Introduction to NHR PoRS2

07.11.2019

Content

Revision History

Version	Dato	Oversigt over rettelser	Ændret af
2-Beta1	04.03.2014	Created	Peter Sone Koldkjær (mySupply)
2-Beta2	02.04.2014	Fully translated to English	Peter Sone Koldkjær (mySupply)
2-Beta3	14.05.2014	Data model description updated, UpdateBusiness example added and a visual representation added.	Peter Sone Koldkjær (mySupply)
2-Beta4	24.06.2014	DateTime now follows ISO8601 syntax – examples updated.	Peter Sone Koldkjær (mySupply)
2-Beta5	13.08.2014	Lots of minor adjustments related to consistency between XSD and implementation (casing and such)	Peter Sone Koldkjær (mySupply)
2-Beta6	25.08.2014	XSD brush-up incl. correct type to int for ID fields I XSD's, sequence on XSD's and lists wrapped in list element.	Peter Sone Koldkjær (mySupply)
2-Beta7	04.09.2014	XSD now has namespaces and other smaller fixes.	Peter Sone Koldkjær (mySupply)
2-Beta8	10.09.2014	Participants removed from Business output, Participant: Sort by UnitName added and minor updates.	Peter Sone Koldkjær (mySupply)
2.0.0	29.09.2014	Authorization control updated and JSON output fixed.	Peter Sone Koldkjær (mySupply)
2.1.0	01.05.2016	Data model updated to support PEPPOL data for NHR being a Danish national SMP.	Peter Sone Koldkjær (mySupply)
2.1.0	09.10.2019	Add description for PEPPOL AS2 and AS4 services in NHR,	Peter Sone Koldkjær (mySupply)

1 Introduction

This introduction contains guidance regarding the PoRS2 interface in the Danish eProcurement Register (NemHandelsRegister: NHR). This guide is targeted for software professionals.

PoRS2 (Profile Registration Service version 2) is an API for NHR, making lookup and changes possible.

1.1 Support

For questions or other support related issues, please contact NemHandel Support: support@nemhandel.dk

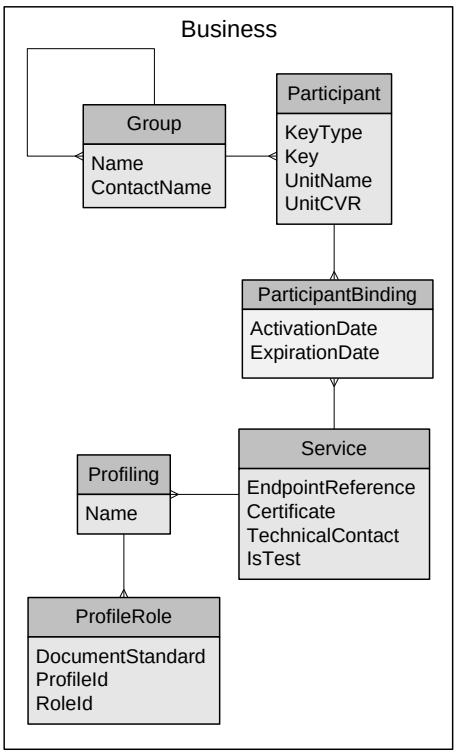
2 Access / technical setup

PoRS2 is a REST-interface, protected by 2-way SSL authentication. An OCES Employee certificate (MOCES) or Function certificate (FOCES) must be used as client certificate (see https://www.nets.eu/dk-da/l%C3%B8sninger/nemid/medarbejdersignatur regarding ordering of such certificates). The certificate identifies the organization (CVR-number) that the client will represent (regarding ownership of the data changed in NHR).
NHR Test environment must be used for functional testing of PoRS2 client software (see more regarding NHR Test at https://digitaliser.dk/resource/2558550).

URLs for PoRS2 will be listed in the group https://digitaliser.dk/group/56190, and changes to the interface will also be announced in this group.

