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.

1.2 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 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 representedPerson) - person (most commonly a private limited company) that can be represented in e-services
- **delegate** - person who have been given a mandate to represent some representee. Delegate normally always has the right to represent oneself (except if the person doesn't posess active legal capacity - piiratud teovõime)
- **namespace** - group of roles that are maintained by a single provider. Read more from chapter 1.2.1
- **role** - group of privileges to be used in a e-service that can be granted to delegate by representee. Role always belongs to a namespace. Read more from chapter 1.2.2
- **mandate** - role that is given to a delegate under some representee. Mandate can have start date, end date and some mandates can be sub-delegated (edasi delegeerima).

1.2.1 What is a namespace

In this document names of namespaces are written in capital letters. If namespace is written in pair with a role then we use 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 (nn katusrollide funktsionaalsus).

- STANDALONE namespace (regular namespace)
    - has subtype: AUTOMATIC namespace (roles in automatic namespaces are automatically assigned based on external property, for example a role in such namespace could be assigned automatically to a person after becoming management board member)
- PARENT namespace (to group together global roles ("katusrollid"), in programming terms each role in parent namespace is like an interface)
- CHILD namespace (in programming terms each role in child namespace is like an implementation of a global role and it is allowed to have more than one implementation of the same role)

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 easisest 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 EMTA currently has single namespace "EMTA".
- For one institution it is also possible to have multiple namespaces.
    - This is only recommended for existing applications and not for new ones.
    - For example Statistikaamet is going to have separate namespaces for two separate systems: "STAT-ESTAT" and "STAT-SSO".

Parent-child namespaces are described in chapter 1.2.3.

1.2.2 What is a role

Roles usually have descriptive names that indicate what the owner can do (VIEWER can view).

Pääsuke is not limiting what 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 responsiblility of the Agency X to:

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

Properties of a role:

- role belongs to a namespace
- role has a name (unique identifier, not shown out) that is unique in that namespace
- role has a title (in Estonian, English and Russian)
- role can have a description (in Estonian, English and Russian)
- last modification date-time.

1.2.3 Namespaces with parent-child relation

RIA wants to define global roles (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 this using an example. RIA creates a parent namespace (by setting type=PARENT, parent_namespace=null) with name GLOBAL1 that contain following roles:

- Bookkeeper
- Human_Resources_Specialist
- Data_viewer

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

Any institution can decide to support some of these roles. Let's say some AgencyA already has added support for all 3 roles.

We have an institution AgencyX that want's to allow using its self-service portal for holders of global roles Human_Resources_Specialist and Data_viewer.

This AgencyX must register a new namespace, let's call it AGENCYX_GLOBAL1 and set that it is a child of GLOBAL1 namespace (by setting parent_namespace="GLOBAL1", type=CHILD).

Child namespaces can only declare roles with names that are declared in the parent namespace, but child namespaces can skip some roles that are declared in parent namespace.

AgencyX declares the roles "Human_Resources_Specialist" and "Data_viewer" in namespace AGENCYX_GLOBAL1.

Pääsuke detects that Agency X now supports these roles and when some representee gives mandates to some delegate then under namespace GLOBAL1 then the representee can now tick Agency X (besides AgencyA) for roles that are supported by that AgencyX:

Namespace: GLOBAL1	
Role:	Bookkeeper
Valid in agencies:	(tickbox) AgencyA
Role:	Human_Resources_Specialist
Valid in agencies:	(tickbox) AgencyA
	(tickbox) AgencyX
Role:	Data_viewer
Valid in agencies:	(tickbox) AgencyA
	(tickbox) AgencyX

If some employee opens namespace GLOBAL1 and adds role "Human_Resources_Specialist" and ticks only that the role is valid in AgencyX then behind the scenes the delegate gets role AGENCYX_GLOBAL1:Human_Resources_Specialist.

1.3 Notes

1.3.1 Playing with the mock services over X-Road

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

You can send requests against that services 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.

1.4 Different types of X-Road services in Pääsuke

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

- X-Road services that are consumed by Pääsuke itself. There are two types:
    - "standardardized services" that 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 services 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)

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

While describing the services a screenshot has been added of the user interface where the data will end up to in Pääsuke.

