Pääsuke X-Road API descriptions

1. Introduction

1.1 Versions

Version	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 indroduced 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	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.	

1.2 Preface

Pääsuke is a system designed for the following use-cases

1. Smaller-scale e-services use Pääsuke to store the mandates centrally.
   a. Pääsuke offers x-road services to these smaller-scale e-services to query mandates from Pääsuke when a person tries to authorize in that service
   b. Pääsuke also offers a list of persons (like management board members, procurers, etc) defined in the Business Registry as a response to this query
2. Pääsuke displays the mandates that are stored in Pääsuke and in different e-services in Estonia
   a. To fulfill this goal medium-scale and larger-scale e-services offer x-road endpoints for Pääsuke (and such endpoints are standardized by Pääsuke)
   b. Pääsuke queries the mandates and displays them to the representees and to the delegates that have signed into eesti.ee
3. For some systems, Pääsuke offers a combination of these two integration patterns.

This is why there are many endpoints. When reading this document it is important to keep in mind which pattern is going to be implemented.

1.3 Terminology

- **Pääsuke** - central access rights management system hosted in eesti.ee
- **RIA** - Riigi Infosüsteemi Amet, agency that develops and runs eesti.ee and Pääsuke
- **institution** - some party who has a self-service system that either pulls mandates from Pääsuke and/or has mandates declared in the system and publishes them in Pääsuke.
- **representee** (previously named represented person) - the person (most commonly a private limited company) that can be represented in e-services
- **delegate** - the person who has been given the mandate to represent some representee. 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 provider. Read more from chapter 1.2.1
- **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 chapter 1.2.2
- **mandate** - a role that is given to a delegate under some representee. Mandates can have a start date, and end date and some mandates can be sub-delegated (in Estonian "*edasi delegeerima*").

1.3.1 What is a namespace

In this document names of namespaces are written in capital letters. If a namespace is written in pair with a role then we use a colon as a separator.

Any institution can own and query roles from multiple namespaces.

Namespaces can be standalone (regular namespaces) or form parent-child relationships to provide grouping (in Estonian "*katusrollide funktsionaalsus*").

- STANDALONE namespace (regular namespace)

- PARENT namespace - used to group together global roles ("katusrollid"), in programming terms each role in the parent namespace is like an interface
- CHILD namespace (in programming terms each role in the child namespace is like an implementation of a global role and it is allowed to have more than one implementation of the same role)

Besides these types, there are AUTOMATIC namespaces. Roles in automatic namespaces are automatically assigned based on external property, for example, a role in such a namespace could be assigned automatically to a person after becoming a management board member.

There are several options for organizing roles into namespaces:

- Several institutions come together and together form a single namespace.
  - For example institutions active in agriculture could come together and define a common namespace "AGRICULTURE"
  - Institutions would agree with themselves which institution is maintaining the roles in this namespace
  - this is the most recommended solution for new systems as this is the easiest to understand for anyone operating in this field.
- Every institution always has at least one namespace (with their name)
  - This namespace could cover all (or most of) the roles of that institution. For example, the Estonian Tax and Customs Board has the namespace "EMTA".
- For one institution it is also possible to have multiple standalone namespaces.
  - This is only recommended for existing applications and not for new ones.
  - For example, Statistikaamet is going to have separate namespaces for two different systems: "STAT-ESTAT" and "STAT-NEW".
- An institution could have an additional child namespace
  - Such a namespace that implements some parent namespace is described in chapter 1.2.3.

1.3.2 What is a role

Roles usually have descriptive code that indicates the profession that needs that role (Accountant) or what the owner can do (like Job.Ad.Editor).

Pääsuke is not limiting what the owner of the role can do in any of the systems that are pulling roles from Pääsuke.

If some institution AgencyX declares a role Editor and then it is solely the responsibility of Agency X to:

- Decide what anyone who has a mandate for that role can do
- Declare the code, title, and description of that role

Properties of a role:

- the role belongs to a namespace
- the role has a code (unique identifier, not shown out) that is unique in that namespace
- the role has a title in Estonian (and possibly also in English and Russian)
- the role can have a description (in any or all of the languages Estonian, English, and Russian)

1.3.3 Namespaces with parent-child relation

RIA defines global roles (in Estonian *katusrollid*) that can be supported by multiple institutions.
The idea behind this is that some of the roles repeat in many institutions and Pääsuke wants to hide the complexity from the persons giving mandates.