3 Data model

The data model used in the PoRS2 interface is a 1-1 mapping of the data model used internally in NHR.
To the right is a simple, visual representation of the data model.
The main idea is to have a smaller, almost static set of Profilings and Services, just adding Participants and Groups, and changing the referenced Service on existing Participant if required.



Below is a technical object based documentation, and in the online NHR Web documentation a visual walk-through can be found: http://www.nemhandel.dk/media/2570/nemhandel registret%20-%20hj%C3%A6lpeguide.pdf

Detailed documentation like fields, XSD, status codes, etc. is located online as part of the REST-site: https://registrationservice-test.nemhandel.dk/nemhandel-pors/rest/
Code files can be generated from XSD files. Search the internet for guides related to your favorite programming language.

3.1 Implementation

All DateTime fields will be handled as ISO 8601 using the syntax: yyyy-MM-ddTHH:mm:ssZ (like 2014-06-18T09:56:21+00:00).
All fields: If data is not present or has the value null, the specific XML element is omitted from the output.

3.2 Business

A Business is a representation of the business/organization/company owning data in NHR.
All data objects in NHR are linked to a specific Business identified by their certificate. PUT + DELETE requests can only be performed on data that belongs to that specific Business identified by their certificate.

3.3 Participant

The Participant object represent a single recipient in NHR, identified by a KeyType (GLN/EAN, CVR, IBAN, ..) and an identifier (number/value).
Each Participant must have at least one ParticipantBinding. The ParticipantBinding represents a period and refers to a Service (by ID). The period can be infinite, by setting activation date and expire date to an empty value.
By using expire and activation dates actively, it is possible to "schedule" a change of the supported Service (and thereby Profiling, see below). Setting expireDate is also the approach to use to avoid overlap/duplicates, when a Participant must change ownership from one Business to another.

3.4 Service

In NHR the Service concept covers 2 types of Service.

3.4.1 Receiver Service

A Receiver Service represents the technical endpoint (URL, certificate and contact data) and the Profiling supported.

3.4.2 Aggregated Service

An Aggregated Service is a special type of Service, mostly used by the Danish government agencies. An Aggregated Service refers/consists of a number of Receiver Service – with the criteria, that the underlying document profiles (from Profiling) cannot overlap.
This way an Invoice can be routed to one endpoint, and Orders to another for the same Participant.

3.5 Profiling

A profiling is a package/set of document profiles (e.g. OIOUBL profiles), each called ProfileRole. A ProfileRole is described by the DocumentStandard, Profile and a role (buyer or seller).

3.6 Group

The purpose of Group, are grouping of Participants, to ease the administration of large numbers of Participants for a single Business. Groups can be arranged into a structure, where a Group can be owned by another group (ownerGroup).
Note: The Group object was in OpenUDDI (and in the transition to NHR) used as "naming tool" for a Participant. Later UnitName has been introduced, and the Group concept is now back to its original purpose: Administrative grouping of Participants.

4 Examples

The examples below have been executed against NHR Test: https://registrationservice-test.nemhandel.dk/nemhandel-pors/rest/

For all requests, the following HTTP header has been used: `Content-Type: text/xml`

