Functional test cases - Invalidation

Version 1.2

Table of contents

Introduction

1

The purpose of this test is that the Economic Operator (EO) can invalidate an already existing declaration before the goods have been released. This is called an invalidation. An invalidation request is sent by the EO to Toldsystemet. A declarant can apply for an invalidation without economic consequences if they apply for invalidation before the goods are released. However, an invalidation request can also be applied for released and paid goods but here we refer to the "Invalidation and repayment" test case.
Further, an invalidation request can be rejected if the declarant does not provide or fulfill the requirements for documentation for the reason of the invalidation.

1.1 Preconditions

To be able to execute the functional test cases for invalidation, the EO must have established a connection to Toldsystemet, see also the Connectivity Guide. Second, the submitter must be able to submit a pre-lodged declaration followed by an invalidation. This must be done using the following services:

- DMS.Import.Declaration.Submit
- DMS.Import.Declaration.Invalidate

To endpoint:

https://secureftpgatewaytest.skat.dk:6384/exchange/CVR_<CVR>_UID_<UID>

1.2 Process flow

The process flow for an invalidation request of an already existing declaration can be seen in Figure 1 below.
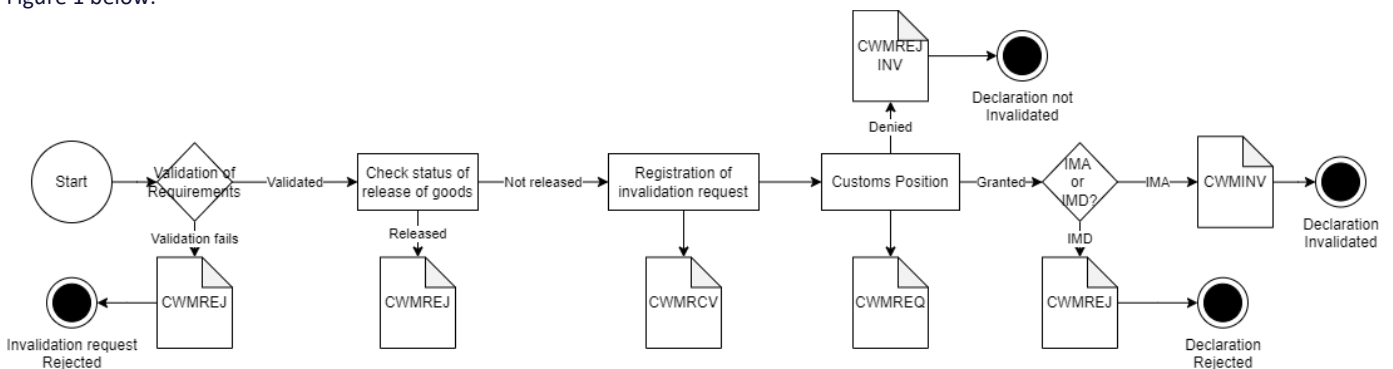

Figure 1 – Invalidation flow

Test scenarios

2

To complete the functional onboarding phase two, the EO must deliver confirmation of
passed test-cases for each test scenario:

Test no	Test Scenario	Passed
1	Send invalidation request which is rejected.	☐
2	Send invalidation request which is accepted.	☐

2.1   Detailed test scenarios

The following sections will describe the aim of the specific test scenarios and it will provide
the desired results of the test. It should be noted that there can be changes in the notifica-
tions provided by Toldsystemet, because the solution is still in development. If at a later
point more notifications will be added the EO will be informed about this.

2.1.1   Test 1 - Send invalidation request which is rejected

The aim of Test 1 is to test if the EO can obtain notification that the invalidation request has
been rejected.

Passed when a CWMREJ notification has been received after the invalidation request.

Test steps:

- Submit initial H7 declaration – you can use the provided IMD.xml (remember to fill
  in LRN)
- Test for wrong content by sending an invalidation request with insufficient reason
  (below 10 characters):

```
<ns3:Amendment>
  <ns3:SequenceNumeric>1</ns3: SequenceNumeric>
  <ns3:ChangeReasonCode>22</ns3:ChangeReasonCode >
  <ns3: ChangeReasonText>INV</ns3: ChangeReasonText>
</ns3: Amendment>
```

- Fill in the insufficient data in a invalidation XML – you can use the provided
  Inv_test1.xml, fill the XML with LRN and assigned MRN and submit the XML.
- Pull notification to see that invalidation request has been rejected – indicated by
  the **CWMREJ** notification.

2.1.2   Test 2 - Send invalidation request which is accepted

The aim of Test 2 is to test if the EO can obtain notification that the invalidation request has
been accepted, including the following notifications.

**Passed when the CWMRCV, CWMREQ and CWMINV notifications has been received after
the invalidation request.**

Test steps:

- Submit initial H7 declaration – you can use the provided IMD.xml (remember to fill
  in LRN)
- Send in an invalidation request with sufficient data:

```
<ns3:Amendment>
    <ns3:SequenceNumeric>1</ns3: SequenceNumeric>
    <ns3:ChangeReasonCode>22</ns3:ChangeReasonCode >
    <ns3: ChangeReasonText>Goods have been returned</ns3: ChangeReasonText>
</ns3: Amendment>
```

- Fill in the reason for invalidation in an invalidation XML – you can use the provided Inv_test2.xml, fill the XML with LRN and assigned MRN and submit the XML.
- Pull notification to see notification of granted invalidation request – **indicated by receiving the CWMRCV, CWMREQ and CWMINV notifications.**