Description of X-Road services offered to Pääsuke

Version 0.6.1

1. Introduction

This document is accompanied by OpenAPI definitions:
https://app.swaggerhub.com/apis/aasaru/paasuhalduse-x-tee-teenused/<version number>

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.

1.2 Preface

1.2.1 Use-cases resolved by Pääsuke

Pääsuke is a system designed for the following use-cases

1. Some e-services use Pääsuke to store the mandates centrally.

    a. Pääsuke offers x-road services to these e-services to query mandates from Pääsuke when a person tries to authorize himself 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 it also displays a list of mandates that are stored in other e-services (like Tax and Customs Board)
    a. To fulfill this goal other types of e-services offer x-road endpoints for Pääsuke (and such queries 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.

This document describes Use Case 2 where Pääsuke queries mandates from external systems to be displayed in eesti.ee
External parties will have to offer a set of queries to Pääsuke and the format of these queries is standardized by Pääsuke.
Following is the description of that standard.

1.2.2 Principles to follow when implementing the queries

Avoid sending null values:

Not recommended	Recommended

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

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. Read more from the chapter What is a namespace
- **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*").

1.3.1 What is a namespace

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.

In this document codes 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.

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).

If some institution (for example AgencyX) declares a role Job.Ad.Editor and then it 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
- 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

1.3.3 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

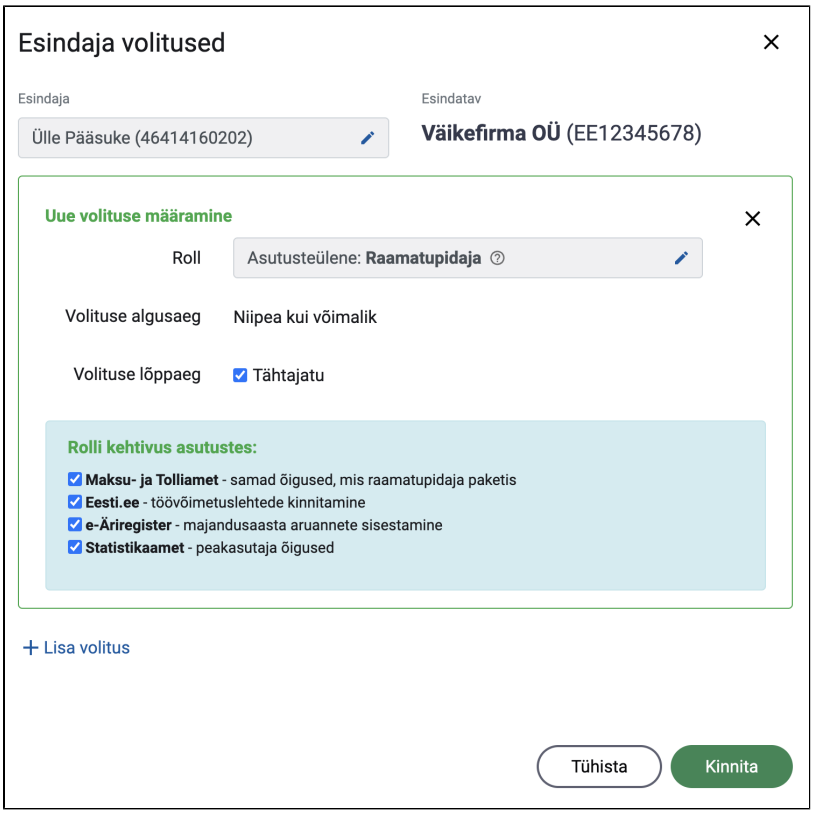
If there is some other institution (let's say Statistics Estonia (from now on STAT) that only wants to add support for the role "Accountant" then a separate child namespace will be created for that (that points to GLOBAL1 parent namespace) and STAT only declares the Accountant role in that namespace:

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.