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! See on töönimi)
title in English	Across institutions (NB! See on töönimi)

This namespace contains the following roles:

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

However, it is not possible for anyone to grant roles to some delegate from such a namespace (that is declared as type=PARENT).

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

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	Accountant	Human_Resources_Specialist	Data_viewer
title in Estonian	Raamatupidaja	Personalitöötaja	Andmete vaataja
description in Estonian	Samad õigused, mis raamatupidaja paketis		

If there is some other institution (let's say STAT) that only wants to add support for Accountant then a separate child namespace is created for that (that points to GLOBAL1 parent namespace) and STAT only declares Accountant role in that namespace:

Definition of a namespace	
namespace	GLOBAL1_STAT
type	CHILD
parent namespace	GLOBAL1

Definition of roles	
namespace	GLOBAL1_STAT
role	Accountant
title in Estonian	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:

Esindaja volitused　×

Esindaja

| Ülle Pääsuke (46414160202) | ✏ |

Esindatav

Väikefirma OÜ (EE12345678)

Uue volituse määramine　×

| Roll | Asutusteülene: **Raamatupidaja** ⓘ | ✏ |

Volituse algusaeg　Niipea kui võimalik

Volituse lõppaeg　☑ Tähtajatu

Rolli kehtivus asutustes:

☑ **Maksu- ja Tolliamet** - samad õigused, mis raamatupidaja paketis
☑ **Eesti.ee** - töövõimetuslehtede kinnitamine
☑ **e-Äriregister** - majandusaasta aruannete sisestamine
☑ **Statistikaamet** - peakasutaja õigused

+ Lisa volitus

[Tühista]　[Kinnita]

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 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.

1.4 Notes

1.4.1 Playing with the mock services 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.

From the HTTP request examples throughout this document the X-Road specific headers (headers beginning with X-Road-...) are removed.

