Acceptance testing when messages arrive on Mercurius

Abstract
When messages arrive on the Mercurius platform, several tests are performed during the flow. Depending on the validation, several actions will be executed. This document tries to enlighten the black box where messages are going through during the different stages of processing in Mercurius before documents get ready to be retrieved by the customers.

Following stages can be identified when a message is arriving at the frontdoor of Mercurius:

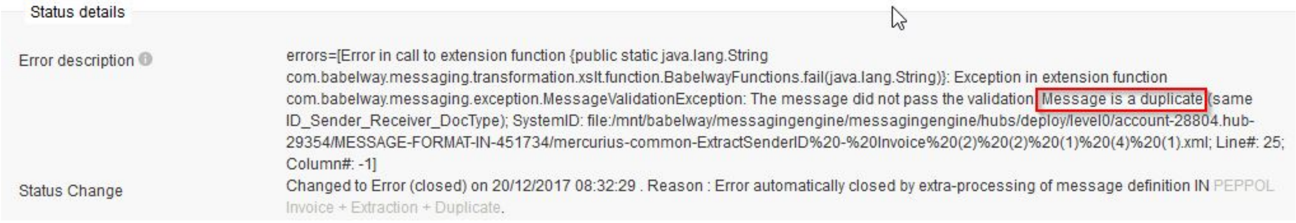
1. Message is received by Mercurius
2. Message is validated against PEPPOL rules (schematrons)
3. Message is tested for double
4. Message is tested for 'special rules' (explained later)
5. Message is tested for ePrior rules
6. Message could not be delivered
7. Message gets expired
8. Message is ready to be downloaded by the customer

Which tests are performed and what action are taken:

1. A message can only arrive on Mercurius when all AS2 traffic is performing well. Since all of this technical stuff is expected inside PEPPOL Access Points (including security and XSD validation), it is the responsibility of the Access Points to make this work. When this step has succeeded, an MDN (Message Disposition Notification) containing the original AS2transmissionID is sent to the sender of the message.
2. When entering the Mercurius system, the message is validated against the PEPPOL rules (called schematrons). Those rules, depending on which kind of document is send, can be consulted publically here . In case the message is not inline with PEPPOL rules, the message is in error and will be dropped. So no document will be extracted and no notification will be send.
   examples of what is tested (in PEPPOL schematrons): following list is not complete and only given as example
   a. valid document currency code
   b. document identifier cannot be null
   c. payable amount cannot be negative
   d. InvoicePeriod.StartDate.Date must be present (if InvoicePeriod is specified)
   e. InvoicePeriod.EndDate.Date must be present (if InvoicePeriod is specified)
   f. InvoicePeriod.EndDate must be bigger or equal to the start date (if InvoicePeriod is specified)
3. Next stage is the testing for doubles. This is done based on 4 elements: ID (documents identification eg. invoice number), sender (identification of the sender eg. CBE number), receiver (identification of the receiver eg. CBE number), and document type (invoice, credit note, ...). When a message is found duplicate of another message, it will be displayed in the portal but no information will be extracted. However, the message will be visible in the change log of the originally received document. In this case a business response can be send (MLR, e-mail, portal).

   > If invoice numbering does not include unique numbering on fiscal year, this could pose a problem when sending the same invoice number to the same customer in the next fiscal period (when numbering starts at 1 every year and few invoices are created, this is a potential risk).

   example:



Status details

Error description: errors=[Error in call to extension function {public static java.lang.String com.babelway.messaging.transformation.xslt.function.BabelwayFunctions.fail(java.lang.String)}: Exception in extension function com.babelway.messaging.exception.MessageValidationException: The message did not pass the validation Message is a duplicate (same ID_Sender_Receiver_DocType); SystemID: file:/mnt/babelway/messagingengine/messagingengine/hubs/deploy/level0/account-28804.hub-29354/MESSAGE-FORMAT-IN-451734/mercurius-common-ExtractSenderID%20-%20Invoice%20(2)%20(2)%20(1)%20(4)%20(1).xml; Line#: 25; Column#: -1]

Status Change: Changed to Error (closed) on 20/12/2017 08:32:29 . Reason : Error automatically closed by extra-processing of message definition IN PEPPOL Invoice + Extraction + Duplicate.

4. In the next step the message is checked for:
   a. identifier string length not more than 250 characters
   b. Identifier has only characters from the 7-bit ASCII table
   c. IssueDate must not be in the future
   If the message fails, it will be visible in the portal, no attachments will be extracted and a business response can be send (MLR, e-mail, portal).
5. Now the message is validated against the ePrior rules (ePrior schematrons). When the message gets failed in this stage, there is a bug suspicion and an e-mail is send to the Mercurius support team to investigate the situation.
6. If for some reason the message can not be delivered afterwards, it is a technical issue and an e-mail is send to the Mercurius support team to investigate the situation.
7. Now the document can be extracted from the message and is ready to be downloaded by the customer. If for some reason, the

document is not downloaded within 30 days, Mercurius signals an expired event and an e-mail is send to the responsable at customer side (or to the Mercurius support team in case of no available email address).

8. Document is downloaded by the customer and business validations can start at customer level (eg. presence of Purchase order number). These validations are customer specific and can result in a business response send back to the supplier.