1.3.4 Special namespace for representation rights loaded from Business Registry (Äriregistrist)

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

This namespace is used in Authorization (see chapter 1.6.5)

1.3.5 Special parent-child namespaces for MISP2 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 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 MISP2.

For this RIA has created a parent namespace namespace "MISP2" and added one role "MISP2_USER" into it.

The agency (Agency-X) that wants its administrative person to add the mandates from MISP2 portal needs to declare a child namespace:

MISP2_AGENCYX and adds a role into it (MISP2_AGENCYX:MISP2_USER).

And then the agency is free to modify the assignableBy and deletableBy properties (described in the chapter 1.6.3.1) of that role:

For example, the following declares a role that only Misp2 users can assign or delete:

assignableBy: ["MISP2_AGENCYX:MISP2_USER"]

deletableBy: ["MISP2_AGENCYX:MISP2_USER"]

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 sub-delegate 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 chapter 1.4.2).

The mock is also accessible over ee-dev X-Road (see chapter 1.4.3).

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-User-Id: 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-User-ID to your own personal id code.

1.5 Different types of X-Road services of Pääsuke

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

1. The 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 the future.
    - all kinds of x-road services used by Pääsuke. These services are not standardized and not covered in this document.
2. The 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)

This document only describes the first group - these are services that are offered to Pääsuke.

1.6 Types

1.6.1 Person

Representee or delegate.

Property	Mandatory	Type	Description
type	mandatory	enum(LEGAL_PERSON, NATURAL_PERSON)	
firstName	nullable	string	given names of a natural person
surname	nullable	string	the surname of a natural person
legalName	nullable	string	legal person name
identifier	mandatory	string	See chapter 1.6.1.1

1.6.1.1 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}

1.6.2 Namespace

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

1.6.3 Role and RoleDefinition

1.6.3.1 RoleDefinition

Property	Mandatory	Type	Description
namespace	yes	string	
code	yes	string (see chapter 1.6.3.2)	
title	yes	Translation (see chapter 1.6.7)	
description	no	Translation (see chapter 1.6.7). If the description is provided it must have at least the description translation in Estonian.	
modified	no	date-time	When this role definition was last modified. It is highly recommended to include this value in the response.
canSubDelegate	no	boolean	Can this role be given out with the right to sub-delegate it further
representeeType	no	enum [NATURAL_PERSON, LEGAL_PERSON, GOVERNMENT_PERSON]	GOVERNMENT_PERSON is a sub-type of LEGAL_PERSON whose Estonian registry code starts with 7. If representeeType is missing from the role definition then it cannot be assigned from Pääsuke.
delegateType	no	enum [LEGAL_PERSON, NATURAL_PERSON]	Type of persons this role can be assigned to. If delegateType is missing from the role definition then it cannot be assigned from Pääsuke.
assignableBy	no	list of strings	List of role combinations that are allowed to assign this mandate. The person who wants to add this role needs to have at least one of listed oles. If the value is empty then this role cannot be assigned from Pääsuke. **Assigning a role through MISP2 portal.** Pääsuke allows adding roles through MISP2 portal (this is a different portal than eesti.ee). If a role must be assignable from there then assignableBy must include "MISP2:MISP2_USER".
deletableBy	no	list of string	List of namespace:role combinations who are allowed to remove this mandate. If this is null then assignableBy roles are used instead.
deletableByDelegate	yes	boolean	Is the delegate allowed to remove the mandate given to himself/herself? Normally this is allowed for each role.

1.6.3.2 Role code

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

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

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.

1.6.4 MandateTriplet

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

Property	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	nullable	array Mandate (see paragraph 1.6.5)	List of mandates that the delegate has for this representee

1.6.5 Mandate

Property	Mandatory	Type	

namespace	yes	namespace code (see paragraph 1.6.2)	
role	yes	role code (see paragraph 1.6.3)	
validityPeriod	no	ValidityPeriod (see paragraph 1.7)	
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 removeMandate (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": "/v1/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 to the user.