If you want to test te 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-User-Id: EE39912310123" \
-H "X-Road-Id: 08544bbd2f41473800309d16bd81c64c0f54193d84b53f8ad287446a2aacdf5e" \
-X GET "https://security-server/r1/ee-dev/GOV/70006317/volitused-mock/volitused-estat/v1/roles"
```

You need to make 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-User-ID to your own personal id code.

1.5 Different types of X-Road services in Pääsuke

When considering X-Road services they come in two different types:

- X-Road queries that are consumed by Pääsuke itself. There are two types:
  - "standardized services" are used by Pääsuke to get mandate-related information from systems that keep mandates on their side (like EMTA and STAT). These services used standardized API so Pääsuke could dynamically add new providers in future.
  - all kinds of x-road services used by Pääsuke. These services are not standardized and not covered in this document.
- X-Road queries that are provided by Pääsuke. There are two types of such services:
  - services to query mandate info
  - services to add mandate info into Pääsuke (employee of an e-service who keeps all its mandates in Pääsuke can insert mandates into the system)

1.6 Types

1.6.1 Person

Representee or delegate.

Field name		
type	mandatory	LEGAL_PERSON or NATURAL_PERSON
firstName	nullable	given names of a natural person
surname	nullable	the surname of a natural person
legalName	nullable	legal person name

identifier	mandatory	NB! Maximum length of this identifier is 256 symbols. Whitespaces should not be used in the identifier.
		
		If the person type is LEGAL_PERSON
		
		the value is "EE" + 8-digit Estonian registry code of the legal entity in the Estonian business registry.
		For example EE12345678
		
		Note about representees of type LEGAL_PERSON
		
		Pääsuke does not collect or query mandates from external systems about representees of type LEGAL_PERSON who are business
		entities registered in other countries besides Estonia.
		
		Note about delegates of type LEGAL_PERSON
		
		When a delegate is a legal person who is not registered in the Estonian business registry (as a separate business entity) then the response has to use the prefix "internal:" + 1...247 symbols.
		
		If the person type is NATURAL_PERSON
		
		The identifier has to be one of:
		
		- "EE" + 11-digit Estonian national identity number
		  - example: "EE3730112310123"
		- uppercase two-letter country code (according to ISO 3166-1 alpha-2 standard) + eIDAS identification (1...254 symbols)
		  - example: "CZ29d18705-fe88-4b23-9b4c-c073ae12673c"
		- "email:" + e-mail address of the natural person
		  - example: "email:mari.maasikas@eesti.ee"
		- "internal:" + (1...247 symbols) - in case the identifier of the natural person is made up using internal rules
		  - example: "internal:EE12111111"

1.6.2 Namespace

Namespace code cannot contain a slash, colon, semicolon, or space

1.6.3 Role

Field	Mandatory	Type		
namespace	yes	string		

code	yes	string (see chapter 1.6.3.1)		
title	yes	Translation (see chapter 1.6.4)		
description	no	Translation (see chapter 1.6.4)		
metadata	no	RoleMetadata (see chapter 1.6.3.2)		

1.6.3.1 Role code

The role code must be unique in the namespace (using case-insensitive comparison!)

Role codes can contain the following symbols:

- uppercase letters (A-Z)
- lowercase letters (a-z)
- numbers (0-9)
- underscore (_)
- dash (-)
- a dot (.)
- hash (#)
- a colon(:)

NB! In the parameters of some queries, the roles have to be prefixed with their namespace that is separated by a colon. This is why implementations have to keep in mind that roles can also contain colons - splitting has to happen only using the first colon in such a pair.

1.6.3.2 RoleMetadata

Field	Mandatory	Type	Description
state	no	enum [ACTIVE, ACTIVE_GOVERNMENT, HIDDEN, REMOVED]	Role is ACTIVE if it can be assigned; ACTIVE_GOVERNMENT if the role can be assigned only to representees that are government agencies (representee registry code starts with 7); HIDDEN if the role is still supported but cannot be assigned to new employees; REMOVED if the role is no longer in action
modified	no	date-time	When this role definition was last modified
canSubDelegate	no	boolean	Can this role be given out with the right to sub-delegate it further
assigneeType	no	enum [LEGAL_PERSON, NATURAL_PERSON]	What type of persons this role can be assigned to.
assignableBy	no		List of namespace:role combinations who are allowed to assign this mandate

1.6.4 MandateTriplet

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

Field	Mandatory	Type	
representee	yes	Person (see paragraph 1.6.1)	The person being represented by the delegate
delegate	yes	Person (see paragraph 1.6.1)	The person who has the right to represent the representee
mandates	yes	array Mandate (see paragraph 1.6.5)	List of mandates that the delegate has for this representee

1.6.5 Mandate

Field	Mandatory	Type	
namespace	yes	namespace code (see paragraph 1.6.2)	
role	yes	role code (see paragraph 1.6.3)	
validFrom	no	date without time	
validThrough	no	date without time	last day the mandate is valid
links	no	MandateLinks (see paragraph 1.6.6)	links are used to indicate what the user can do with the mandate

1.6.6 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.
xRoadDeleteMandate	**/ns/**{ns}**/representees/**{representeeIdentifier}**/delegates/**{delegateIdentifier}**/mandates/**{mandateIdentifier}
xRoadPostSubDelegate	**/ns/**{ns}**/representees/**{representeeIdentifier}**/delegates/**{delegateIdentifier}**/mandates/**{mandateIdentifier}**/subdelegates**
uiExternalView	has to be a valid URL
uiExternalEdit	has to be a valid URL

1.6.6.1 xRoadDeleteMandate

If this property is present with a non-null value it indicates that the mandate can be removed using Pääsuke.

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 removeMandateFromDelegate (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": {
    "xRoadDeleteMandate": "/ns/EMTA/representees/1234/delegates/5678/mandates/901234"
}
```

When the user decides to remove 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/ns/EMTA
/representees/1234/delegates/5678/mandates/901234'
```

1.6.6.2 xRoadPostSubDelegate

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": {
    "xRoadPostSubDelegate": "/ns/GLOBAL1_EMTA/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
```

1.6.6.3 uiExternalView

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

1.6.6.4 uiExternalEdit

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

1.6.4 Translation

TODO

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 the representee has visibility all over the Estonian e-services of the mandates that are currently valid.