2.1. Clarifications

2.1.1 Historyc data is not returned

The services only return mandates that are currently valid or will become valid in the future. The history is 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?namespace=STAT"
```

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

Role name can contain following symbols: "A-Za-z0-9_.". Role name must be unique in the namespace.

Role name is not displayed to the user, it is 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 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).

2.3 Query "getRepresenteeDelegatesWithMandates"

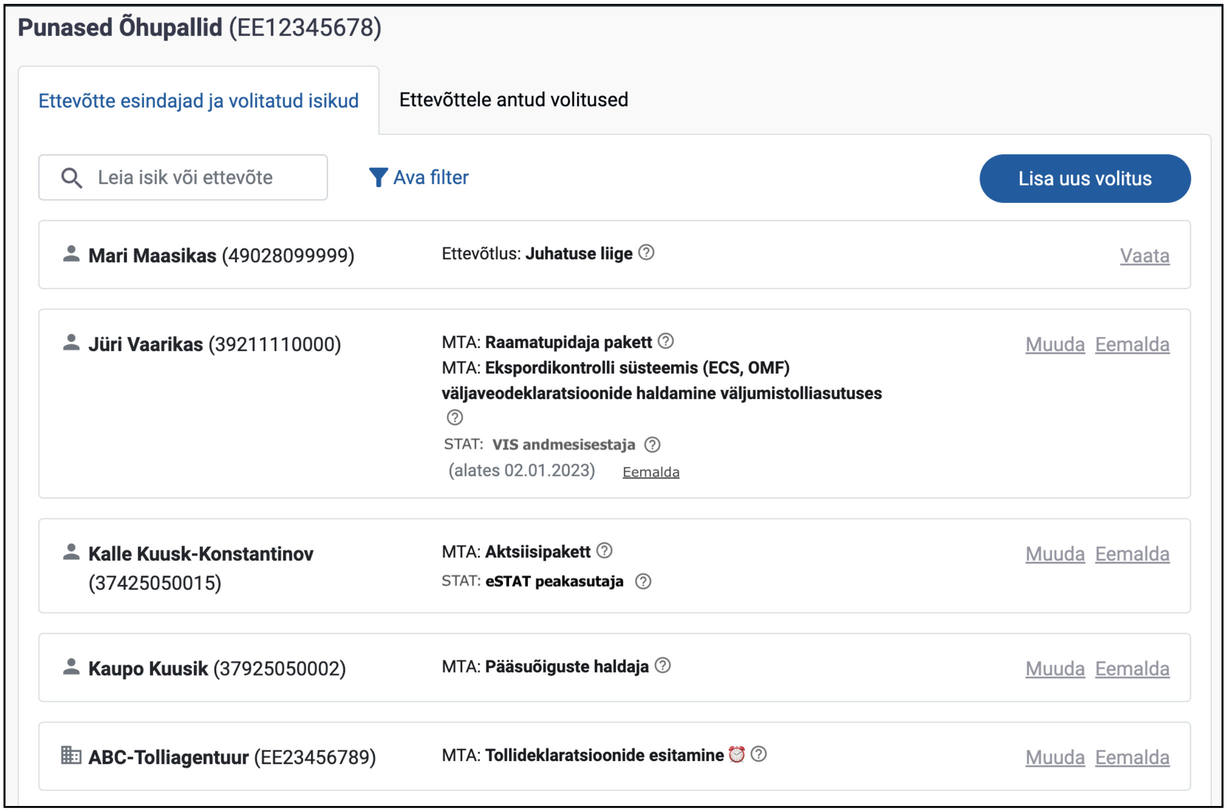
CURL query

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

Returns all persons, 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.



Punased Õhupallid (EE12345678)

Ettevõtte esindajad ja volitatud isikud | Ettevõttele antud volitused

Mari Maasikas (49028099999) — Ettevõtlus: **Juhatuse liige** — Vaata

Jüri Vaarikas (39211110000) — MTA: **Raamatupidaja pakett**
MTA: **Ekspordikontrolli süsteemis (ECS, OMF) väljaveodeklaratsioonide haldamine väljumistolliasutuses**
STAT: **VIS andmesisestaja** (alates 02.01.2023) Eemalda — Muuda Eemalda

Kalle Kuusk-Konstantinov (37425050015) — MTA: **Aktsiisipakett**
STAT: **eSTAT peakasutaja** — Muuda Eemalda

Kaupo Kuusik (37925050002) — MTA: **Pääsuõiguste haldaja** — Muuda Eemalda

ABC-Tolliagentuur (EE23456789) — MTA: **Tollideklaratsioonide esitamine** ⏰ — Muuda Eemalda

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

In the future Pääsuke will have a view to see list of subdelegators.

Imagine following situation. Väikefirma OÜ (EE1111111) has given a role BOOKKEEPER (from the GLOBAL namespace) to Raamatupidajad OÜ (EE22222222) with right to sub-delegate.

Now Raamatupidajad OÜ has sub-delegated this role to its employees Reijo Raamatukogu and Raili Raamatukoi.

Now a representator of Raamatupidajad OÜ clicks on "Väikefirma OÜ" in view "Ettevõttele antud volitused" and sees list with two employees.

UI makes following query:

CURL query