1.6.6.4 uiExternalEdit

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

1.6.7 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

1.6.8 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

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

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

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

```
"authorizations": [
  {
    "userIdentifier": "EE49028099999",
    "hasRole": "FROM_BUSINESS_REGISTRY:FULL:JUHL"
  }
]
```

1.6.8.3 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": "FROM_BUSINESS_REGISTRY:PARTIAL:JUHL"
  },
  {
    "userIdentifier": "EE39211110000",
    "hasRole": "FROM_BUSINESS_REGISTRY:PARTIAL:JUHL"
  }
]
```

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

1.6.9 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).

1.6.10 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
detail	no	Translation (chapter see 1.6.4)	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

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 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.

2.1.2 X-road headers

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

- X-Road-User-Id - 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 remove 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/paasuhalduse-x-tee-teenused/0.5.1#/Offered%20to%20P%C3%A4%C3%A4suke/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
      ?ns=STAT,GLOBAL1_STAT"
```

The return type is the array of RoleDefinition (described in chapter 1.6.3).

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 use namespace codes and role codes in the mandate payload. For the Pääsuke UI to translate these codes into different languages

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.
- 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 Filtering by namespace

Pääsuke keeps track of what kind of namespaces are used by different service providers.

The list of namespaces currently configured is included in the "ns" filter when Pääsuke makes the request.

Even if the service provider returns roles with other namespaces Pääsuke ignores such roles in the output.

2.3 Query "getRepresenteeDelegatesWithMandates"

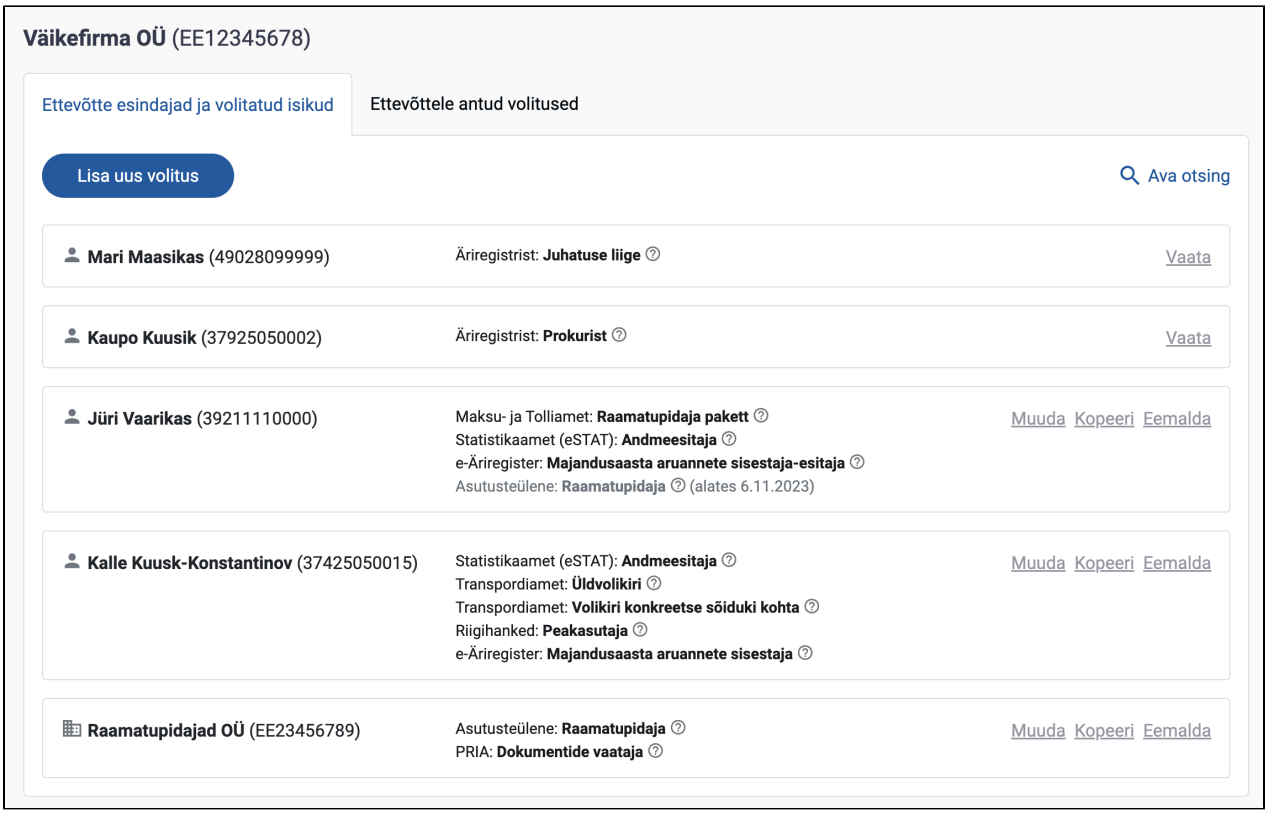
CURL query

```
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 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:

The screenshot describes the following situation. Väikefirma OÜ (EE12345678) has given roles GLOBAL1: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
      ?subDelegatedBy=EE23456789"
```