Pääsuke uses the following services to query systems that among other things store mandates. These services are called standard services and 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.

2.2 Query "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"
```

This query returns all the roles with translations in Estonian, English, and Russian.

The role name is not displayed to the user, it is the identifier of the role. It is recommended to use short names for roles.

When 'If-Modified-Since' is included in the request the service responds with HTTP Status code 304 if nor roles have been changed since that time. Otherwise, all results (that match the filters) are returned (even the ones that have modified time earlier than the If-Modified-Since parameter).

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, who have a right to represent the representee currently or in the future.

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).

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

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

Screenshot describes the following situation. Väikefirma OÜ (EE12345678) has given roles GLOBAL: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 parties who support Accountant mandate:

CURL query

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

And the query returns Raamatupidajad OÜ employees Reijo Raamatukogu and Raili Raamatukoi as these mandates were subdelegated by Raamatupidajad OÜ (EE22222222).

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).

2.3.3 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 Response structure of query getRepresenteeDelegatesWithMandates

This is described in paragraph 2.5

2.4 Query getDelegateRepresenteesWithMandates

GET

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

Returns all representees that the delegate has the right to represent.

This query serves the following UI-s.

2.4.1 View "Ettevõttele antud volitused"

Ettevõte (näit raamatupidamisbüroo) vaatab, millised teised ettevõtted talle on volitusi andnud.

Punased Õhupallid (EE12345678)

Ettevõtte esindajad ja volitatud isikud | **Ettevõttele antud volitused**

🏢 **Pikad Puud OÜ** (EE88765432)   Ettevõtlus: **Andmete edastaja** ⑦   Muuda Volita edasi

in the future, there will be UI to display the list of persons that points to view "Ettevõtte <ettevõttenimi> poolt antud volituste edasivolitused" described above.

2.4.2 View "Mulle antud volitused"

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).

Mari Maasikas (49028099999)

Mulle antud volitused | Minu esindajad

🏢 **Raamatupidajad OÜ** (EE23456789)   Äriregistrist: **Juhatuse liige (ühisesindus)** ⑦   Vaata edasivolitusi Volita
                                         Statistikaamet (eSTAT): **Andmeesitaja** ⑦   edasi Loobu
                                         Asutusteülene: **Raamatupidaja** ⑦

🏢 **Väikefirma OÜ** (EE12345678)   Äriregistrist: **Juhatuse liige** ⑦   Vaata

2.4.3 At least one of the mandates has a link that allows adding a sub-delegate

The user (legal person or a natural person) is viewing the list of mandates that have been given by different representees.

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

2.5 Response structure of queries getRepresenteeDelegatesWithMandates and getDelegateRepresenteesWithMandates

Both queries have the same response structure.

This is list of MandateTriplets.

2.6 Query removeMandateFromDelegate

```
curl -X DELETE \
     "https://security-server/r1/ee-dev/GOV/70006317/volitused-mock/volitused-estat/v1/ns/{ns}/representees/
{representeeIdentifier}/delegates/{delegateIdentifier}/mandates/{mandateIdentifier}"
```