4.1 Static data / metadata

A number of services can return the static metadata from NHR (see https://registrationservice-test.nemhandel.dk/nemhandel-pors/rest-documentation.html for documentation).
The service https://registrationservice-test.nemhandel.dk/nemhandel-pors/rest/list/keytypes provides (right now) the following XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<KeyTypeList xmlns="http://registrationservice.nemhandel.dk/2016/05/01/pors">
    <KeyType>
        <Id>1</Id>
        <Iso6523Code>0088</Iso6523Code>
        <PeppolSchemeId>GLN</PeppolSchemeId>
    </KeyType>
    <KeyType>
        <Id>2</Id>
        <Iso6523Code>0062</Iso6523Code>
        <PeppolSchemeId>DUNS</PeppolSchemeId>
    </KeyType>
    <KeyType>
        <Id>3</Id>
        <Iso6523Code>0021</Iso6523Code>
        <PeppolSchemeId>IBAN</PeppolSchemeId>
    </KeyType>
    <KeyType>
        <Id>4</Id>
        <Iso6523Code>0028</Iso6523Code>
        <PeppolSchemeId>ISO6523</PeppolSchemeId>
    </KeyType>
    <KeyType>
        <Id>5</Id>
        <Iso6523Code>9901</Iso6523Code>
        <PeppolSchemeId>DK:CPR</PeppolSchemeId>
    </KeyType>
    <KeyType>
        <Id>6</Id>
        <Iso6523Code>9902</Iso6523Code>
        <PeppolSchemeId>DK:CVR</PeppolSchemeId>
    </KeyType>
    <KeyType>
        <Id>7</Id>
        <Iso6523Code>0096</Iso6523Code>
        <PeppolSchemeId>DK:P</PeppolSchemeId>
    </KeyType>
    <KeyType>
        <Id>8</Id>
        <Iso6523Code>9904</Iso6523Code>
        <PeppolSchemeId>DK:SE</PeppolSchemeId>
    </KeyType>
    <KeyType>
        <Id>9</Id>
        <Iso6523Code>9905</Iso6523Code>
        <PeppolSchemeId>DK:VANS</PeppolSchemeId>
    </KeyType>
</KeyTypeList>
```

These ID values must be used when client data are generated.

4.2 Get Business

The service https://registrationservice-test.nemhandel.dk/nemhandel-pors/rest/business provides the Business corresponding to the CVR-number given by the client certificate:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Business xmlns="http://registrationservice.nemhandel.dk/2016/05/01/pors">
    <Id>492</Id>
    <LastModified>2014-08-20T15:11:16+02:00</LastModified>
    <Name>Digitaliseringsstyrelsen</Name>
    <ContactName>Peter Sone Koldkjær</ContactName>
    <ContactEmail>support@nemhandel.dk</ContactEmail>
    <Key>34051178</Key>
    <WebsiteUrl>http://www.digst.dk</WebsiteUrl>
</Business>
```

This XML output contains the Business object.

4.3 Update Business

An existing Business can be updated, by sending a PUT request to https://registrationservice-test.nemhandel.dk/nemhandel-pors/rest/business

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Business>
    <Id>492</Id>
    <Name>Digitaliseringsstyrelsen</Name>
    <ContactName>Peter Sone Koldkjær</ContactName>
    <ContactEmail>support@nemhandel.dk</ContactEmail>
    <Key>34051178</Key>
    <WebsiteUrl>http://www.digst.dk</WebsiteUrl>
</Business>
```

4.4 Create Profiling

In this example we will create a new Profiling, being a NemHandel Profiling, setup to receive the OIOUBL profiles Procurement-BilSim-1.0 and Reference-Utility-1.0.

To get ID's for these 2 profiles, first find the ID of the OIOUBL-2.02 standard at https://registrationservice-test.nemhandel.dk/nemhandel-pors/rest/list/documentstandards The ID to use is 2. Then use this ID to get the OIOUBL-2.02 profiles at https://registrationservice-test.nemhandel.dk/nemhandel-pors/rest/list/2/profiles
The two ID's to use are 3 and 30.
Since we want to receive, we have to find the Customer roleID's for the 2 profiles from https://registrationservice-test.nemhandel.dk/nemhandel-pors/rest/list/2/profile/3/roles and https://registrationservice-test.nemhandel.dk/nemhandel-pors/rest/list/2/profile/30/roles - both being ID=1.
Based on the data above, we can now construct the UpdateProfiling XML to PUT to https://registrationservice-test.nemhandel.dk/nemhandel-pors/rest/profiling

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<UpdateProfiling>
    <Id>0</Id>
    <OwnerBusinessKey>34051178</OwnerBusinessKey>
    <NetworkTypeId>1</NetworkTypeId>
```

```
    <Name>Demo-profiling</Name>
    <ProfileRoles>
        <ProfileRole>
            <Id>0</Id>
            <DocumentStandardId>2</DocumentStandardId>
            <ProfileId>3</ProfileId>
            <RoleId>1</RoleId>
        </ProfileRole>
        <ProfileRole>
            <Id>0</Id>
            <DocumentStandardId>2</DocumentStandardId>
            <ProfileId>30</ProfileId>
            <RoleId>1</RoleId>
        </ProfileRole>
    </ProfileRoles>
</UpdateProfiling>
```

