Party - identifier of the legal person currently being represented (for example EE11065244)

However, there might be requests from Pääsuke that are made by some automatic process. Then these two headers are missing.

One example of such a request could be a situation where:

1. The Employment Register (TÖR) has identified that person P left company C one day ago
2. The Employment Register triggers a request to Pääsuke asking if P has any valid mandates under C (no information about the user as it is a background process).
3. Pääsuke makes a request to other systems that store mandates on their side (and it doesn't add these headers) to find out if P has any valid mandates under C
4. If any matches are found then TÖR sends out an e-mail to management board members of C with a warning (we noticed that some person recently left your company but it seems the person still has valid mandates. Please go to Pääsuke and review the mandates of your company).

2.1.3 Who performs access rights check

Pääsuke is built to verify if the person is allowed to add, edit, or withdraw/waive any mandate according to role configuration.

The party that provides x-road services is welcome to add their own validations. This forms a two-layer authorization check.

2.2 Query "getRoles"

OpenAPI definition: https://app.swaggerhub.com/apis/aasaru/x-road-services-consumed-by-paasuke/<version>#/default/getRoles

CURL query

```
curl -H "If-Modified-Since: 2022-11-12T00:00:00+02:00" \
     -X GET \
     "https://security-server/r1/ee-dev/GOV/70006317/volitused-mock/volitused-estat/v1/roles"
```

The return type is the array of RoleDefinition (described in chapter 3.3.2).

Pääsuke uses this query to periodically fetch all translations of roles and also the metadata about roles. This metadata tells Pääsuke who is allowed to assign a particular role.

2.2.1 Why query "getRoles" is needed

The queries that return mandates, these queries use role codes in the mandate payload. For the Pääsuke UI to translate these codes into different languages it uses this getRoles query to fetch the translations.

2.2.2 Limiting the data that needs to be transferred with each request using 'If-Modified-Since' header

Pääsuke does not include the 'If-Modified-Since' header in the first request.

When the service returns the list of roles, Pääsuke goes over all the returned roles and memorizes the latest modification date (if at least one of the roles had a value for metadata.modification).

If the latest modification date has been stored by Pääsuke then on the next request it includes this value on the 'If-Modified-Since' header in the request. This is done to indicate the date and time of the latest role modification Pääsuke has already copied over.

The service can:

- Respond with HTTP Status code 304 if no roles have been changed since that time. The service provider is also allowed to ignore that property and never respond with HTTP Status code 304.
- Otherwise, all results (that match the filters) are returned (even the ones that have modified time earlier than the If-Modified-Since parameter).

2.2.3 Hidden roles

There are use cases where mandates are used to send additional info that is needed by Pääsuke but not displayed to the end user. For this Pääsuke allows the roles to be configured as hidden.

Let's look at the following example. PRIA only allows new mandates to be added from Pääsuke only for (legal and natural) persons who are registered as PRIA customers.

The definition of the role "PRIA:PRIA.customer" has the following parameters:

Field name	Field value
code	PRIA:pria.customer
title.et	PRIA klient
hidden	true

And a MandateTriplet with that role would be returned as:

CURL query

```
{
  "representee": {
    "type": "LEGAL_PERSON",
    "legalName": "Agro Agro AS",
    "identifier": "EE11430169"
  },
  "delegate": {
    "type": "LEGAL_PERSON",
    "legalName": "Agro Agro AS",
    "identifier": "EE11430169"
  },
  "mandates": [
    {
      "role": "PRIA:PRIA.customer"
    }
  ]
}
```

Note that the values of the delegate and the representee are the same!

2.3 Query "getRepresenteeDelegatesWithMandates"

CURL query

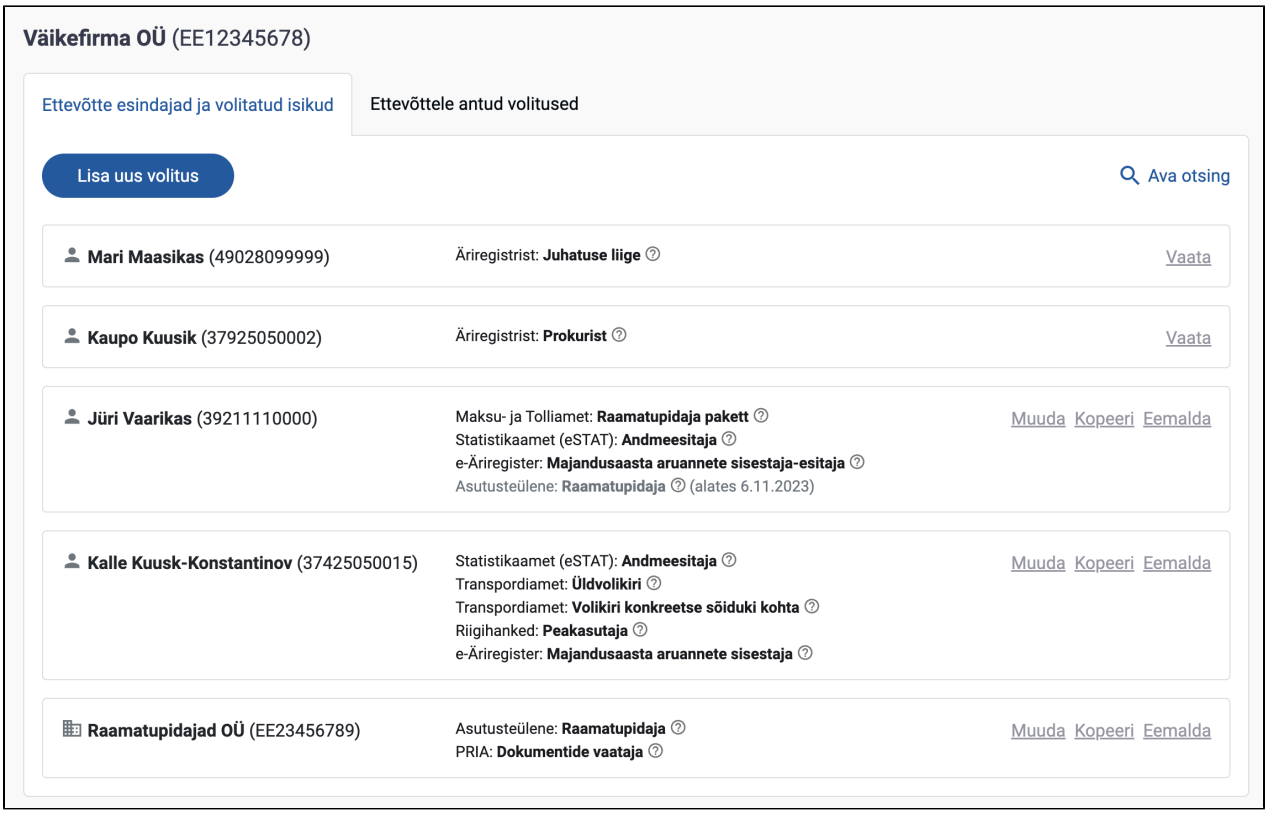
```
curl  -X GET \
     "https://security-server/r1/ee-dev/GOV/70006317/volitused-mock/volitused-estat/v1/representees/
{representee}/delegates/mandates"
```

Returns all delegates (with mandates), who have a right to represent the representee currently or are scheduled to become active in the future.