```
Removes role from a delegate in the given namespace (denoted {ns}).
```

Values of representeeIdentifier, delegateIdentifier, and mandateIdentifier are taken by Pääsuke from the output of the query that produced the list (link with rel "xroadDelete").

2.6.1 View to remove mandates from delegate

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

Esindaja volitused ✕

Esindaja
Raamatupidajad OÜ (EE23456789)

Esindatav
Väikefirma OÜ (EE12345678)

Asutusteülene: **Raamatupidaja** ⓘ

Rolli kehtivus järgnevates asutustes:
☑ **Maksu- ja Tolliamet** - samad õigused, mis on raamatupidaja paketis *(alates 14.12.2022 - tähtajatu)*
☑ **Eesti.ee** - töövõimetuslehtede kinnitamine *(alates 14.12.2022 - 31.12.2023)*
☑ **e-Äriregister** - majandusaasta areuannete sisestamine *(alates 20.12.2022 - tähtajatu)*
☐ **Statistikaamet** - peakasutaja õigused

Märgitud eemaldamisele ✕

PRIA: **Dokumentide vaataja** ⓘ 6.11.2020 - 31.12.2021
Alusdokument

Märgitud eemaldamisele ✕

+ Lisa volitus Eemalda kõik

Tühista Kinnita

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

2.7 Query addMandateSubDelegate

2.7.1 View where adding a sub-delegate can be started

The user (legal person or a natural person) is viewing the list of mandates that have been given by different representees.

If at least one of the mandates has a flag to sub-delegate, then the UI adds a button to initiate adding a sub-delegate ("Volita edasi")

2.8.2 Path parameters

The parameters of the payload are described in paragraph 1.6.6.2.

2.8.3 Payload

subDelegate is of type Person

validThrough is the last day when the sub-delegated mandate is valid. this cannot exceed the validThrough date of initial mandate.

validIndefinitely is true if the validThrough date is not limited.

Query to be sent out by Pääsuke

```
{
  "subDelegate": {
    "type": "NATURAL_PERSON",
    "firstName": "Jüri",
    "surname": "Juurikas",
    "legalName": "Juurikas",
    "identifier": "EE38302250123"
  },
  "validIndefinitely": false,
  "validThrough": "2024-02-21"
}
```

2.8 Query addMandatesToDelegate

2.8.1 View to add mandates to a delegate

TODO

3. X-road services offered by Pääsukese to query mandates that are stored in Pääsuke.

The following services are optimized to be queried often and returned fast. Internally a read-only replica or other similar systems might be used to ensure high availability.

NB! When playing with the mock service and testing services described in chapters 3.1 and 3.2 always set Namespace to "AGENCYX" this way the mock service responds correctly.

3.1 Query getDelegateRepresentees

```
curl -X GET
     "https://security-server/r1/ee-dev/GOV/70006317/volitused-mock/volitused-estat/v1/delegates/{delegate}
/representees
     ?namespace=AGENCYX
     &representeeType=LEGAL_ENTITY
     &hasRoleIn=AGENCYX%3AEditor"
```

Returns all representees, who the delegate has the right to represent in given namespace(s).

3.1.1 Filtering options

Parameters:

- representeeType (comma separated) - LEGAL_ENTITY (include legal entities) or NATURAL_PERSON (include individuals)
- hasRoleIn (comma separated) - filter out representees where the delegate doesn't have any of the listed roles

3.1.2 Output

List of matching representees without mandates.

Sample response:

```
[
  {
    "type": "LEGAL_ENTITY",
    "name": "Raamatupidajad OÜ",
    "identityCodeCountry": "EE",
    "identityCode": "10689305",
    "links": [
      {
        "href": "xroad:/namespaces/GLOBAL/representees/EE10689305/delegates/EE60102030405/mandates",
        "rel": "delegateMandates"
      }
    ]
  },
  {
    "type": "LEGAL_ENTITY",
    "name": "Suurfirma AS",
    "identityCodeCountry": "EE",
    "identityCode": "10234958",
    "links": [
      {
        "href": "xroad:/namespaces/GLOBAL/representees/EE10234958/delegates/EE60102030405/mandates",
        "rel": "delegateMandates"
      }
    ]
  }
]
```

3.2 Query getRepresenteeDelegateWithMandates

Query

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

3.2.1 Filtering options

Parameters:

- hasRoleIn (comma separated) - For filtering out representees where the delegate doesn't have any of the roles in the list.

3.2.2 Output

Lists active mandates of the delegate under representee.

Expired mandates or mandates that will become in effect in the future are not returned.

It is possible that in Pääsuke some end user has sevaral mandates with the same role.
These identical mandates could differ by validFrom or expiration date or other properties (right ot subdelegate).
But this service only returns every role once. This is the reason why id-s are not returned and validFrom is not returned.

If mandate will become invalid at the end of the current date only then the validThrough is included, otherwise it is null. This request is made on 31. October 2022 and the validThrough date informs that this mandate will become expired at end of today.