Based on the ID=0, a new Profiling object is created and PoRS2 now respond with this HTTP response:

```
Status Code   : 201 Created
Cache-Control : no-cache
Content-Length: 0
Content-Type  : text/xml
Date          : Tue, 18 Mar 2014 10:51:29 GMT
Expires       : Thu, 01 Jan 1970 01:00:00 CET
Location      : https://registrationservice-test.nemhandel.dk/nemhandel-
pors/rest/profiling/85444
Pragma        : No-cache
Server        : Apache-Coyote/1.1
```

The new object is created and assigned ID=85444 (see Location header).

4.5 Create ReceiverService

In this example we will create a new ReceiverService, of type NemHandel, using the Profiling from above (ID= 85444).

Based on these data, we can now construct the UpdateReceiverService XML to PUT to https://registrationservice-test.nemhandel.dk/nemhandel-pors/rest/service

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<UpdateReceiverService>
    <Id>0</Id>
    <OwnerBusinessKey>34051178</OwnerBusinessKey>
    <NetworkTypeId>1</NetworkTypeId>
<EndpointReference>http://raspservice.nemhandel.gov.dk/RASP/1.2/reciever.svc</E
ndpointReference>
    <Certificate>OID.2.5.4.5=CVR:34051178-FID:1359538750995 + CN=DIGST Demo
endpoint (funktionscertifikat), O=Digitaliseringsstyrelsen // CVR:34051178,
C=DK</Certificate>
    <ProfilingId>46382</ProfilingId>
    <Alias>Demo-receiverService</Alias>
    <IsTest>false</IsTest>
    <ContactName>NemHandel Support</ContactName>
    <ContactEmail>suport@nemhandel.dk</ContactEmail>
</UpdateReceiverService>
```

Based on the ID=0, a new ReceiverService object is created and PoRS2 now respond with this HTTP response:

```
Status Code   : 201 Created
```

```
Cache-Control : no-cache
Content-Length: 0
Content-Type  : text/xml
Date          : Tue, 18 Mar 2014 13:47:27 GMT
Expires       : Thu, 01 Jan 1970 01:00:00 CET
Location      : https://registrationservice-test.nemhandel.dk/nemhandel-pors
/rest/service/84140
Pragma        : No-cache
Server        : Apache-Coyote/1.1
```

The new object is created and assigned ID=84140 (see Location header).

4.6 Create Group

In this example we will create a new Group.
Data model notes:

- OwnerGroupId is optional, and is the ID of the parent group to this new group.

PUT the UpdateGroup XML to https://registrationservice-test.nemhandel.dk/nemhandel-pors/rest/group

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<UpdateGroup>
    <Id>0</Id>
    <OwnerBusinessKey>34051178</OwnerBusinessKey>
    <Name>New group name</Name>
    <Description>Group description</Description>
    <ContactName>ContactName</ContactName>
    <ContactEmail>support@nemhandel.dk</ContactEmail>
    <OwnerGroupId>0</OwnerGroupId>
</UpdateGroup>
```

Based on the ID=0, a new Group object is created.

4.7 Read Participant

Based on the list above, we would like to extract all data for GLN 5798009811578:
https://registrationservice-test.nemhandel.dk/nemhandel-pors/rest/participant/GLN/5798009811578

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Participant xmlns="http://registrationservice.nemhandel.dk/2016/05/01/pors">
    <Id>45115</Id>
    <LastModified>2014-08-21T15:52:48+02:00</LastModified>
    <OwnerBusinessId>492</OwnerBusinessId>
    <Key>5798009811578</Key>
    <KeyType>GLN</KeyType>
    <UnitName>DIGST NemHandel Demo-endepunkt</UnitName>
    <UnitCVR>00000000</UnitCVR>
    <Group>
        <Id>42289</Id>
        <Name>DIGST NemHandel Demo-endepunkter</Name>
        <Description>Demo-endepunkter kan benyttes til at teste/verificere den
tekniske udveksling af NemHandel dokumenter.</Description>
        <ContactName>NemHandel Support</ContactName>
        <ContactEmail>support@nemhandel.dk</ContactEmail>
        <OwnerGroupId>0</OwnerGroupId>
        <OwnerBusinessId>0</OwnerBusinessId>
```