NB! The query should not return mandates that have been valid in the past.

2.3.1 View "Ettevõtte esindajad ja volitatud isikud"

This query is used to serve the following view in the Pääsuke UI. It displays all mandates that the representee has given out to others or that are assigned by law (Äriregistrist).



2.3.3 View "Minu esindajad"

In the future natural person can use Pääsuke to see what kind of natural persons he/she has given mandates to represent himself/herself.
To show the mandates the application also performs the query described at the beginning of this paragraph (2.3).

2.3.4 View "Ettevõtte <ettevõttenimi> poolt antud volituste edasivolitused"

There is one additional view that is served by the same query (and by setting query parameter "subDelegatedBy"), this use case is described in chapter 2.8.

It is only important if the external service is planning to allow adding sub-delegates.

2.3.5 Response structure of query getRepresenteeDelegatesWithMandates

This query and the next query (described in 2.4) have identical response structures, both return a list of MandateTriplets. The MandateTriplet type is described in chapter 3.4.

This query cannot return links of type "addSubDelegate" but it can return links of type "delete".

2.3.6 Rendering links for deleting a mandate

If at least one of the mandates has a link "delete" then the UI adds a button that initiates deleting the mandate.

2.4 Query getDelegateRepresenteesWithMandates

```
curl  -X GET \
     "https://security-server/r1/ee-dev/GOV/70006317/volitused-mock/volitused-estat/v1/delegates/{delegate}
/representees/mandates"
```

Returns all representees (with mandates) that the delegate has the right to represent.

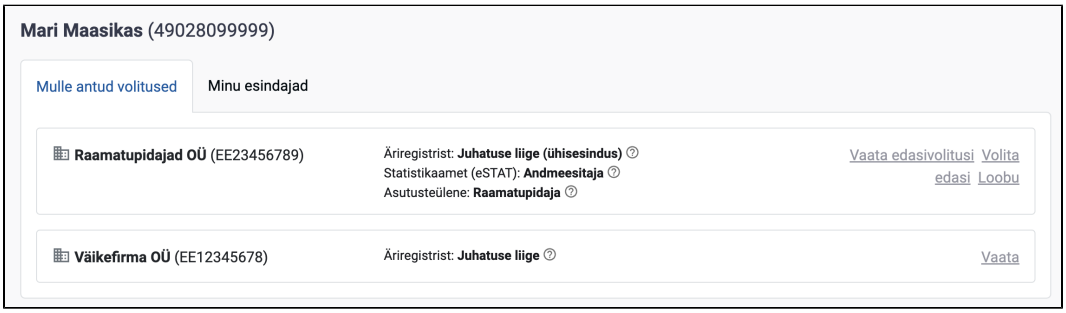
This query serves views that are described in chapters 2.4.1 and 2.4.2.

2.4.1 View "Ettevõttele antud volitused"

A legal entity (like an accountant bureau) is looking, at what kind of mandates other legal entities have given him.

2.4.2 View "Mulle antud volitusted"

A natural person opens Pääsuke to see what kind of mandates he has been given anywhere in the Estonian e-services (that are present in Pääsuke).



2.4.3 Response structure of query getDelegateRepresenteesWithMandates

This query and the previous query (described in 2.3) have identical response structures, both return a list of MandateTriplets. The MandateTriplet type is described in Chapter 3.4.

This query can return links of type "addSubDelegate" and "delete".

2.4.4 Differences between waiving and withdrawing a mandate

If a person has added a mandate to a delegate and then wishes to delete this mandate then this is called withdrawing (tagasivõtmine). If the delegate wants to delete this mandate that has been given to her then this is called waiving (loobumine)

2.4.5 Links for waiving a mandate ("Loobu")

If at least one of the mandates has a link "delete" then the UI adds a button that initiates giving up a mandate.
It is similar to deleting a mandate but the deleting is initiated from the delegate side (the receiver of the power of the attorney).

2.4.6 Links for adding a sub-delegate ("Volita edasi")

If at least one of the mandates in the list has a link "addSubDelegate" then the UI adds a button to initiate adding a sub-delegate ("Volita edasi").

2.5 Query editMandate

Currently only deleting is allowed.

2.5.1 Path parameters

```
curl -X PUT \
     "https://security-server/r1/ee-dev/GOV/70006317/volitused-mock/volitused-estat/v1/representees/
{representeeId}/delegates/{delegateId}/mandates/{mandateId}"
```

Values of representeeId, delegateId, and mandateId are taken by Pääsuke from the output of the query that produced the list (link with rel "delete"). The value of ns is taken from the role of the mandate (the first part of the role until the first colon (which is excluded)).

2.5.2 Payload

```
{
  "action": "DELETE_WITHDRAW",
  "authorizations": [
    {
      "userIdentifier": "EE39912310123",
      "hasRole": "BR_REPRIGHT:SOLEREP"
    }
  ],
  "document": {
    "uuid": "5b72e01c-fa7f-479c-b014-cc19efe5b732",
    "singleDelegate": true
  }
}
```

This request tells the provider of the service to end the validity of the mandate.