```
{
  "representee": {
    "type": "LEGAL_ENTITY",
    "name": "Suurfirma AS",
    "identityCodeCountry": "EE",
    "identityCode": "10234958"
  },
  "delegate": {
    "type": "NATURAL_PERSON",
    "name": "Jüri Juurikas",
    "identityCodeCountry": "EE",
    "identityCode": "60102030405"
  },
  "mandates": [
    {
      "namespace": "GLOBAL",
      "role": "ACCOUNTANT"
    },
    {
      "namespace": "GLOBAL",
      "role": "JURIST",
      "validThrough": "2022-10-31"
    }
  ]
}
```

3.3 Query getNamespaces

CURL query

```
curl -X GET
      "https://security-server/r1/ee-dev/GOV/70006317/volitused-mock/volitused-estat/v1/ns"
```

Get available namespace configuration.

Namespace name can contain:

- uppercase letters (A-Z)
- numbers (0-9)
- dash
- underscore
- dot

The namespace value is not shown to the end user.

3.3.1 Output

Some of the less important fields (translations, owner, hiddenDescription) are removed

```json
[
  {
    "namespace": "GLOBAL1",
    "parentNamespace": null,
    "title": [
      {
        "et": "Asutusteülesed katusrollid"
      }
    ],
    "type": "PARENT"
  },
  {
    "namespace": "LHV_GLOBAL1",
    "parent_namespace": "GLOBAL1",
    "title": [
      {
        "et": "LHV pank"
      }
    ],
    "type": "CHILD",
  },
  {
    "namespace": "AUTOM_BUSINESSREG",
    "parent_namespace": null,
    "title": [
      {
        "et": "Äriregistri andmete põhjal automaatselt lisatavad ja eemaldatavad rollid"
      }
    ],
    "type": "AUTOMATIC"
  }
]
```

3.4 Query 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
     ?namespace=AGENCYX"
```

This query returns all the matching roles with translations in Estonian, English and Russian.

This query needs to be called only to verify that Pääsuke has correct list of roles and the translations are correct.

3.3.1 Filtering options

- When 'If-Modified-Since' is included in the request the service responds HTTP Status code 304 if nor roles have been changed since that time. Otherwise all results (that match the filters) are returned (even the ones that have modified time earlier than If-Modified-Since parameter).
- can filter by namespace

4. X-road services to modify mandates that are stored in Pääsuke

An e-service can keep all its mandates in Pääsuke so that granting new mandates for that system would take place in eesti.ee.

But there are cases where the end user will interact directly with e-service provider so that the e-service provider has to enter the mandates manually. For example the end user could present power of attorney to get access to e-service.

For that use case there are additional x-road services to add mandates manually.

4.1 Service addMandatesToDelegate

Post request:

```
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'
```

With following payload:

```
{
  "namespace": "MYNS",
  "comment": "End user presented power of attorney to get viewer privileges",
  "mandates": [
    {
      "role": "VIEWER",
      "validFrom": "2017-07-21",
      "validThrough": "2024-02-21"
    }
  ]
}
```

If there is end date then instead of validThrough attribute the caller has to provide:

```
"validIndefinitely": true
```

This x-road service is designed to be Misp2 compatible so that these requests can be made with MISP2 portal. The User interface in MISP2 for that service looks like this:

Add mandates to a delegate under a representee

Person code or company code of the representee: `EE10391131` *

Person code or company code of the delegate: `EE38302250123` *

The reason why mandates are added manually: `End user presented power of a`

Mandates to be added:

Role: `VIEWER`

Valid from date.: `21.07.2017`

Is valid indefinitely: ☐

Valid through (last day is inclusive, used only if not valid indefinitely): `21.02.2024`

The namespace the mandates are added under: `MYNS`

Esita päring

5. Integration with Pääsuke without implementing any X-road services

Currently to most of Estonian goverment e-services the authentication is provided by Tara.

For any e-service using Tara there is no need to implement x-road services as Tara provides a 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 user to select a representee.

This way GovSSO would provide the selected representee as part of OpenID connect flow together with details of authenticated person.

If the user later wants to switch to a different representee 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.