TAQ02 V1.0.0 - API Specification

1.Document Change Control

Project Code: TAQ02

Interface Type: Web service

Interface Name: SendTransactionFile

Author: Vish Murugesan

Date: 03.09.2018

MDG Specification Version: V1.0.0

MDG Specification Status: Draft

Referred QAM Specification: CDS128 Quota Allocation - System Design Document.docx

SDD: Tariff and Quota Solution Design Document v2.0.pdf

SDD Crossreference/Feature: DIT file integration (SDD v2.0 data flow 4 and 5)

1.1 Version Control

MDG Specification Version	Change	Updated by	Updated Date
0.1	Initial Creation Added QAM WSDL and Schemas	Vish	21-Aug-2018

2. MDG API Properties

Property	Value
Context	**/taricfile/sendtransactionfile/v1**
Name	SendTransactionFile
Protocol	https
Response Caching	Disabled
Throttling	Unlimited
Version	v1

Path: /taricfile/sendtransactionfile/1.0.0

Produces: "text/xml"

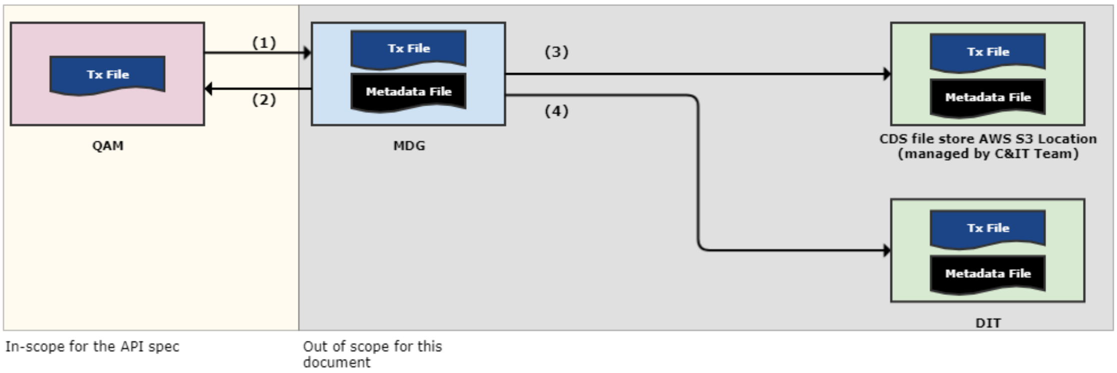
Consumes: "text/xml"

3. Objective

The objective of TAQ02 - sendtransactionfile API is as below:

(1) QAM to send a request with base64 encoded transaction log in the messageData element of the request payload

(2) MDG to receive the request and send an appropriate response to QAM


Tx File
QAM
(1)
(2)
Tx File
Metadata File
MDG
(3)
(4)
Tx File
Metadata File
CDS file store AWS S3 Location (managed by C&IT Team)
Tx File
Metadata File
DIT
In-scope for the API spec
Out of scope for this document

4. Scope

4.1 In Scope

1. Receive transaction log from QAM
2. Respond to QAM with appropriate response
3. Only QAM transaction log XML file is in scope

4.2 Out of Scope

1. Guaranteed delivery

5.Risks, Assumptions, Issues, Dependencies and Decisions

5.1 Risks

MDG are requested to conform to QAM WSDL and schema. Hence if there is a change to the QAM WSDL and schemas then there is a potential rework for MDG.

5.2 Assumptions

None Identified.

5.3 Issues

None Identified

5.4 Dependencies

1. Approved SDD
2. QAM to review and agree MDG API spec

5.5 Decisions

None Identified.

6. Validation Requirements

1. MDG to validate the requests sent by QAM using QAM request schema for uploadFile operation.

7. Security Requirements

1. QAM SOAP call will be a HTTPs call

8. Non-functional Requirements

None Identified

9. Return Codes

HTTP Status Code	Error scenario
202	If the request is accepted by MDG for processing
400	This status code will be returned in case of application error/business functional errors such as incorrect data, incorrect data format, missing parameters etc.
500	In case of a system error such as time out, server down etc, this HTTP status code will be returned

Note:

1. In cases such as server time out where a valid error code will not be returned by the back end system error code will be same as the HTTP status code i.e. 500
2. In case of business error scenario such as "duplicate submission", MDG will respond with errorCode 400 and appropriate errorMessage
3. Error response structure will conform to QAM's service fault schema

10. MDG Request and Response Structure

MessageData Request Structure

MDG request will conform to the QAM schema. Hence the below field structure is an exact replica of the QAM request structure.