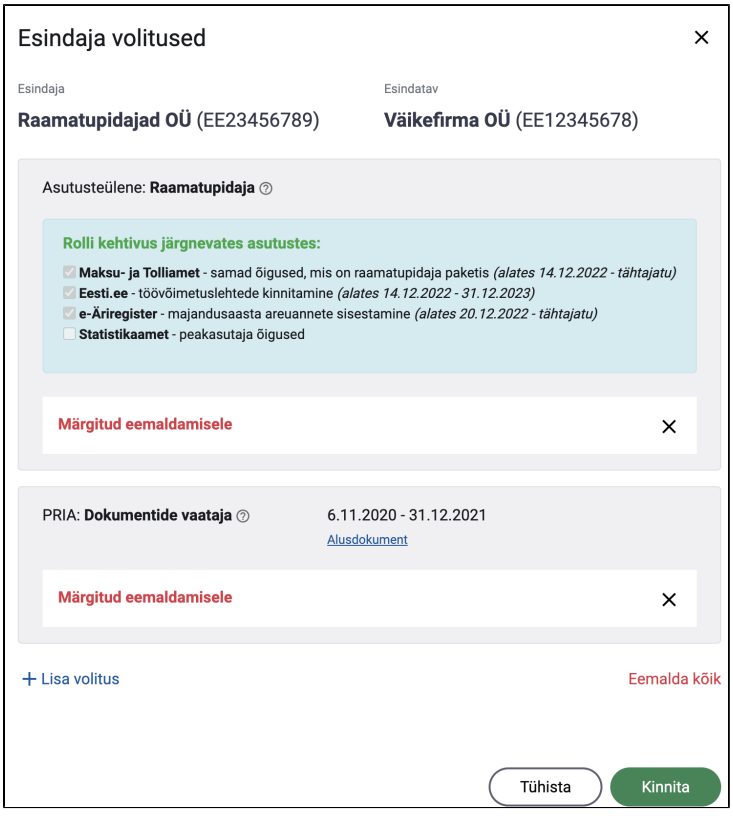
The action is a required attribute.

- If the person who gave the mandate (representee) is ending the validity of the mandate then the action is set to 'DELETE_WITHDRAW'.
- If the mandate is being ended by the person who was the delegate of the mandate then the action is set to 'DELETE_WAIVE'.
- Until version 0.9.4 the value was always set to 'DELETE'.

- The property "document" is present if the deleting of the mandate was digitally signed. Signing will be added in Pääsuke version 2.0

2.5.3 View to either withdraw mandates from the delegate or for the delegate to waive the mandates

Serves the following view of the Pääsuke UI. This view allows the user to individually pick the mandates to be removed.



For each mandate that was selected for deletion - Pääsuke performs this delete request.

2.5.4 Deleting a mandate that has been sub-delegated

NB! If the mandate to be deleted (by withdrawing it or waiving it) has been further sub-delegated then all the sub-delegated mandates need to be deleted as well. Pääsuke will not send separate requests to delete these sub-delegated mandates.

The following example illustrates this case.

Let's say Väikefirma OÜ adds a role A to Raamatupidajad OÜ and as a result of this a mandate A1 is created.

Raamatupidajad OÜ adds Raili Raamatukoi as a sub-delegate to this mandate A1. As a result, a new mandate A2 is created so that Raili Raamatukoi can now represent Väikefirma OÜ. This mandate A2 must be linked to mandate A1 in the database of the provider of the service. Pääsuke won't store this mandate or the link between A2 and A1 on Pääsuke side.

Raamatupidajad OÜ adds Ülle Pääsuke as a second sub-delegate to this mandate A1. Technically a new mandate A3 is created so that Ülle Pääsuke can now represent Väikefirma OÜ. This mandate A3 must be linked to mandate A1 in the database of the provider of the service.

Now when Väikefirma OÜ deletes the mandate A1 that has been given to Raamatupidajad OÜ then Pääsuke will send a delete request for that mandate A1. At the same time, the implementor of the service must also delete mandates A2 and A3, Pääsuke won't send separate delete queries for them. If deleting the mandates A2 or A3 fails on the implementor's side then deleting A1 must fail as well and the error message returned by the implementor will be displayed to the end user.

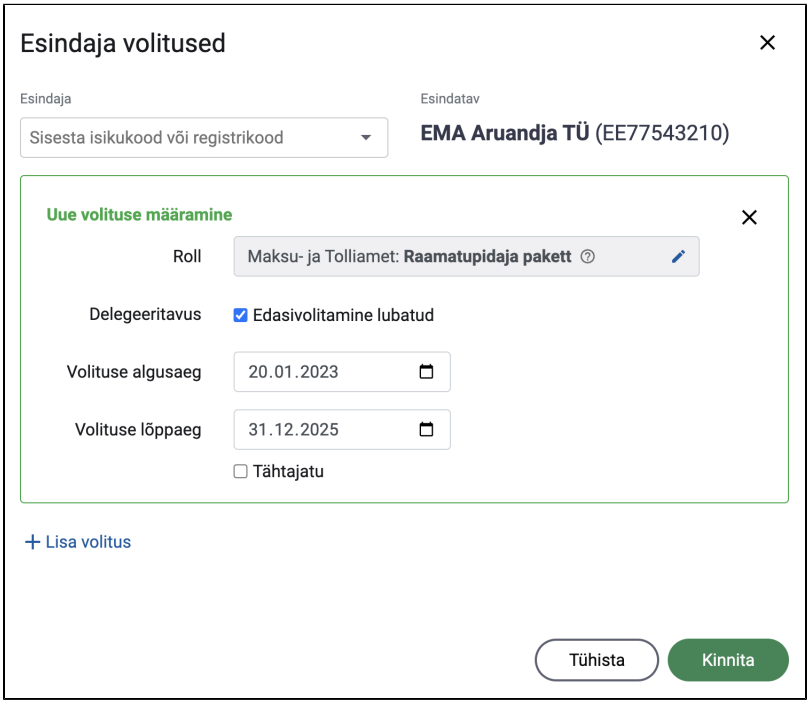
2.5.5 Digitally signing the request

- if at least one role is being withdrawn or waived (see 2.4.4 for differences) and this role has RoleDefintion.waivingMustBeSigned=true or RoleDefintion.withdrawalMustBeSigned=true (see chapter 3.3.2 for detailed info) then the user must first digitally sign this request and only after the user has signed then the process will continue and a separate PUT request is sent out for each mandate

2.6 Query addMandate

2.6.1 View to add mandates to a delegate

This is how a new mandate can be added:



The user can add several mandates from the same screen. After confirming (by pressing "Kinnita") the flow might continue with the signing step or the adding operations might be performed right away:

- if at least one role to be added has RoleDefintion.addingMustBeSigned (see chapter 3.3.2 for detailed info) then the user must first digitally sign this request and only after the user has signed then the process continues

Pääsuke makes a separate request with each of the mandates to be added:

2.6.2 Path parameters

Query to be sent out by Pääsuke

```
curl -X 'POST' \
  'https://security-server/r1/ee-dev/GOV/70006317/volitused-mock/volitused-estat/v1/representees/EE10391131
/delegates/EE38302250123/mandates' \
  -H 'Content-Type: application/json' \
  -d ' <Payload is described in next paragraph> '
```

2.6.3 Payload

Query to be sent out by Pääsuke

```
{
  "representee": {
    Person to be represented
  },
  "delegate": {
    Person getting the representation rights
  },
  "mandate": {
      "role": "GLOBAL1_EMTA:GLOBAL1_EMTA:ACCOUNTANT",
      "canSubDelegate": true,
      "validityPeriod": {
        "from": "2017-07-21",
        "through": "2024-02-21"
      }
  },
  "authorizations": [
    {
      "userIdentifier": "string",
      "hasRoles": "MANAGEMENT_BOARD_MEMBER"
    }
  ],
  "document": {
    "uuid": "5b72e01c-fa7f-479c-b014-cc19efe5b732",
    "singleDelegate": false
  }
}
```

Authorizations are used to show the information about who has confirmed that change and on what grounds (see chapter 3.8).