And 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).

2.3.3 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

```
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.

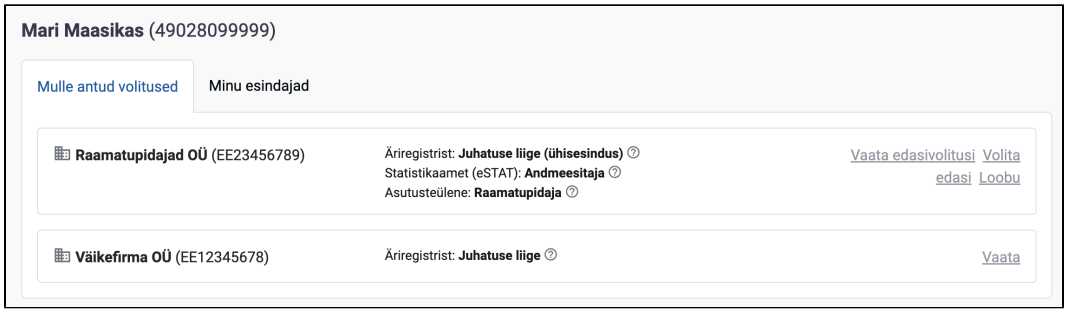
2.4.1 View "Ettevõttele antud volitused"

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

When the user clicks "Vaata edasivolitusi" then it opens up a view "Ettevõtte <ettevõttenimi> poolt antud volituste edasivolitused" described above.

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 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 link allowing to add a 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 a list of MandateTriplets (described in chapter 1.6.4)

2.6 Query removeMandate

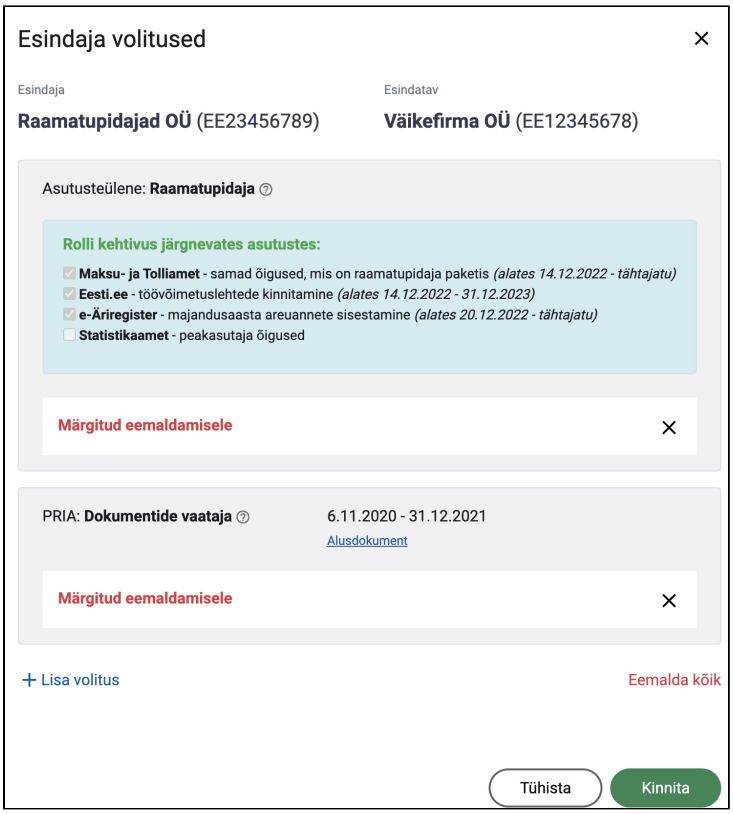
```
curl -X DELETE \
     "https://security-server/r1/ee-dev/GOV/70006317/volitused-mock/volitused-estat/v1/ns/{ns}/representees/
{representeeIdentifier}/delegates/{delegateIdentifier}/mandates/{mandateIdentifier}"
```

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

Values of ns, 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.



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

2.7 Query addMandateSubDelegate

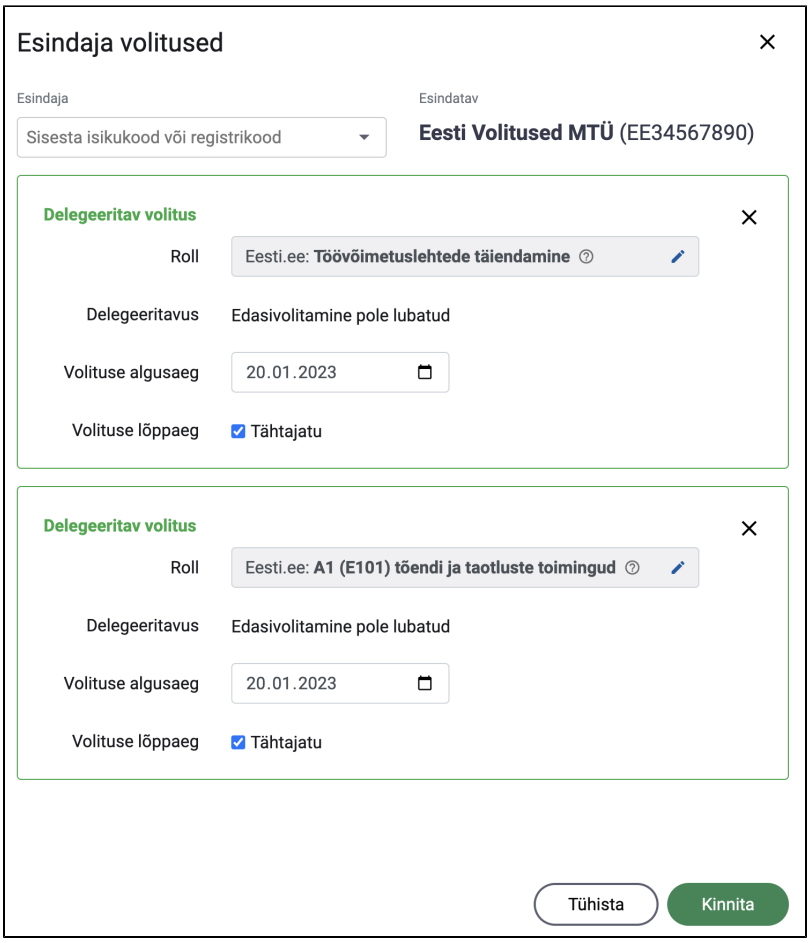
2.7.1 View where adding a sub-delegate can be started

