

# TAQ05 V1.0.0 - API Specification

## 1.Document Change Control

---

**Project Code:** TAQ05

**Interface Type:** Web service

**Interface Name:** sendTariffTransactionFile

**Author:** Vish Murugesan

**Date:** 28.11.2018

**MDG Specification Version:** V1.0.0

**MDG Specification Status:** **Draft**

**Referred ED-Tariff Specification:** [ECDP-QAM Web Services-v1.0.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	Vish	28-Nov-2018

## 2. Tariff API Properties

---

Property	Value
Context	/taricfile/sendTariffTransactionFile/v1
Response Caching	Disabled
Protocol	https
Name	sendTariffTransactionFile
Throttling	Unlimited
Version	v1

**Produces:** "text/xml"

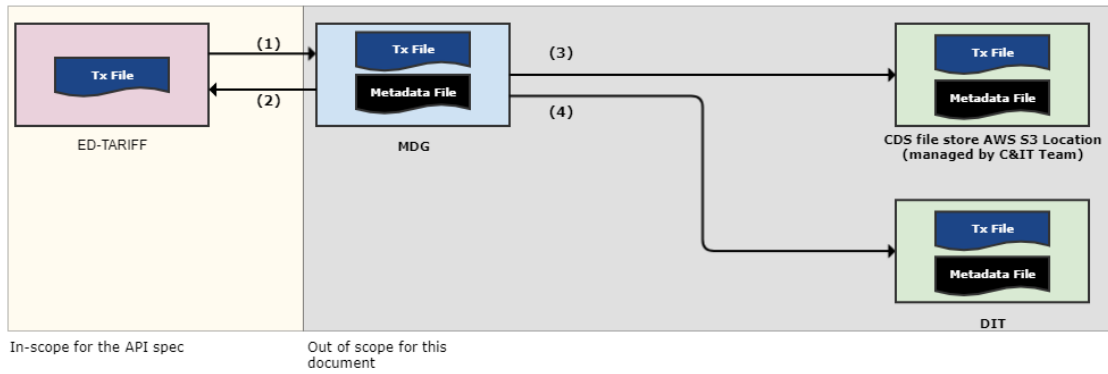
**Consumes:** "text/xml"

### 3. Objective

---

The objective of TAQ05 - sendTariffTransactionFile API is as below:

- (1) ED-Tariff to send a request with base64 encoded transaction log in the payload.
- (2) MDG to receive the request and send an appropriate response.



## 4. Scope

### 4.1 In Scope

1. Receive transaction log from ED-Tariff
2. Respond to ED-Tariff with appropriate response
3. Only ED-Tariff 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

If ED-Tariff does not conform to ED InboundCommService.uploadFile schemas then there is a potential risk of MDG rework and a potential impact to project timelines.

### 5.2 Assumptions

It is assumed that ED-Tariff will conform to ED's InboundCommService request and response schemas.

### 5.3 Issues

None Identified

### 5.4 Dependencies

1. Approved SDD
2. ED-Tariff to review and agree MDG API spec

### 5.5 Decisions

None Identified.

## 6. Validation Requirements

1. MDG to validate the requests sent by ED-Tariff

## 7. Security Requirements

1. Tariff's SOAP call will be a HTTPs call

## 8. Non-functional Requirements

None Identified

## 9. Return Codes

HTTP Status Code	Error scenario
204	If the request is processed successful
40X	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 ie 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 - [MDG Error Response Structure.pdf](#)

## 10. MDG Request and Response Structure

### Request Structure

#### uploadFile Request Structure

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

#### request complex element structure

Field Name	Size	Optional/Mandatory	Field Type	Description
commSource		M	Complex	
commTarget		O	Complex	
commId		M	String	
commCorrelationId		O		
messageType		M	String	Enums: TARIC3 QUOTA2 QUOTA2_REQ QUOTA2_ACK QUOTA2_ALLOC TXLOG
messageData		M	base64binary	

### commSource complex element structure

Field Name	Size	Optional/ <u>M</u> andatory	Field Type	Description
protocol		O	String	Enums: HTTP HTTPS
host		M	String	Tariff would populate "Tariff" when the request is sent to MDG
port		M	String	