```
    </Group>
    <ParticipantBindings>
        <ParticipantBinding>
            <Id>45790</Id>
            <LastModified>2014-08-21T15:52:48+02:00</LastModified>
            <NetworkTypeId>1</NetworkTypeId>
            <OwnerServiceId>46972</OwnerServiceId>
            <OwnerService xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="ReceiverService">
                <Id>46972</Id>
                <LastModified>2014-08-21T15:52:48+02:00</LastModified>
                <NetworkTypeId>1</NetworkTypeId>
                <Name>Demo-endepunkt</Name>
                <DisplayName>Demo-endepunkt</DisplayName>
                <OwnerBusinessId>0</OwnerBusinessId>
                <CanConsolidate>false</CanConsolidate>

<EndpointReference>http://raspservice.nemhandel.gov.dk/RASP/1.2/reciever.svc</E
ndpointReference>
                <Certificate>OID.2.5.4.5=CVR:34051178-FID:1359538750995 +
CN=DIGST Demo endpoint (funktionscertifikat),
                    O=Digitaliseringsstyrelsen // CVR:34051178, C=DK
                </Certificate>
                <IsTest>false</IsTest>
                <ContactName>NemHandel Support</ContactName>
                <ContactEmail>support@nemhandel.dk</ContactEmail>
                <ProfilingId>46382</ProfilingId>
            </OwnerService>
        </ParticipantBinding>
    </ParticipantBindings>
</Participant>
```

This XML output contains the Participant and all belonging ParticipantBindings.
Given the Participant.ID=45115 from the output, the same output can be found using https://registrationservice-test.nemhandel.dk/nemhandel-pors/rest/participant/45115

4.8 Update Participant

Based on the data above, we would like to update data for GLN 5798009811578.
PUT request must be done to: https://registrationservice-test.nemhandel.dk/nemhandel-pors/rest/participant

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<UpdateParticipant>
    <Id>45115</Id>
    <OwnerBusinessKey>34051178</OwnerBusinessKey>
    <Key>5798009811578</Key>
    <KeyType>GLN</KeyType>
    <UnitName>DIGST NemHandel Demo-endepunkt</UnitName>
    <UnitCVR>00000000</UnitCVR>
    <GroupId>42289</GroupId>
    <Bindings>
        <Binding>
            <Id>45790</Id>
            <NetworkTypeId>1</NetworkTypeId>
            <ActivationDate>2014-01-01T00:00:00+01:00</ActivationDate>
            <ExpirationDate></ExpirationDate>
            <ServiceId>46972</ServiceId>
        </Binding>
    </Bindings>
</UpdateParticipant>
```

Please note the format for Date Time. Using blank Date fields corresponds to null, meaning no start and no end.

4.9 Create Participant

Based on the data above, we would like to create Participant with GLN 5798009800001.
PUT request must be done to: https://registrationservice-test.nemhandel.dk/nemhandel-pors/rest/participant