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

These use cases are described in chapters 2.3.2 and 2.4.2.

If at least one of the mandates has a link xRoadPostSubDelegate (described in chapter 1.6.6.3) then the UI adds a button to initiate adding a sub-delegate ("Volita edasi")

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



The user is allowed to:

- set of the delegate as long as it is natural person
- edit the start date as long as it is later than the original start date.
- edit the end date as long as it is earlier than the original end date.
- delete a role from the list - this means only a portion of roles get sub-delegated, not all

The user is restricted from:

- changing role
- adding a new mandate on this screen
- setting a start date to the past
- setting the mandate to be allowed for sub-delegation
- setting the end date to be today or earlier

2.8.2 Path parameters

The parameters of the payload are set by the response that loaded the mandates. The parameters are described in paragraph 1.6.6.3.

2.8.3 Payload

- subDelegate is of type Person (described in chapter 1.6.1)
- validityPeriod from is only present if it was changed by the user
- 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

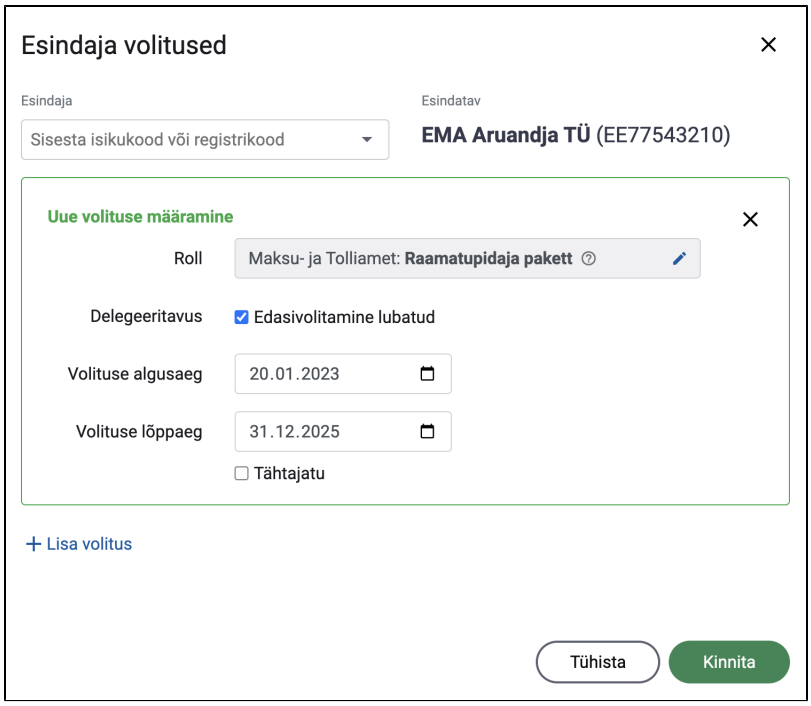
Query to be sent out by Pääsuke