The presence of "document" indicates that the request was digitally signed (see chapter 3.11).

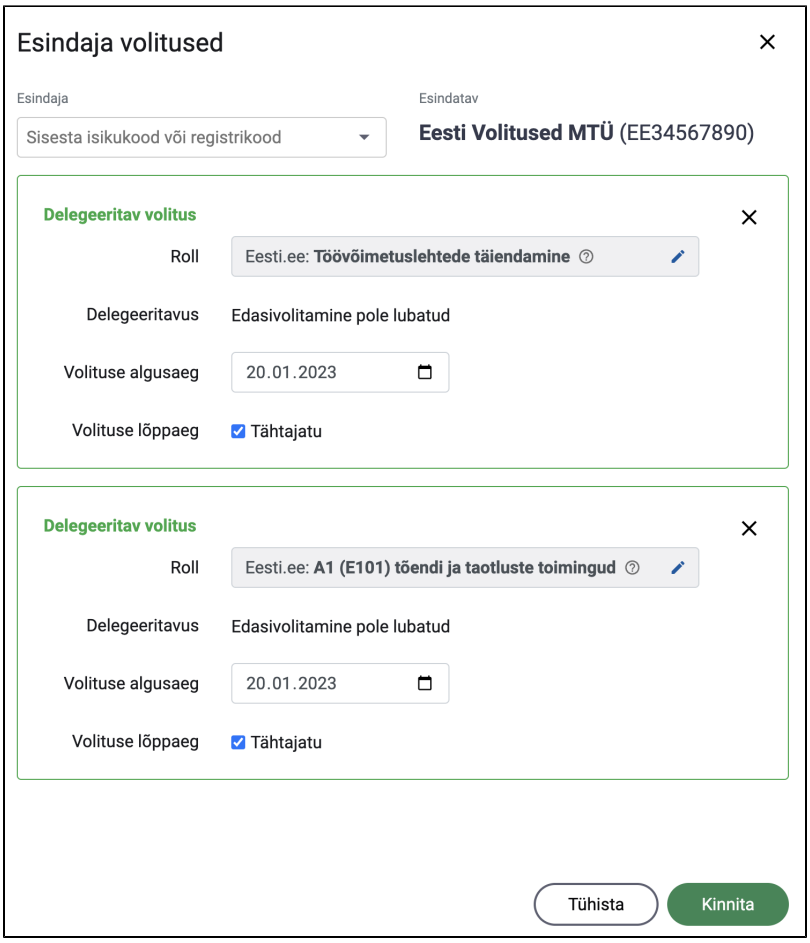
2.7 Query addSubDelegate

NB! The following is important to only those external parties who plan to offer the functionality of adding sub-delegates.

2.7.1 View where adding a sub-delegate can be started

This flow is started in chapter 2.4 (by pressing the button "Volita edasi) and it can only be started if the "addSubDelegate" link was added.

If the user clicks on that button a view opens up prefilled with copies of the original mandates:



The user is allowed to:

- Set the delegate. The delegate can be both a natural person and a legal person.
- Edit the start date as long as it is the same or later than the original start date.
- Edit the end date as long as it is earlier than the original end date. The end date can be set to infinity (tähtajatu) only if the original mandate had the end date set to infinity.
- Delete a role from the list - this means that only a portion of roles get sub-delegated and the role that was deleted from the list will not be sub-delegated.

The user is restricted from:

- Changing the roles
- Adding a new mandate on this screen
- Setting a start date to the past
- Setting a start date to an earlier date than the original start date.
- Setting the mandate to be allowed for sub-delegation (User interface doesn't display this option)
- Setting the end date to be earlier than today.
- Setting the end date to a later value than the original value.

2.7.2 Digitally signing the request

- if at least one role to be sub-delegated has RoleDefintion.subDelegatingMustBeSigned=true (see chapter 3.3.2 for detailed info) then the user must first digitally sign this request and only after the user has signed then the process continues

2.7.3 Path parameters

The parameters of the payload are set by the response that loaded the mandates. The parameters are described in paragraph 3.6.3.

2.7.4 Payload

- subDelegate is of type Person (described in chapter 3.1)
- validityPeriod from is only present if it was changed by the user. If it is not present then it must be set to today's date.
- validityPerod through is the last day when the sub-delegated mandate is valid. this cannot exceed the validityPeriod->through of the initial mandate.
- validityPerod through without a value (null) means it is valid indefinitely. This is only allowed if the original mandate was valid indefinitely

Query to be sent out by Pääsuke

```
{
  "subDelegate": {
    "type": "NATURAL_PERSON",
    "firstName": "Jüri",
    "surname": "Juurikas",
    "identifier": "EE38302250123"
  },
  "validityPeriod": {
    "from": "2017-07-21",
    "through": "2024-02-21"
  },
  "authorizations": [
    {
      "userIdentifier": "EE39912310123",
      "hasRole": "BR_REPRIGHT:JUHL_SOLEREP"
    }
  ],
  "document": {
    "uuid": "5b72e01c-fa7f-479c-b014-cc19efe5b732",
    "singleDelegate": true
  }
}
```

The presence of "document" indicates that the request was digitally signed (see Chapter 3.11). Signing support will be added in Pääsuke version 2.0

2.8 View "Ettevõtte <ettevõttenimi> poolt antud volituste edasivolitused"

NB! The following is important to only those external parties who plan to offer the functionality of adding sub-delegates.

In the future, Pääsuke will have a view to see a list of sub-delegators. Let's look at the following screenshot:

The screenshot describes the following situation. Väikefirma OÜ (EE1111111) has given roles "GLOBAL1_EMTA:Accountant" and "PRIA: DocumentViewer" to Raamatupidajad OÜ (EE23456789) with the right to sub-delegate these roles (so Raamatupidajad OÜ can further delegate it to its employees).