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<UpdateParticipant>
    <Id>0</Id>
    <OwnerBusinessKey>34051178</OwnerBusinessKey>
    <Key>5798009800001</Key>
    <KeyType>GLN</KeyType>
    <UnitName>PoRS2 create test</UnitName>
    <UnitCVR>00000000</UnitCVR>
    <Bindings>
        <Binding>
            <Id>0</Id>
            <NetworkTypeId>1</NetworkTypeId>
            <ActivationDate>2014-01-01T00:00:00+01:00</ActivationDate>
            <ExpirationDate></ExpirationDate>
            <ServiceId>46972</ServiceId>
        </Binding>
    </Bindings>
</UpdateParticipant>
```

Participant is created (HTTP status 201) and Location header is https://registrationservice-test.nemhandel.dk/nemhandel-pors /rest/service/80294

4.10 Delete Participant

The newly created Participant with ID=80294 is deleted by making a DELETE request https://registrationservice-test.nemhandel.dk/nemhandel-pors /rest/service/80294

HTTP status 200 OK is returned (no Response body).

4.11 Error handling

Below are some common error situations.

4.11.1 Invalid ownerBusinessKey

On PUT calls, the ownerBusinessKey must match the CVR number found in the client certificate.
Example using the UpdateParticipant XML from section 4.8, but with another ownerBusinessKey.
PUT request is sent to: https://registrationservice-test.nemhandel.dk/nemhandel-pors/rest/participant

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<UpdateParticipant>
    <Id>45115</Id>
    <OwnerBusinessKey>00001234</OwnerBusinessKey>
    <Key>5798009811578</Key>
    <KeyType>GLN</KeyType>
    <UnitName>DIGST NemHandel Demo-endepunkt</UnitName>
    <UnitCVR>00000000</UnitCVR>
    <GroupId>42289</GroupId>
    <Bindings>
        <Binding>
            <Id>45790</Id>
            <NetworkTypeId>1</NetworkTypeId>
            <ActivationDate>2014-01-01T00:00:00+01:00</ActivationDate>
            <ExpirationDate></ExpirationDate>
            <ServiceId>46972</ServiceId>
        </Binding>
    </Bindings>
</UpdateParticipant>
```

The HTTP response has status code 409, and the Response Body is:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Errors>
    <Error>
        <Code>access.unauthorized.cvr</Code>
        <ErrorType>Error</ErrorType>
        <Id>ownerBusinessKey</Id>
    </Error>
</Errors>
```

Same error code is returned upon trying to delete data not owned by you.

4.11.2 Validation errors

All data PUT to NHR are validated centrally. Below is an example, with multiple validation errors.
Example using the UpdateParticipant XML from section 4.8, but with multiple data errors.
PUT request is sent to: https://registrationservice-test.nemhandel.dk/nemhandel-pors/rest/participant

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<UpdateParticipant>
    <Id>0</Id>
```

```
    <OwnerBusinessKey>34051178</OwnerBusinessKey>
    <Key>5798009811578</Key>
    <KeyType>GLN</KeyType>
    <UnitName></UnitName>
    <UnitCVR></UnitCVR>
    <GroupId>-1</GroupId>
    <Bindings>
        <Binding>
            <Id>45790</Id>
            <NetworkTypeId>1</NetworkTypeId>
            <ActivationDate>2014-01-01T23:59:00+01:00</ActivationDate>
            <ExpirationDate>2014-01-01T00:00:00+01:00</ExpirationDate>
            <ServiceId>46972</ServiceId>
        </Binding>
    </Bindings>
</UpdateParticipant>
```

The HTTP response has status code 409, and the Response Body is:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Errors>
    <Error>
        <Code>participant.update.unitname.empty</Code>
        <ErrorType>Error</ErrorType>
        <Id>unitName</Id>
    </Error>
    <Error>
        <Code>participant.update.expiration.before.now</Code>
        <ErrorType>Error</ErrorType>
        <Id>bindings</Id>
    </Error>
    <Error>
        <Code>participant.update.expiration.before.start</Code>
        <ErrorType>Error</ErrorType>
        <Id>bindings</Id>
    </Error>
    <Error>
        <Code>participant.update.key.already.exists</Code>
        <ErrorType>Error</ErrorType>
        <Id>key</Id>
    </Error>
</Errors>
```

4.11.3 Validation warnings

Validation warnings are similar to errors, but data are updated/persisted, but it “could be better” – you should be aware.

Example below uses UpdateParticipant XML from section 4.8, but with multiple data errors.

In this XML, we have 2 sequential binding periods, but there is a gap between them.

PUT request is sent to: https://registrationservice-test.nemhandel.dk/nemhandel-pors/rest/participant

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<UpdateParticipant>
    <Id>0</Id>
    <OwnerBusinessKey>34051178</OwnerBusinessKey>
    <Key>5798009800001</Key>
    <KeyType>GLN</KeyType>
    <UnitName>PoRS2 create test</UnitName>
    <UnitCVR>00000000</UnitCVR>
    <Bindings>
        <Binding>
            <Id>0</Id>
```

```
            <NetworkTypeId>1</NetworkTypeId>
            <ActivationDate></ActivationDate>
            <ExpirationDate>2018-01-01T00:00:00+01:00</ExpirationDate>
            <ServiceId>46972</ServiceId>
        </Binding>
        <Binding>
            <Id>0</Id>
            <ActivationDate>2018-01-01T00:05:00+01:00</ActivationDate>
            <ExpirationDate></ExpirationDate>
            <ServiceId>46972</ServiceId>
        </Binding>
    </Bindings>
</UpdateParticipant>
```