```
{
  "subDelegate": {
    "type": "NATURAL_PERSON",
    "firstName": "Jüri",
    "surname": "Juurikas",
    "legalName": "Juurikas",
    "identifier": "EE38302250123"
  },
  "validityPeriod": {
    "from": "2017-07-21",
    "through": "2024-02-21"
  }
}
```

2.8 Query addMandatesToDelegate

2.8.1 View to add mandates to a delegate

This is how a new mandate can be added:



2.8.2 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> '
```

2.8.2 Payload

Query to be sent out by Pääsuke

```
{
  "representee": {
    Person to be represented
  },
  "delegate": {
    Person getting the representation rights
  },
  "mandates": [
    {
      "namespace": "GLOBAL1_EMTA",
      "role": "GLOBAL1_EMTA:ACCOUNTANT",
      "canSubDelegate": true,
      "validityPeriod": {
        "from": "2017-07-21",
        "through": "2024-02-21"
      }
    }
  ],
  "authorizations": [
    {
      "userIdentifier": "string",
      "hasRoles": "MANAGEMENT_BOARD_MEMBER"
    }
  ]
}
```

Authorizations are used to show the information about who has confirmed that change and on what grounds.

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

This use case is going to be described in another document.

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

This use case is going to be described in another document.

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

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 representee.

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