```
curl -H "namespace: EMTA" \
     -H "subDelegatedBy: EE22222222" \
     -X GET \
     "https://security-server/r1/ee-dev/GOV/70006317/volitused-mock/volitused-estat/v1/representees/EE1111111
/delegates/mandates"
```

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 we have subDelegatedBy filter parameter in place).

2.3.3 View "Minu esindajad"

Kaugemas tulevikus saab eraisik Pääsukesest vaadata, kellele ta on andnud volituse ennast esindada ja millistes rollides

2.4 Query getDelegateRepresenteesWithMandates

GET

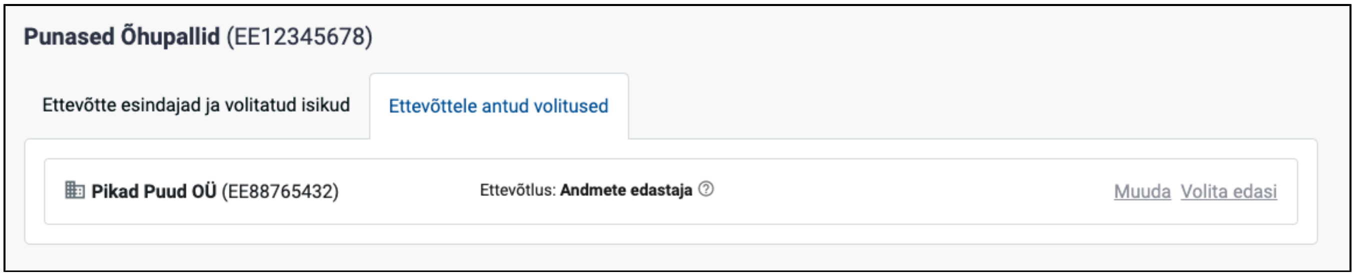
```
curl  -X GET \
     "https://security-server/r1/ee-dev/GOV/70006317/volitused-mock/volitused-estat/v1/delegates/{delegate}
/representees/mandates"
```

Returns all persons that the delegate has 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)

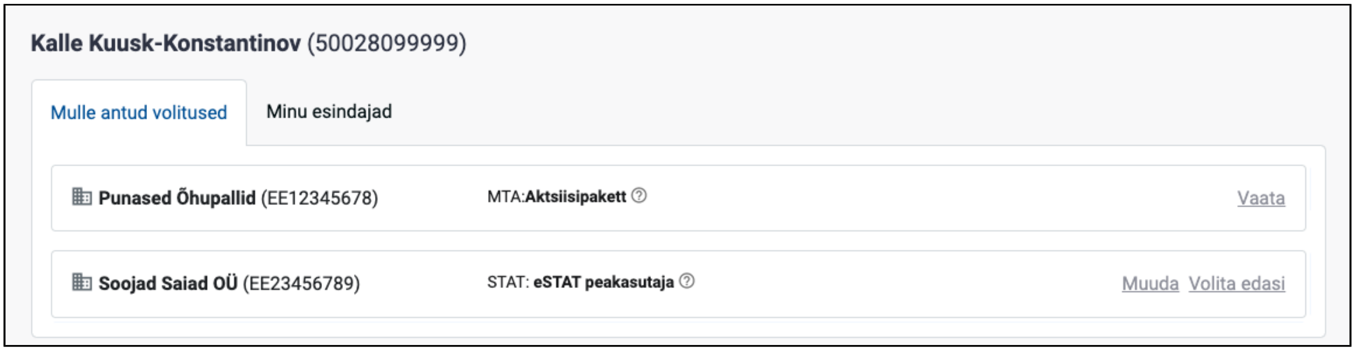
Ettevõtte esindajad ja volitatud isikud
Ettevõttele antud volitused

Pikad Puud OÜ (EE88765432) Ettevõtlus: **Andmete edastaja**

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 volitusted"

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



Kalle Kuusk-Konstantinov (50028099999)