Now Raamatupidajad OÜ has sub-delegated (by pressing "Volita edasi") this role to its employees Reijo Raamatukogu and Raili Raamatukoi.

Now a representative of Raamatupidajad OÜ wants to know to whom Raamatupidajad OÜ has sub-delegated these mandates. For that the representative opens "Ettevõttele antud volitused" in the row of "Väikefirma OÜ" he clicks "List sub-delegators" (Vaata edasivolitusi).

UI asks the back end to perform the following query to several external parties

CURL query

```
curl -X GET \
     "https://security-server/r1/ee-dev/GOV/70006317/volitused-mock/volitused-estat/v1/representees/EE1111111
/delegates/mandates
      ?subDelegatedBy=EE23456789"
```

As mentioned in chapter 2.3.4 it is the same query described in chapter 2.3 with additional query parameter "subDelegatedBy".

The query returns Raamatupidajad OÜ employees Reijo Raamatukogu and Raili Raamatukoi as these mandates were subdelegated by Raamatupidajad OÜ (EE23456789).

If Väikefirma OÜ has given mandates to other parties then they are not returned (since in the query there is "subDelegatedBy" filter parameter in place).

2.9 Support for TÖR (Töötamise Register) use-case

One of the problems Pääsuke wants to resolve is that often times when an employee leaves a company, then nobody will remove the left person's mandates. To overcome this issue Pääsuke has taken the following aspect (that has not yet been implemented).

1. 11 days after person P has left company C a background process in the employment registry (TÖR) will send a notification to Pääsuke over X-road.
    a. This notification is only sent if the person has not been re-registered with company C within this 10-day period
    b. Also there are a lot of other exceptions - if person is still in the management board etc.
2. Pääsuke will make queries to detect if the person still has any mandates.
    a. Pääsuke will make queries "getRepresenteeDelegatesWithMandates" described in chapter 2.3 with the following changes:
        i. Pääsuke will add one more query parameter ?delegate=EE12345678901 to indicate that Pääsuke is only interested in mandates with this delegate.
        ii. Pääsuke will not add the header parameter "X-Road-UserId" to this query as this is a query triggered by a background process and not a human being.
3. If Pääsuke finds any valid mandates for person P under company C it will send an email to company C informing them about the situation.

3. Types

3.1 Person

Representee or delegate.

Property	Mandatory	Type	Description
type	mandatory	enum(LEGAL_PERSON, NATURAL_PERSON, OTHER, UNKNOWN)	Pääsuke uses this type to display an icon next to the person's name.
firstName	nullable	string	Given names of a natural person. It is not returned together with legalName.
surname	nullable	string	The surname of a natural person. It is not returned together with legalName.
legalName	nullable	string	Legal person name. It is not used together with firstName and surname.
identifier	mandatory	string	See chapter 3.1.1

3.1.1 Person identifier

The maximum length of this property is 256 symbols.

The need for a standard comes from the fact that Pääsuke in its user interface groups together mandates (received from different e-services) of the same person:

Each identifier belongs to one of two groups:

1. Two-letter country code (ISO 3166 ALPHA-2) in capital letters followed by person code (see below about person code)
    a. EE followed by an 8-digit legal entity code from Estonian Business Registry (Äriregister)
        - example: "EE70006317"
    b. EE followed by an 11-digit national identity number
        - example: "EE60001019906"
    c. two-letter country code followed by eIDAS identification (1...254 symbols) - this is returned by Tara
        - example: "CZ29d18705-fe88-4b23-9b4c-c073ae12673c"
2. URI - https://en.wikipedia.org/wiki/Uniform_Resource_Identifier . Any valid URI is allowed.

a. urn:uuid:{UUID} is recommended if the ID is generated by the party itself. Pääsuke doesn't group such identifiers (for example if Tax and Customs Board and Statistics Estonia return details about a delegate with equal UUID then Pääsuke won't group these records)
    i. example: "urn:uuid:6e8bc430-9c3a-11d9-9669-0800200c9a66"
b. URI for e-mails and phone numbers
    - example: "mailto:John.Doe@example.com" - recommended format for emails (Pääsuke ignores case when grouping)
    - example: "tel:+37251234567" - recommended format for phone numbers
c. If two different parties use the same identifiers (for example if Statistics Estonia and Agricultural Registers and Information Board (PRIA) would like to express the same person they would have to agree on common URN)
    i. urn:{agreed urn value}

3.2 Namespace

Namespace codes are given out by RIA and they cannot contain a slash, colon, semicolon, or space.

3.3 Role and RoleDefinition

3.3.1 Role and 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.

3.3.2 RoleDefinition parameters

Some of the features described in this document are planned to be supported in the future. For this colored stars are used to indicate when Pääsuke will start to support some specific feature:

- Version 1.0 - available in summer 2023
- Version 1.1 - gets to production in October 2023
- Version 2.0 - gets to production in February 2024

Note: all boolean parameters default to false.