### commTarget complex element structure

Field Name	Size	<u>O</u> ptional/ <u>M</u> andatory	Field Type	Description
protocol		O	String	Enums: HTTP HTTPS
host		M	String	Tariff would populate "DIT" when the request is sent to MDG
port		M	String	

## Response Structure

### uploadFileResponse Structure

Field Name	Size	<u>O</u> ptional/ <u>M</u> andatory	Field Type	Description
result		M	Complex	

### result complex element structure

Field Name	Size	<u>O</u> ptional/ <u>M</u> andatory	Field Type	Description
status		M	String	Enums: SUCCESS FAILURE
description		O	String	
error		O	Complex	

### error complex element structure

Field Name	Size	<u>O</u> ptional/ <u>M</u> andatory	Field Type	Description
fault		O	Complex	
errorCode		O	String	
errorMessage		O	String	
message		O	String	

## 12. Request and Response Examples

### Request Example:

```
<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-7A196EF35B9E961731154100341267224">
        <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"
        >qXF0cz+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>Tariff</host>
                </commSource>
                <commTarget>
                    <host>DIT</host>
                </commTarget>
                <commId>2020</commId>
                <messageType>TXLOG</messageType>

<messageData>UESDBBQAAAAIAI2TX02y8p974gEAAKQGAAAbABwAVEdCMjAwMzJfMjAxODE
wMzExNTU4MDMueGls

VVQJAAMq2NlbKtjZW3V4CwABBOgDAAAE6AMAAJ1VXW+bQBB8969AvPt82G0TI5vIaqhVKU2q
xKna

xz03TU6Cg9yHlfz7Lhw4BIPj1LIfvDszt7u3DIuL5yz1dqC0yOXSDwj1PZBJzoV8WPr3m2/j
c/8i

GilA7kL8QZoX4CFF6qVvlQwLu01FIng4Di/Xm9Xv+8tws7r9/jX8Ed/drdZxWC1WhJI/SFrH
1/Ht

6iqMr3/FVzc/a57gS39K6WzmRyMPP1UZrjGpWWKw4AoQ1Mk9gBUFYUBr9gAHgAqU261TyYQu
```