The HTTP response has status code 201, and the Response Body is:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Errors>
    <Error>
        <Code>participant.update.expiration.gap.more.than.one.minute</Code>
        <ErrorType>Warning</ErrorType>
        <Id>bindings</Id>
    </Error>
</Errors>
```

5 PEPPOL SMP data

Besides Danish NemHandel, NHR also acts as SMP in the PEPPOL infrastructure. This section describes how to utilize this.

5.1 NetworkTypes

NHR has been updated with 2 PEPPOL NetworkTypes:

ID	Name
1	NemHandel / RASP
2	PEPPOL / AS2
3	PEPPOL / AS4

In all examples above, NetworkTypeId has always been set to 1, but syntax for registering PEPPOL endpoints for a participant is the same.

5.2 What to do

Having understood the data model above, a few details needs to be added to utilize the NHR as PEPPOL SMP.

5.2.1 Profiling

Create a Profiling having NetworkTypeId set to 2 or 3, using PEPPOL enabled profiles. A profile is PEPPOL enabled if the property IsPeppolEnabled is set to true.

PEPPOL has the DocumentStandard ID=9, so a list of the PEPPOL specific profiles can be listed using https://registrationservice-test.nemhandel.dk/nemhandel-pors/rest/list/9/profiles
Please note, that some of the OIOUBL profiles are also PEPPOL enabled.

5.2.2 Receiver Service

Create a Receiver Service having NetworkTypeId set to 2 or 3 – depending on the PEPPOL protocol to use.
URL must be a valid PEPPOL Access Point (AP) URL (starting with https://) and the certificate must be the AP certificate in Base64 format (without -----BEGIN CERTIFICATE-----).

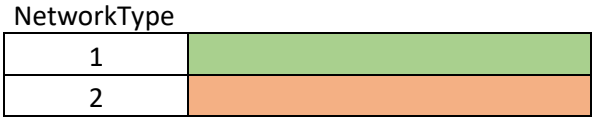
The referred ProfilingId can be either a Profiling with NetworkType 2 or 3, so that it is possible to have one Profiling used for both AS2 and AS4.

5.2.3 ParticipantBinding

It is now possible to have up to 3 ParticipantBindings in parallel – one per NetworkType.
The usual validation rules regarding overlap, gaps etc. for ActivationDate and ExpirationDate are evaluated per NetworkType.

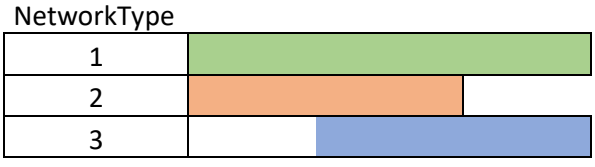
5.3 Examples

5.3.1 Example 1



This Participant has 2 ParticipantBindings – one for NemHandel and one for PEPPOL AS2 (no ActivationDate and ExpirationDate set).

5.3.2 Example 2



This Participant has 3 ParticipantBindings – one for each NetworkType. The PEPPOL AS2 has ExpirationDate set, and the PEPPOL AS4 has the ActivationDate set.