HQ EOI API Gateway

Overview

Api gateway is to allow utilization within HQ by HQ internal needs or vendors to receive real time. It utilizes the typical vendor tokenization for external vendors and internal tokenization for internal use.

The API call to API gateway is an EOI order that carries subtype. The subtype is looked up within HQ definitions and expanded into dataset request and sent to store. EOI connector then executes a data collection for each dataset and sends back all datasets. Parametrization for OPENDATE will be supported.

API URLs

URL and conventions for calling API gateway (same as EOI)

PROD: https://www.pixelpointhq.com/api/v1/eoi
QA: https://qa.pixelpointhq.com/api/v1/eoi
CI: https://ci.pixelpointhq.com/api/v1/eoi

Method: POST
Content-Type: "application/xml"

Sample request and transformation

Sample of XML sent for example by vendor:

```
<?xml version="1.0"?>
<Request>
  <RequestType>REALTIME_DATA</RequestType>
  <RequestSubType>SALES_101</RequestSubType>
  <DataParameters OPENDATE="20220110"  />
```

```
  <EOITransfer>
      <EOIAuthenticationToken>Vendor+Store</EOIAuthenticationToken>
  </EOITransfer>
</Request>
```

Such a request is then transformed by HQ based on stored definitions into for example:

```
<?xml version="1.0"?>
<Request>
  <RequestType>REALTIME_DATA</RequestType>
  <RequestSubType>SALES_101</RequestSubType>
  <DataParameters OPENDATE="20220110"  />
  <EOITransfer>
    <EOIAuthenticationToken>Vendor+Store</EOIAuthenticationToken>
  </EOITransfer>
  <DataSets>
       <DataSet>
         select UseVat, now() as StoreLocalDateTime from sysinfo
       </DataSet>
       <DataSet>
         select count(*) as TransactionCount , sum(NetTotal) as NetTotal_sum ,
         sum(FinalTotal) as FinalTotal_sum
         from posheader
         where posheader.opendate = :OPENDATE
       </DataSet>
  </DataSets>
</Request>
```

For the transformation it will need to use a table in core DB.

Primary key can be the value corresponding to <RequestData>
There should be a column for IsActive. Only active records should take effect.
There should be provision for 9 SQL Blobs. There should be a column for internal description and external one.. (to be given out .. upon request catalog call)
An automated Hash of 9 SQL BLOBs would be nice.. To allow comparison of records between environments.

Much later on we will add a promotion flag to each record.

It might be nice to only load each definition once to memory from DB… and use then form stored memory data. Total size of all SQLs is expected to be under 1KB per record on average.

Assuming that we end up using the “API gateway” for many purposes, I would expect it be less than 50 records within a year and 100-200 solves a lot of our needs.

Parametrization: OPENDATE

For current OPENDATE, date can be passed in yyyymmdd format or can use the key word: CURRENT or MAX which will use date from last (=max) UID in DAYINFO table.
List of features not supported by EOI gateway (compared to EOI orders processed by POS):

The use of RequestUID for “repeatability” is implemented at this time.

Parameter case has to be same case in SQL and parameter provided in <DataParameters> node.

```
<?xml version="1.0"?>
<Request>
  <RequestType>REALTIME_DATA</RequestType>
  <RequestSubType>SALES_101</RequestSubType>
  <DataParameters OPENDATE="20220110"  />
  <DataSets>
    <DataSet>
       select opendate, sum( NUMCUST) as Y , sum(NETTOTAl) as X from posheader where opendate = :OPENDATE group by opendate
    </DataSet>
  </DataSets>
  <EOITransfer>
    <EOIAuthenticationToken>EOI001-EOI17CBC4B44262849</EOIAuthenticationToken>
  </EOITransfer>
</Request>
```

ERRORs

Example of how a failure might be express in response XML

Pretty Raw Preview XML

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
    <ErrorNum>0</ErrorNum>
    <Status/>
    <POSProcessTimeUTC>2022-11-04T17:06:38.891</POSProcessTimeUTC>
    <DataSets time_taken="0.05">
        <DataSet debug_paramters="0" id="1" message="error_in_sql_processing" time_taken="0.05"/>
    </DataSets>
    <MiscInfoStore>
        <EoiPath>C:\PixelPos\</EoiPath>
        <EoiVer>1.3.1.230</EoiVer>
    </MiscInfoStore>
    <EoiStoreReceived>2022-11-04 17:06:40.0</EoiStoreReceived>
    <Misc>
        <EOIServer>172.31.21.228</EOIServer>
        <EOIQueueID>1000000001386</EOIQueueID>
        <CloudStoreID>14000249</CloudStoreID>
    </Misc>
</Response>
```

If request contain multiple datasets, some might produce “OK” but some might have errors… an error/pass is per dataset

Filed names in Result

The field names in query need to be defined.
Don't use for example: SUM(X)
Instead use named result: SUM(X) as SUM_X

Example of correct use:

```
<DataSet>
   select opendate, sum( NUMCUST) as Y , sum(NETTOTAl) as X from posheader where opendate = :OPENDATE  group by opendate
</DataSet>
```

OUTPUT

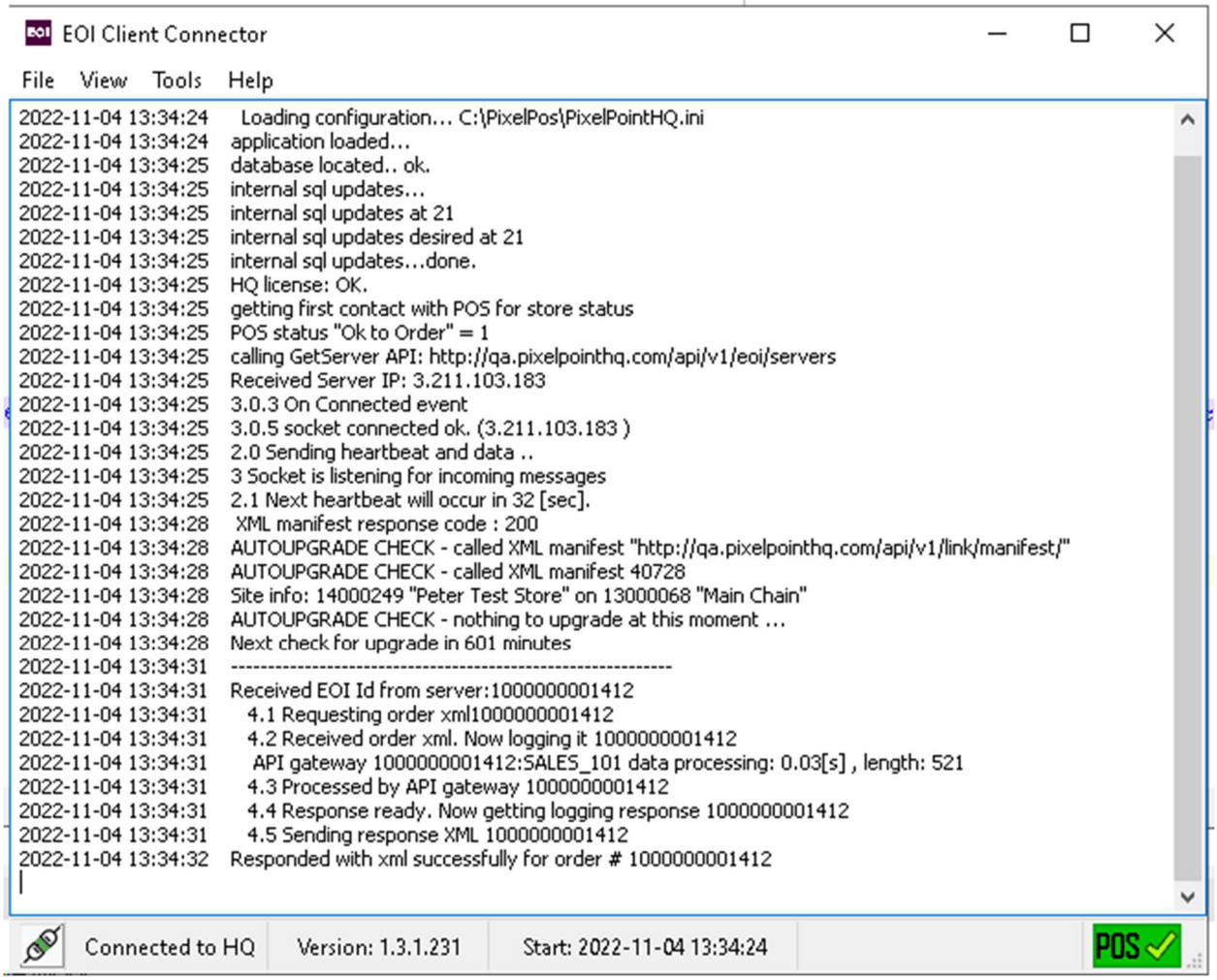
Sample request and to such request:

```
<?xml version="1.0"?>
<Request>
  <RequestType>REALTIME_DATA</RequestType>
  <RequestSubType>SALES_101</RequestSubType>
  <DataParameters OPENDATE="20220110"  />
  <DataSets>
    <DataSet>
       select opendate, sum( NUMCUST) as Y , sum(NETTOTAl) as X from posheader
       where opendate = :OPENDATE  or  opendate = :OPENDATE - 7
       group by opendate
    </DataSet>
    <DataSet>
       select now() as StoreDateTime
    </DataSet>
  </DataSets>
  <EOITransfer>
    <EOIAuthenticationToken>[redacted]</EOIAuthenticationToken>
  </EOITransfer>
</Request>
```

EOI connector

To use the feature you must run EOI connector at minimum version of: 1.3.1.231

Such EOI orders have an extra line in console logging:


EOI Client Connector
File View Tools Help
2022-11-04 13:34:24 Loading configuration... C:\PixelPos\PixelPointHQ.ini
2022-11-04 13:34:24 application loaded...
2022-11-04 13:34:25 database located.. ok.
2022-11-04 13:34:25 internal sql updates...
2022-11-04 13:34:25 internal sql updates at 21
2022-11-04 13:34:25 internal sql updates desired at 21
2022-11-04 13:34:25 internal sql updates...done.
2022-11-04 13:34:25 HQ license: OK.
2022-11-04 13:34:25 getting first contact with POS for store status
2022-11-04 13:34:25 POS status "Ok to Order" = 1
2022-11-04 13:34:25 calling GetServer API: http://qa.pixelpointhq.com/api/v1/eoi/servers
2022-11-04 13:34:25 Received Server IP: 3.211.103.183
2022-11-04 13:34:25 3.0.3 On Connected event
2022-11-04 13:34:25 3.0.5 socket connected ok. (3.211.103.183)
2022-11-04 13:34:25 2.0 Sending heartbeat and data ..
2022-11-04 13:34:25 3 Socket is listening for incoming messages
2022-11-04 13:34:25 2.1 Next heartbeat will occur in 32 [sec].
2022-11-04 13:34:28 XML manifest response code : 200
2022-11-04 13:34:28 AUTOUPGRADE CHECK - called XML manifest "http://qa.pixelpointhq.com/api/v1/link/manifest/"
2022-11-04 13:34:28 AUTOUPGRADE CHECK - called XML manifest 40728
2022-11-04 13:34:28 Site info: 14000249 "Peter Test Store" on 13000068 "Main Chain"
2022-11-04 13:34:28 AUTOUPGRADE CHECK - nothing to upgrade at this moment ...
2022-11-04 13:34:28 Next check for upgrade in 601 minutes
2022-11-04 13:34:31 ----------
2022-11-04 13:34:31 Received EOI Id from server:1000000001412
2022-11-04 13:34:31 4.1 Requesting order xml1000000001412
2022-11-04 13:34:31 4.2 Received order xml. Now logging it 1000000001412
2022-11-04 13:34:31 API gateway 1000000001412:SALES_101 data processing: 0.03[s] , length: 521
2022-11-04 13:34:31 4.3 Processed by API gateway 1000000001412
2022-11-04 13:34:31 4.4 Response ready. Now getting logging response 1000000001412
2022-11-04 13:34:31 4.5 Sending response XML 1000000001412
2022-11-04 13:34:32 Responded with xml successfully for order # 1000000001412
Connected to HQ
Version: 1.3.1.231
Start: 2022-11-04 13:34:24
POS

```xml
<?xml version="1.0" encoding="UTF-8"?>
<Response>
    <ErrorNum>0</ErrorNum>
    <Status/>
    <POSProcessTimeUTC>2022-11-04T17:27:09.815</POSProcessTimeUTC>
    <DataSets time_taken="0.02">
        <DataSet id="1" message="sql_located" time_taken="0.02">
            <Record>
                <OPENDATE>2022-01-03T00:00:00</OPENDATE>
                <Y>83</Y>
                <X>1650.58</X>
            </Record>
            <Record>
                <OPENDATE>2022-01-10T00:00:00</OPENDATE>
                <Y>51</Y>
                <X>884.18</X>
            </Record>
        </DataSet>
        <DataSet id="2" message="sql_located" time_taken="0.00">
            <Record>
                <STOREDATETIME>2022-11-04T13:27:09</STOREDATETIME>
            </Record>
        </DataSet>
    </DataSets>
    <MiscInfoStore>
        <EoiPath>C:\PixelPos\</EoiPath>
        <EoiVer>1.3.1.230</EoiVer>
    </MiscInfoStore>
    <EoiStoreReceived>2022-11-04 17:27:11.0</EoiStoreReceived>
    <Misc>
        <EOIServer>172.31.21.228</EOIServer>
        <EOIQueueID>1000000001411</EOIQueueID>
        <CloudStoreID>14000249</CloudStoreID>
    </Misc>
</Response>
```

It is a sample of getting some basic sales for a given day and day a week prior (to compare …)
There is also a 2nd dataset in the sample, that is for getting what time is/was at the local store.