+ /7v

+hwP5T4wjLe170tRkOSKHybfFuzaJoJHwWLSEx2mO32CNwvR7Iw6djs4TNV22wbSmtwJv3u0  
hieL

qwVE2mwLqulgIDssZwvODBDzUkCj0Q4NE59sbhjh8FdIUQ6sHzoIJ7oc+3xen9mbPkURuwVV  
tlne

GZlPv9B5W7SLOK66Y6ngwrwQbZgypJxENKXBdEwD/DrdPsyJqiD5XvOsV3OPQO6HB1AN9dP5  
58H+

3x/rLk9thrsQNIW5/8dJ1a2xlDRkR+1Ej0tk7FlkNiMFrrAofaRROUwcf0oUnprgsXg7OMY/  
TqUT

PVHCPCrQj3mKe9WRec30PyK9W93jVq2ntuOrr45UG2vLmCcdZ64NfdJxdHzVTNrvmgbj4B  
UEsB

Ah4DFAAAAAgAjZNfTbLyn3viaQAAPAYAABsAGAAAAAAAAQAAAKSBAAAAFRHQjIwMDMyXzIw  
MTgx

MDMxMTU1ODAzLnhtbFVUBQADKtjZW3V4CwABBOgDAAAE6AMAAFBLBQYAAAAAAQABAGEAAAA3  
AgAA

AAA=</messageData>  
</request>

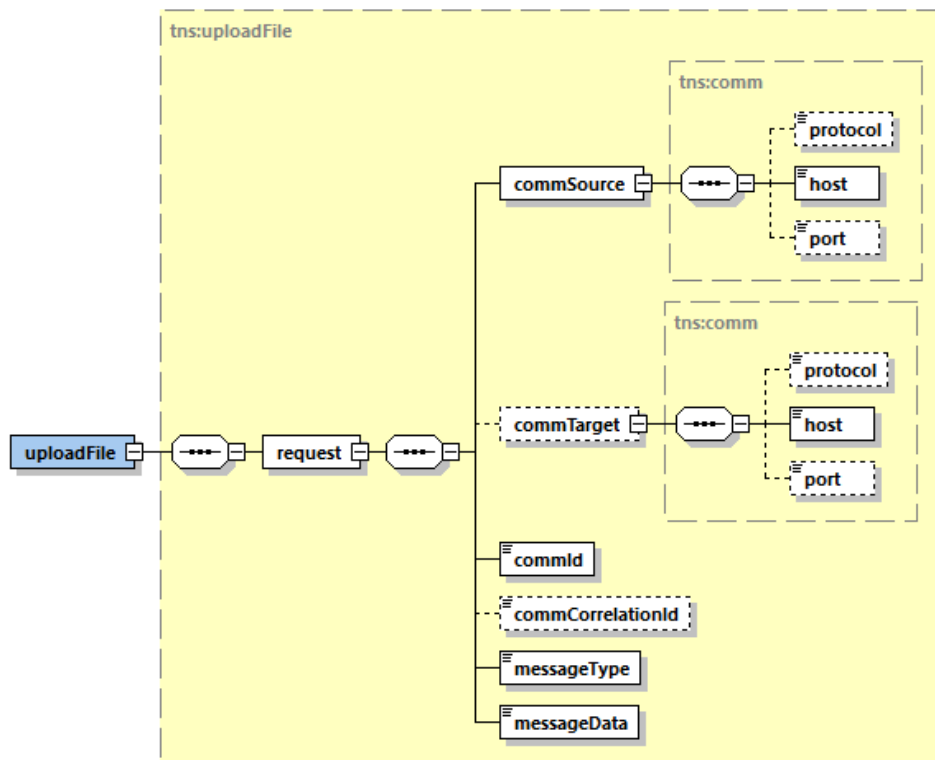
```
        </all:uploadFile>
    </soapenv:Body>
</soapenv:Envelope>
```

#### Response Example:

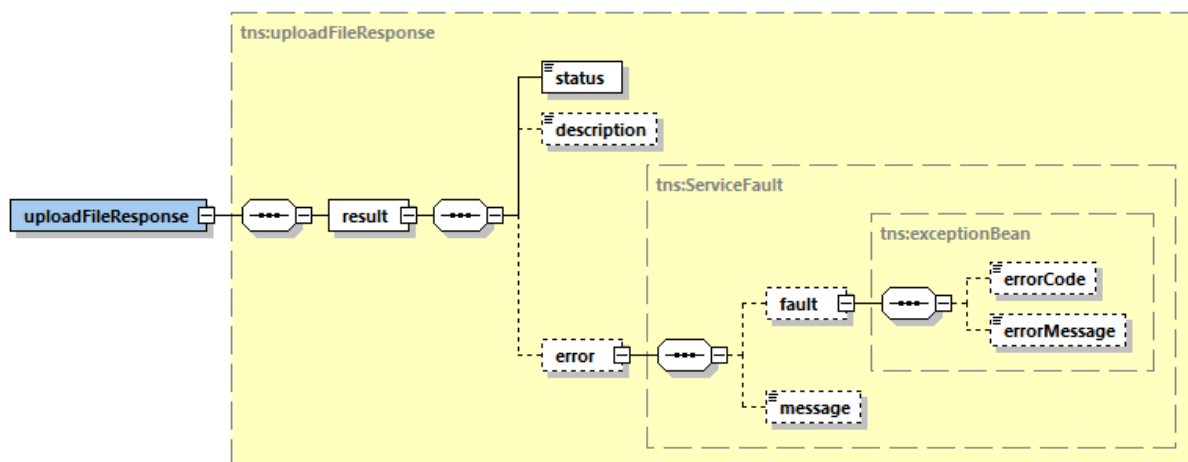
```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Action xmlns="http://www.w3.org/2005/08/addressing"
>http://allocation.quota.eurodyn.com/InboundComm/uploadFileResponse</Action>
    <MessageID xmlns="http://www.w3.org/2005/08/addressing"
>urn:uuid:8a42979a-5810-4e53-887c-0d6a06bc27d4</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>
```

## 13. Request Schema

InboundCommService\_schema1.xsd



## 14. Response Schema



## 15. WSDL

InboundCommService\_with\_policy.wsdl