Request Structure:

Field Name	Size	Optional/Mandatory	Field Type	Description
uploadFile		M	complex	
request		M	complex	
commSource		M	complex	
protocol		O	string	Enums: HTTP HTTPS
host		M	string	
port		O	int	
commTarget		O	complex	
protocol		O	string	Enums: HTTP HTTPS
host		M	string	
port		O	int	
commId		M	string	

commCorrelationId		O	string	
messageType		M	string	Enums: TARIC3 QUOTA2 QUOTA2_REQ QUOTA2_ACK QUOTA2_ALLOC TXLOG
messageData		M	base64Binary	TXLog content will be populated in this element. This must be extracted and transferred to DIT.

Response Structure

uploadFileRequest Response Structure

Field Name	Size	Optional/Mandatory	Field Type	Description
result		M	Complex	

result complex element structure

Field Name	Size	Optional/Mandatory	Field Type	Description
status		M	String	Enums: SUCCESS FAILURE
description		O	String	
error		O	Complex	

error complex element structure

Field Name	Size	Optional/Mandatory	Field Type	Description
fault		O	Complex	
errorCode		O	String	
errorMessage		O	String	
message		O	String	

11. Error responses

Error response will follow the below format:

Field Name	Size	Optional/Mandatory	Field Type	Description
ServiceFault		M	complex	
fault		O	complex	
errorCode		O	string	
errorMessage		O	string	
message		O	string	

12. Request and Response Examples

Request Example: Request from QAM to MDG

```
<soapenv:Envelope xmlns:all="http://allocation.quota.eurodyn.com/"
   xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
   <soapenv:Header xmlns:wsa="http://www.w3.org/2005/08/addressing">
      <wsse:Security soapenv:mustUnderstand="1"

xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wsse
curity-secext-1.0.xsd"

xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssec
urity-utility-1.0.xsd">
         <wsse:UsernameToken
wsu:Id="UsernameToken-7A196EF35B9E96173115410034126722 4">
            <wsse:Username>txlogToDitUser</wsse:Username>
            <wsse:Password

Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-t
oken-profile-1.0#PasswordDigest"
               >bBPhFgtSp3j8PhwbBznoXSRmKwA=</wsse:Password>
            <wsse:Nonce

EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-so
ap-message-security-1.0#Base64Binary"
               >qXFOcz+0NlPUrVEot3Of3w==</wsse:Nonce>
            <wsu:Created>2018-10-31T16:30:12.672Z</wsu:Created>
         </wsse:UsernameToken>
      </wsse:Security>
      <wsa:Action soapenv:mustUnderstand="1"

>http://allocation.quota.eurodyn.com/InboundComm/uploadFile</wsa:Action>
      <wsa:MessageID soapenv:mustUnderstand="1"
         >uuid:d38d2fda-fe34-42ec-a40f-2005f855df49</wsa:MessageID>
      <wsa:To soapenv:mustUnderstand="1"

>http://tariff-app3.eurodyn.com:8080/InboundCommon/InboundCommService</w
sa:To>
   </soapenv:Header>
   <soapenv:Body>
      <all:uploadFile>
         <request>
            <commSource>
               <host>QAM</host>
            </commSource>
            <commTarget>
               <host>DIT</host>
            </commTarget>
            <commId>2020</commId>
            <messageType>TXLOG</messageType>

<messageData>UEsDBBQAAAAIAI2TX02y8p974gEAAKQGAAAbABwAVEdCMjAwMzJfMjAxODE
wMzExNTU4MDMueG1s
```

```
VVQJAAMq2NlbKtjZW3V4CwABBOgDAAAE6AMAAJ1VXW+bQBB8969AvPt82G0TI5vIaqhVKU2q
xKna

xzO3TU6Cg9yHlfz7Lhw4BIPj1LIfvDszt7u3DIuL5yz1dqC0yOXSDwj1PZBJzoV8WPr3m2/j
c/8i

Gi1A7kL8QZoX4CFF6qVvlQwLu01FIng4Di/Xm9Xv+8tws7r9/jX8Ed/drdZxWClWhJI/SFrH
1/Ht

6iqMr3/FVzc/a57gS39K6WzmRyMPP1UZRjGpWWKw4AoQ1Mk9gBUFyUBr9gAHgAqU261TyYQu
+/7v

+hwP5T4wjLe17OtRkOSKHybfFuzaJoJHwWLSEx2mO32CNwvR7Iw6djs4TNV22wbSmtwJv3u0
hieL

qwVE2mwLqulgIDssZwvODBDzUkCj0Q4NE59sbhjh8FdIUQ6sHzoIJ7oc+3xen9mbPkURuwVV
t1ne

GZ1Pv9B5W7SLOK66Y6ngwrwQbZgypJxENKXBdEwD/DrdPsyJqiD5XvOsV3OPQO6HB1AN9dP5
58H+

3x/rLk9thrsQNIW5/8dJ1a2xlDRkR+1Ej0tk7FlkNiMFrrAofaRROUwcF0oUnprgsXg7OMY/
TqUT

PVHCPCrQj3mKe9WRec30PyK9W93jVq2ntuOrr45UG2vLmCcdZ64NfdJxdHzVTNrvmsgbjf4B
UEsB

Ah4DFAAAAAgAjZNfTbLyn3viAQAApAYAABsAGAAAAAAAAQAAAKSBAAAAAFRHQjIwMDMyXzIw
MTgx

MDMxMTU1ODAzLnhtbFVUBQADKtjZW3V4CwABBOgDAAAE6AMAAFBLBQYAAAAAAQABAGEAAAA3
AgAA
               AAA=</messageData>
         </request>
      </all:uploadFile>
   </soapenv:Body>
```

```
</soapenv:Envelope>
```