Punased Õhupallid (EE12345678) MTA:**Aktsiisipakett**

Soojad Saiad OÜ (EE23456789) STAT: **eSTAT peakasutaja**

2.5 Query removeMandateFromDelegate

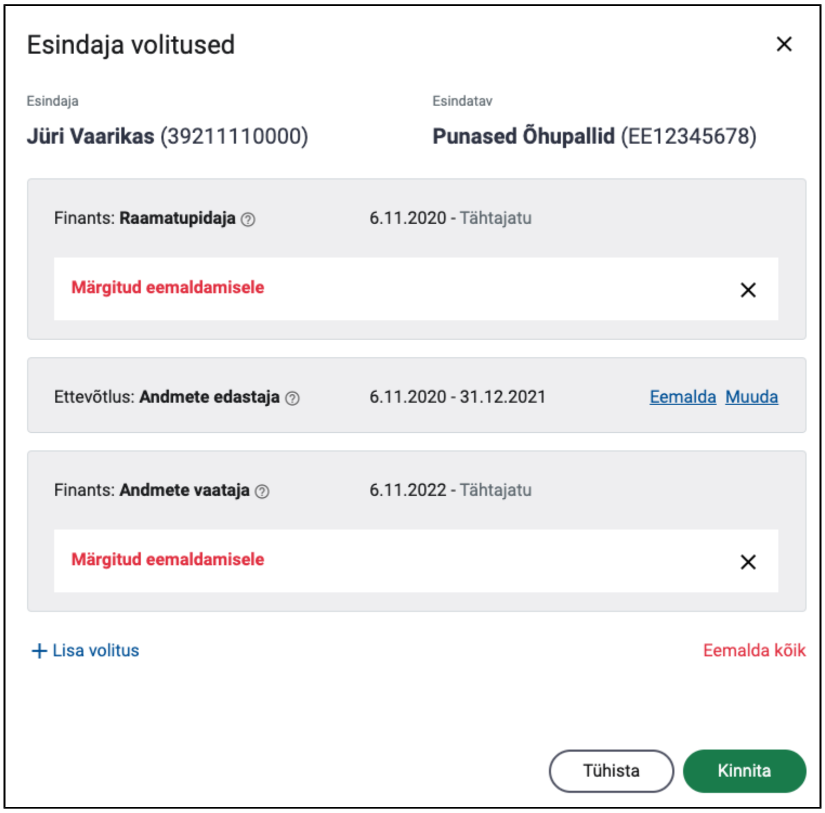
```
curl  -X DELETE \
     "https://security-server/r1/ee-dev/GOV/70006317/volitused-mock/volitused-estat/v1/representees/
{representee}/delegates/{delegate}/mandates/{mandateIdentifier}
     ?namespace={namespace}"
```

Removes role from delegate in given namespace.

Values of representee, delegate ja role are taken by Pääsuke from the output of the companies (link with rel "xroadDelete").

2.5.1 View to remove mandates from delegate

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



Esindaja volitused

Esindaja
Jüri Vaarikas (39211110000)

Esindatav
Punased Õhupallid (EE12345678)

Finants: **Raamatupidaja** 6.11.2020 - Tähtajatu

Märgitud eemaldamisele

Ettevõtlus: **Andmete edastaja** 6.11.2020 - 31.12.2021

Finants: **Andmete vaataja** 6.11.2022 - Tähtajatu

Märgitud eemaldamisele

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

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 INDIVIDUAL (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/namespaces"
```