Property	Available starting version	Man-da-tory	Type	Precondition (s)	Description
code	1.0	yes	string		*Unique identifier of the role. The prefix until the first colon is called the role namespace. Max length 4000 characters.*
title	1.0	yes	Translation (see chapter 3.7)		*Role title in different languages.*
delegateType	1.0	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 "Käibedeklaratsiooni (KMD) andmete saatmine masin-masin liidese vahendusel (code XT_MM_KMD)".* See Chapter 2.2.3 for more info.
representeeType	1.0	yes	enum [NATURAL_PERSON, LEGAL_PERSON, GOVERNMENT_PERSON]		*Type of representees who can add a mandate with this role. GOVERNMENT_PERSON is a sub-type of LEGAL_PERSON whose Estonian registry code starts with 7. Since LEGAL_PERSON includes GOVERNMENT_PERSON it is never needed to list both types for the same role.*
addableBy	1.0	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.* **Note about natural persons** *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).* **Assigning a role through Admin Portal (MISP2)** Pääsuke allows adding roles through its MISP2 based admin portal (this is a different portal than eesti.ee). 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	2.0	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 representee 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 representee must have the role "PRIA:PRIA. customer"* Before 0.9.0 this was named assignableOnlyIfRepresenteeHasRoleIn
addingMustBeSigned	2.0	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.*
canSubDelegate	2.0	no	boolean		*If this is set as true then a mandate with this role can be added with the right to further sub-delegate it.* If this parameter is changed from true to false (which is not recommended) for a role then existing mandates and sub-delegates are left intact but it wouldn't be possible to add new sub-delegates. The delegates who have previously received this mandate with the right to sub-delegate should be notified that they cannot execute this right anymore (this notifying has to be carried out outside Pääsuke as there is no automatic process in place for this).
delegateCanEqualToRepresentee	2.0	no	boolean	addableBy contains "ADMIN_PORTAL: ADMIN_USER".	*If this is set to true then in the Pääsuke Admin Portal it is possible to create mandates with this role where the value of the representee equals the value of the delegate. This functionality is used for the eesti.ee RR partner services. These types of roles where representee=delegate can only be added and deleted from the Admin portal (so in eesti.ee they are displayed as read-only).*
description	1.0	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	1.0	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	2.0	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.*
validityPeriodThroughMustBeUndefine	2.0	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.*
subDelegateType	2.0	no	enum [LEGAL_PERSON, NATURAL_PERSON]	canSubDelegate=true	*Type of persons this role can be sub-delegated to. This should have a value if canSubDelegate is set to true and normally (except in rare corner cases) the value will be NATURAL_PERSON.*
subDelegableBy	2.0	no	list of strings	canSubDelegate=true	*In order to add a sub-delegate for a mandate with this role the user representing the delegate must have a valid mandate with a role in this list. This should have a value if canSubDelegate is set to true.*
subDelegatingMustBeSigned	2.0	no	boolean	canSubDelegate=true	*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	1.1	no	list of strings		*The user representing the delegate must have a valid mandate with a role in the list to waive this mandate from the delegate side (volitusest loobumine). If this is set to null or to an empty list then this mandate cannot be waived.* **Some examples** 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. 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. Note: before version 0.9.0 there existed a boolean parameter named deletableByDelegate
waivingMustBeSigned	2.0	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	1.1	no	list of strings		*The user representing the representee must have a mandate with a role in the list to withdraw this mandate from the representee side (volituse tagasivõtmine). This should have a value if canSubDelegate is set to true.* Note: before version 0.9.0 this parameter was named deletableBy
withdrawalMustBeSigned	2.0	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).*

Types deprecated starting 0.9.0, removed starting 0.9.3

Property	Replaced with
assignableBy	addableBy
assignableOnlyIfRepresenteeHasRoleIn	addableOnlyIfRepresenteeHasRoleIn
deletableBy	withdrawableBy
deletableByDelegate	waivableBy
visible	visible: false replaced with "hidden:true"

modified (removed in 0.9.4)	-

3.3.3 Representation rights loaded from Business Registry

Note: The following applies to any configuration parameter that takes a list of roles: addableBy, waivableBy, withdrawableBy, subdelegableBy etc.

Pääsuke loads representation rights from the Business Registry (Äriregister), this information is stored under the namespace BR_REPRIGHT.

So it is possible to indicate that a role can be added only by management board members (juhatuse liikmed). For this one can define:

addableBy = BR_REPRIGHT:JUHL

There are some legal entities in Estonia that have stated that more than one management board member can represent the legal entity. So usually we also need to check that the person has a right to represent the company alone. That is why a more realistic definition would be:

addableBy = BR_REPRIGHT:JUHL_SOLEREP

This indicates that the role can be added by anyone who is a management board member and has the right to represent alone.

If one wants to allow several roles (PROK and ASES in addition to JUHL), the above definition would be changed into:

addableBy = BR_REPRIGHT:JUHL_SOLEREP, BR_REPRIGHT:PROK_SOLEREP, BR_REPRIGHT:ASES_SOLEREP

This indicates that the person could have a PROCURIST representation right instead or the person could have the role "ASES" which means that the legal entity is a state or a local government body (registry code starts with 7) and this person has the right to represent that legal entity.

NB! By law the role ASES only has an informative meaning in Äriregister. So keep that in mind when adding that to the list of supported representation rights.

3.4 MandateTriplet

This is called a triplet has it always has 3 components:

Property	Mandatory	Type	
representee	yes	Person (see paragraph 3.1)	The person being represented by the delegate
delegate	yes	Person (see paragraph 3.1)	The person who has the right to represent the representee
mandates	nullable	array Mandate (see paragraph 3.5)	List of mandates that the delegate has for this representee. NB! In order to use reactive processing on both ends the list of mandates in one MandateTriplet is allowed to be up to 100. If one delegate has more than 100 mandates for a representee then additional MandateTriplet(s) must be returned. For example, if the delegate has 121 mandates under a representee then two MandateTriplets would be returned, the first MandateTriplet with 100 mandates and the second MandateTriplet with 21 mandates.

3.5 Mandate

Property	Mandatory	Type	
namespace	yes	namespace code (see paragraph 3.2)	
role	yes	role code (see paragraph 3.3)	
validityPeriod	no	ValidityPeriod (see paragraph 1.7)	
subDelegatorIdentifier	no	Person identifier (see chapter 3.1.1)	If this mandate was created using sub-delegating then this field points to the (legal or natural) person who had the original mandate. NB! If this mandate was sub-delegated by a legal entity then the field must point to the identifier of the legal entity and it must not point at the natural person (like a board member or that legal entity) who actually carried out the adding of the sub-delegate.
links	no	MandateLinks (see paragraph 3.6)	links are used to indicate what the user can do with the mandate

3.6 MandateLinks

MandateLinks is a key-value mechanism that allows the provider of the query to indicate what actions can be done with the mandate in the Pääsuke UI.

The list of properties is fixed but new keys might be added at over time.

The value of each property has to follow a pre-defined format, but the format lets the provider of the query use identifiers inside the value

All the keys of this type are nullable so if some action is not supported by the mandate then the corresponding value of the key is null (or not included at all in the response).

Property name	Format of the value NB! Everything that is not surrounded by curly brackets is fixed.
delete	**/representees/**{representeeId}**/delegates/**{delegateId}**/mandates/**{mandateId}
addSubDelegate	**/representees/**{representeeId}**/delegates/**{delegateId}**/mandates/**{mandateId}**/subdelegates**
origin	Reserved for future. Pääsuke would open a new window/tab for the USER with this URL to view the mandate at its origin.

3.6.1 delete

If this property is present with a non-null value it indicates that the mandate can be deleted using Pääsuke (this means withdrawing a mandate or waiving a mandate).

If the property is missing or null then Pääsuke forbids the user from removing this mandate.

If the user confirms removing this mandate from Pääsuke then Pääsuke sends out the deleteMandate (see paragraph 2.6) query using parameters parsed from the value.