Response Example:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Header>
        <Action xmlns="http://www.w3.org/2005/08/addressing"

>http://allocation.quota.eurodyn.com/InboundComm/uploadFileResponse</Act
ion>
        <MessageID xmlns="http://www.w3.org/2005/08/addressing"
            >urn:uuid:8a42979a-5810-4e53-887c-0d6a06bc27d4</MessageID>
        <To xmlns="http://www.w3.org/2005/08/addressing"
            >http://www.w3.org/2005/08/addressing/anonymous</To>
        <RelatesTo xmlns="http://www.w3.org/2005/08/addressing"
            >uuid:d38d2fda-fe34-42ec-a40f-2005f855df49</RelatesTo>
    </soap:Header>
    <soap:Body>
        <ns2:uploadFileResponse
xmlns:ns2="http://allocation.quota.eurodyn.com/">
            <result>
                <status>SUCCESS</status>
                <description>OK</description>
            </result>
        </ns2:uploadFileResponse>
    </soap:Body>
</soap:Envelope>
```

Error Response Example:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Header>
        <Action xmlns="http://www.w3.org/2005/08/addressing"

>http://allocation.quota.eurodyn.com/InboundComm/uploadFileResponse</Act
ion>
        <MessageID xmlns="http://www.w3.org/2005/08/addressing"
            >urn:uuid:4ce6ddd2-2d88-42b7-b76e-83eed774f749</MessageID>
        <To xmlns="http://www.w3.org/2005/08/addressing"
            >http://www.w3.org/2005/08/addressing/anonymous</To>
        <RelatesTo xmlns="http://www.w3.org/2005/08/addressing"
            >uuid:daf720d4-4dc4-45c2-bcf3-4f64ee706169</RelatesTo>
    </soap:Header>
    <soap:Body>
        <ns2:uploadFileResponse
xmlns:ns2="http://allocation.quota.eurodyn.com/">
            <result>
                <status>FAILURE</status>
                <description>ERROR CODE: 04</description>
                <error>
                    <fault>
                        <errorCode>04</errorCode>
                        <errorMessage>Source and target can't be
equal</errorMessage>
                    </fault>
                    <message>Logical error</message>
                </error>
            </result>
        </ns2:uploadFileResponse>
    </soap:Body>
</soap:Envelope>
```

13. Schema

Schema: InboundCommService_schema1.xsd

WSDL: InboundCommService_with_policy.wsdl

Samples: Samples.zip