Get available namespace configuration.

3.3.1 Output

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

```
[
  {
    "namespace": "GLOBAL1",
    "parent_namespace": null,
    "title": [
      {
        "lang": "et",
        "value": "Asutusteülesed katusrollid"
      }
    ],
    "type": "PARENT"
  },
  {
    "namespace": "LHV_GLOBAL1",
    "parent_namespace": "GLOBAL1",
    "title": [
      {
        "lang": "et",
        "value": "LHV pank"
      }
    ],
    "type": "CHILD",
  },
  {
    "namespace": "AUTOM_BUSINESSREG",
    "parent_namespace": null,
    "title": [
      {
        "lang": "et",
        "value": "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'
```

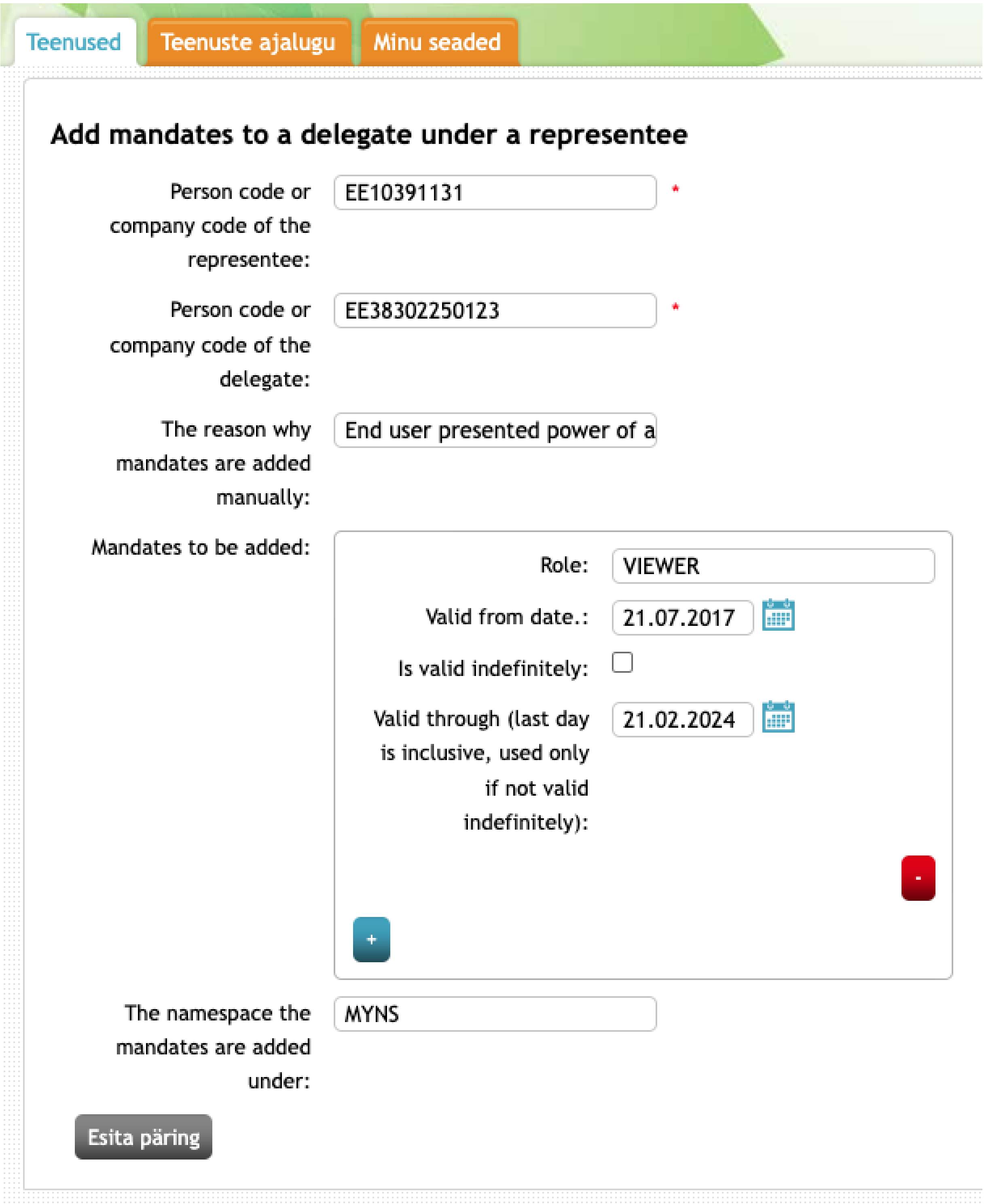
With following payload:

```
{
  "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

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.