The value of the "{ns}" has to match the namespace of the role of the mandate.

Fragment of example output

```
"links": {
    "delete": "/v1/representees/1234/delegates/5678/mandates/901234"
}
```

When the user decides to delete the mandate from Pääsuke then Pääsuke sends out the following query to the same party that returned the response.

So if the mandate to be deleted was served to Pääsuke by "ee-dev/GOV/70006317/volitused-mock/volitused-emta" then Pääsuke sends out the following query:

Query to be sent out by Pääsuke

```
curl -X 'DELETE' 'https://security-server/r1/ee-dev/GOV/70006317/volitused-mock/volitused-emta/representees/1234
/delegates/5678/mandates/901234'
```

3.6.2 addSubDelegate

If this property is present with a non-null value it indicates that the mandate can be further sub-delegated.

If the property is missing or null then Pääsuke forbids the user from sub-delegating this mandate.

If the role definition metadata states that the role cannot be sub-delegated then Pääsuke forbids the user from sub-delegating this mandate even if this property is present in the output.

If the user sub-delegates this mandate in Pääsuke then Pääsuke sends out the addMandateSubDelegate (see paragraph 2.7) query using parameters parsed from the value.

The value of the "{ns}" has to match the namespace of the role of the mandate.

Fragment of example output

```
"links": {
    "addSubDelegate": "/representees/R987/delegates/D654/mandates/M321/subdelegates"
}
```

When the user adds a sub-delegate then

So if the mandate to be deleted was served to Pääsuke by "ee-dev/GOV/70006317/volitused-mock/volitused-emta" then Pääsuke sends out the following query:

Query to be sent out by Pääsuke

```
curl -X 'POST' \
  'https://security-server/r1/ee-dev/GOV/70006317/volitused-mock/volitused-emta/GLOBAL1_EMTA/representees
/EE10391131/delegates/EE38302250123/mandates/M321/subdelegates' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' payload of the message is described in paragraph 2.7
```

3.6.3 update

This is reserved for the future to update the record's validity period and the record's boolean flag "can add sub-delegate".

3.6.4 origin

URL to self-service where the mandate information can be displayed to the user.

This is reserved for the future.

3.7 Translation

Pääsuke runs within eesti.ee portal that is offered to end users in Estonian, English, and Russian.

Returning translations in Estonian is mandatory.

If the English or Russian translation is missing then for that part the user interface of Pääsuke uses the Estonian translation instead.

Property name	Mandatory	Type	Description
et	yes	string	Translation in Estonian.
en	no	string	Translation in English
ru	no	string	Translation in Russian

3.8 Authorization

This list is added to some of the payloads to reflect the information on why the person doing a modification was allowed by Pääsuke to perform the action.

There are several options

3.8.1 Person has a role that allows creating, sub-delegating, and removing mandates

```
 "authorizations": [
   {
     "userIdentifier": "EE49028099999",
     "hasRole": "STAT:Peakasutaja"
   }
 ]
```

3.8.2 Person is on the board and has the right to represent the legal entity alone

```
"authorizations": [
  {
    "userIdentifier": "EE49028099999",
    "hasRole": "BR_REPRIGHT:JUHL_SOLEREP"
  }
]
```

3.8.3 Several people who are on the board and they have partial rights (ühisesindusõigus) but together they can represent the legal entity

```
"authorizations": [
  {
    "userIdentifier": "EE49028099999",
    "hasRole": "BR_REPRIGHT:JUHL_GROUPREP"
  },
  {
    "userIdentifier": "EE39211110000",
    "hasRole": "BR_REPRIGHT:JUHL_GROUPREP"
  }
]
```

This use case (ühisesindusega juhatuse liikmed saavad Pääsukese abil rolle peale panna) is not yet supported but the API is already designed to be able to support it in the future

3.9 ValidityPeriod

Property	Mandatory	Type	Description
from	nullable	date	The first day (inclusive). Can be both in the past and in the future.
through	nullable	date	The last day (inclusive). If the value is missing (or null - sending nulls is discouraged) it means the end date is not specified (infinity). Normally this date can never be in the past (as Pääsuke only returns mandates that are currently valid or become valid in the future).

3.10 Problem

https://www.rfc-editor.org/rfc/rfc7807

https://blog.axway.com/learning-center/apis/api-design/introduction-to-rfc-7807

Property	Mandatory	Type	Description
type	no		An absolute URI that identifies the problem type
href	no		An absolute URI that, when dereferenced, provides human-readable documentation for the problem type (e.g. using HTML).
title	yes		A short summary of the problem type. Written in English and readable for engineers (usually not suited for non-technical stakeholders and not localized). Example: Service Unavailable
status	no		This reflects the HTTP status code and is a convenient way to make problem details self-contained. That way they can be interpreted outside of the context of the HTTP interaction in which they were provided
translation	no	Translation (chapter see 3.4)	A human-readable description of the problem *instance*, explaining why the problem occurred in this specific case. This value could and often will be displayed to the user.
ticket	no		ticket number
<future attributes>	no		adding other attributes is allowed

3.11 Document

- This section is present if the persons listed in the authorizations block added their digital signature. Signing support will be added in Pääsuke version 2.0

Property	Mandatory	Type	Description
uuid	yes	string	Document UUID. This identifier can be used to download the signed container from Pääsuke over X-road.
singleDelegate	yes	boolean	This is set to true in case the signed document contains information about a single delegate and it is safe to reveal the document to the delegate. False means that the signed container contains information about multiple delegates and it is only safe to reveal it to the representee and not to any of the delegates.

4. Details about roles and namespaces

4.1 Namespaces with parent-child relation

RIA defines global roles (in Estonian *katusrollid*) that can be supported by multiple institutions.

The idea is to provide all needed permissions in different institutions for a job as one single access role.

Let's look at this using an example. RIA has created a parent namespace "GLOBAL1" with the following configuration

Definition of a namespace		
namespace	GLOBAL1	
type	PARENT	
parent namespace	<null>	
title in Estonian	Asutusteülene	NB! This is a draft name for the role
title in English	Across institutions	NB! This is a draft name for the role
owner	GOV/70006317	This refers to Riigi Infosüsteemi Amet. The owner refers to the party who declares roles in this namespace.

This namespace contains the following roles:

Definition of roles			
namespace	GLOBAL1	GLOBAL1	GLOBAL1
role code	GLOBAL1:Accountant	GLOBAL1:Human_Resources_Specialist	GLOBAL1:Data_viewer
title in Estonian	Raamatupidaja	Personalitöötaja	Andmete vaataja

However, it is not possible for anyone to directly grant roles to some delegate from such a parent namespace (as any role of type parent is an interface that declares some properties for any child roles).

Any institution can decide to support all or only a selection of these roles. Let's say the Estonian Tax and Customs Board (EMTA) has added support for all 3 roles.

For this Pääsuke has defined a separate namespace GLOBAL1_EMTA for EMTA:

Definition of a namespace	
namespace	GLOBAL1_EMTA
type	CHILD
parent namespace	GLOBAL1
owner	GOV/70000349

In this namespace there are definitions of the same 3 roles with the same names for the roles (descriptions are different):

Definition of roles			
namespace	GLOBAL1_EMTA	GLOBAL1_EMTA	GLOBAL1_EMTA
role (allowed values are fixed by roles in the parent namespace)	GLOBAL1_EMTA: Accountant	GLOBAL1_EMTA: Human_Resources_Specialist	GLOBAL1_EMTA: Data_viewer

title (title must be set but it is ignored by Pääsuke)	Raamatupidaja	Personalitöötaja	Andmete vaataja
description in Estonian	samad õigused, mis raamatupidaja paketis	Töötajate registry kasutamise õigused	Andmete vaataja õigused

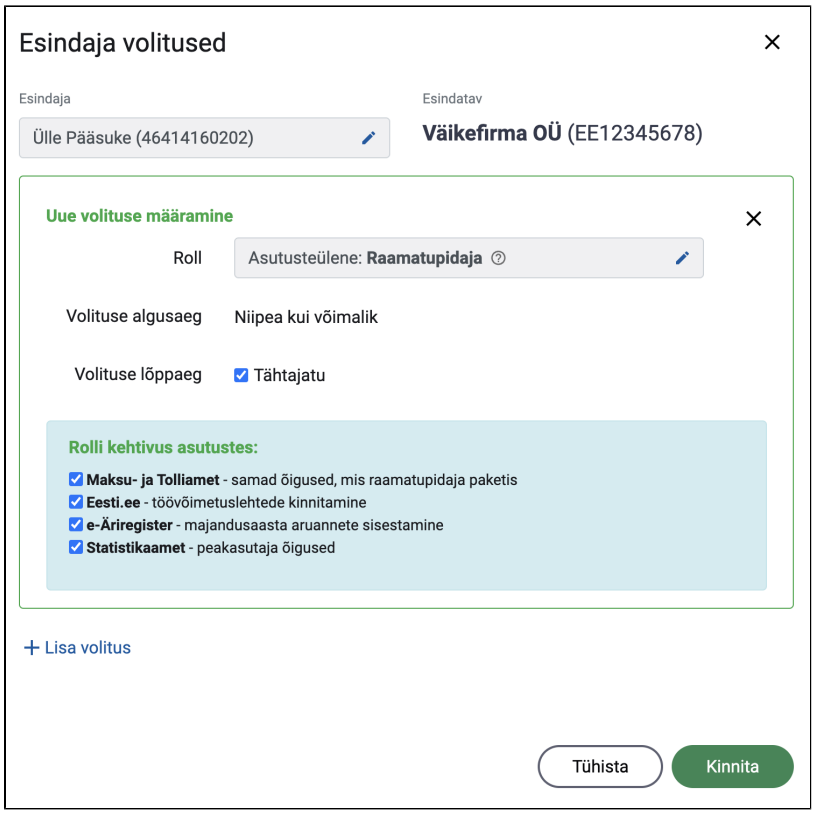
If there is some other institution (let's say Statistics Estonia (from now on STAT) that only wants to add support for the role "Accountant" then a separate child namespace will be created for that (that points to GLOBAL1 parent namespace) and STAT only declares the Accountant role in that namespace:

Definition of a namespace	
namespace	GLOBAL1_STAT
type	CHILD
parent namespace	GLOBAL1
owner	GOV/70000332

Definition of roles	
namespace	GLOBAL1_STAT
role	GLOBAL1_STAT:Accountant
title (title values of global child roles must be set but are ignored by Pääsuke)	Raamatupidaja
description in Estonian	Peakasutaja õigused

The following screenshot demonstrates how the Accountant role will be visible to the end user who starts to assign the Accountant role to some employee:

The screenshot assumes there are 4 agencies that support that role:



The person that is assigning the role can untick some of the institutions. Eventually, when the person clicks confirm (Kinnita) Pääsuke performs one POST request to each institution over the x-road to add the role for each institution.
So if the person didn't untick anything then Pääsuke will make 4 separate post requests to add 4 separate roles.

4.2 Special parent-child namespaces for Admin Portal access rights

NB! This chapter is relevant for parties who keep mandates in Pääsuke.

Let's say some government agency (Agency-X) keeps all the mandates in Pääsuke and queries them from there.

Now some person who is a management board member of some company (Company-C) sends a digitally signed request and asks for a mandate to be added for the employee (Employee-E) of his company.

The administrative user of that agency (after verifying that the management board member has the right to represent that company) needs an administrative user interface to enter this mandate into Pääsuke.

For this Pääsuke is going to offer X-road services that the agency can use using the MISP2-based Admin Portal portal of that agency.

There is a need for a mechanism for that agency to declare which of its roles can be assigned or deleted via Admin Portal.

For this RIA has created a parent namespace namespace "ADMIN_PORTAL" and added one role "ADMIN_USER" into it.

The agency (Agency-X) that wants its administrative person to add the mandates from Admin Portal portal needs to declare a child namespace:

ADMIN_AGENCYX and adds a role into it (ADMIN_AGENCYX:ADMIN_USER).

And then the agency is free to modify the addableBy and withdrawableBy properties (described in the chapter 3.3.1) of that role:

For example, the following declares a role that only Admin Portal users can assign or delete:

addableBy: ["ADMIN_AGENCYX:ADMIN_USER"]

withdrawableBy: ["ADMIN_AGENCYX:ADMIN_USER"]

4.3 Dedicated namespaces

These namespaces are used in Authorization (see Chapter 3.8) and also in RoleDefinition as values in addableBy, withdrawableBy, waivableBy, subDelegableBy.

4.3.1. Dedicated namespace for representation rights loaded from Business Registry (Äriregistrist)

BR_REPRIGHT is a namespace in Pääsuke that indicates that the representation rights are loaded by Pääsuke from Business Registry.

4.3.2. Dedicated namespace for representation rights between natural persons

NAT_REPRIGHT is a namespace in Pääsuke that indicates that the natural person has a right to represent (esindaja teovõime ei ole piiratud) another natural person (that could be oneself).