Oracle FCCM Cloud Service Transaction Filtering

Using Transaction Filtering API

Release 25.08.01
G40822-01
August 2025

Oracle FCCM Cloud Service Transaction Filtering Using Transaction Filtering API, Release 25.08.01

G40822-01

Contents

Preface

Using Transaction Filtering API introduces information sources that can help you use the Oracle Financial Services Transaction Filtering Cloud Service (OFS TF CS) API.

Audience

This document is intended for users who are responsible for provisioning and activating Oracle Transaction Filtering Cloud services or for adding other users who would manage the services, or for users who want to develop Oracle Cloud applications.

Help

Use Help Icon  to access help in the application. If you don't see any help icons on your page, click your user image or name in the global header and select Show Help Icons. Not all pages have help icons. You can also access the https://docs.oracle.com/en/ to find guides and videos.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

Related Resources

For more information, see these Oracle resources:

- Oracle Public Cloud: http://cloud.oracle.com
- Community: Use https://community.oracle.com/customerconnect/ to get information from experts at Oracle, the partner community, and other users.

- Training: Take courses on Oracle Cloud from https://education.oracle.com/oracle-cloud-learning-subscriptions.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Comments and Suggestions

Please give us feedback about Oracle Applications Help and guides! You can send an e-mail to: https://support.oracle.com/portal/.

1

About the REST APIs

You can use Oracle REST APIs to view data stored in Oracle Cloud Service. A REST API (also known as RESTful API) is an application programming interface (API or web API) that conforms to the constraints of REST architectural style and allows for interaction with RESTful web services.

Oracle Transaction Filtering Cloud Service provides a Representational State Transfer (REST) API to perform user tasks in an automated manner. The REST API is an application-programming interface that provides a simplified way to exchange data through HTTP requests from a client to the server. In REST APIs, a resource is an object with a type, associated data, and relationships to other resources. You can use a set of HTTP methods to access each resource.

Resources are organized in a hierarchical structure that enables:

- Better organization, by grouping related data so that you can efficiently customize the resources.
- Improved performance by using a single HTTP request to handle multiple resources.

2
Getting Started

Before executing the Rest APIs and performing various tasks, refer to the following topics to meet the necessary requirements:

- Prerequisites
- Authentication
- Supported Methods
- Supported Headers
- Status Code

2.1 Prerequisites

Prerequisites for executing Rest APIs.

The following are the set of prerequisites required for executing/invoking Rest APIs.

- Access to Transaction Filtering (TF) service.
- Appropriate user privileges to access the services.
- Technical and functional knowledge to understand and execute the REST APIs and configuration knowledge.
- Knowledge of REST concepts, JSON, and browser-based REST clients.
- Knowledge of an interactive and automatic tool for verifying the APIs, such as Postman.

2.1.1 Obtain Account Information

The account creation e-mail from Oracle contains the identity domain name for the Oracle Transaction Filtering Cloud instance. If you do not have this information, then contact your service administrator.

2.2 Authentication

The Authentication Process involves the use of cURL Commands in a CLI Tool to generate the access token and invoke REST APIs. The Authentication Token is generated through the OAuth Client ID and Secret Credentials created in IAM/IDCS during Provisioning. The Authentication Token does not require that you log in to the AFCS Application to invoke the REST APIs from external applications.

Ensure that you have the appropriate log-in credentials for accessing the Oracle Transaction Filtering cloud service, and the appropriate role for creating, managing, and deleting service instances.

To get authentication, follow these steps:

1. Log in to **Admin Console**. For more information, see Admin Console.

2. Go to **Component Details** and click **AUTH** tab. The Client ID and Client Password details are displayed.
3. Copy Client ID and Client password.
4. Open Postman or relevant API tools. Select POST method and paste the URL: `https://ip:port/oauth2/v1/token`.

> **Note**
>
> Replace `ip:port` with tenant URL or domain name.

5. Click the **Authorization** tab. Go to the Type field and select **Basic Auth** from the drop-down list. The User name and Password fields are displayed.
6. Enter the User Name and Password that you have copied in step-3.
7. Go to the **Body** tab. Select request format as: `x-www-form-urlencoded`.
8. Enter the KEY and Value fields as mentioned in the subsequent table:

Table 2-1 Key and Value

KEY	VALUE
grant_type	client_credentials
scope	urn:opc:idm:__myscopes__

9. Click the **Header** tab. The Header details are displayed.
10. Enter the details explained in the subsequent table:

Table 2-2 Key and Value

KEY	VALUE
ofs_remote_user	OFS_SRV_ACCT
Content-Type	application/json
accept-language	en-US,en-U
authorization	Bearer <Access_token>
locale	en-US

11. Click **Send**. An Authorization token is generated in the Response body. For example, `<eyJ4NXQjUzI1NiI6Ikk3cWxndm1Kal...>`

> **Note**
>
> This authentication key is valid for a stipulated time.

12. Copy only the Authorization token details.

2.3 Supported Methods

Methods to invoke/execute Rest APIs.

- **GET**: Retrieve information about the service instance.
- **POST**: Create, scale, backup, start, and stop the service instance.

2.3.1 Media Types

The following media type is supported by the Oracle Transaction Filtering Cloud REST API:

- application/json

2.4 Supported Headers

Headers supported in the Rest APIs.

The REST API supports headers that may be passed in the header section of an HTTP Request or Response.

Table 2-3 Supported Headers

Headers	Description	Example
Content-Type	The media type of the body of the request. Required for POST and PUT requests, and the supported types vary with each endpoint.	Content-Type: application/json
Accept	The media type of the body of the response.	Accept: application/json

2.5 Status Code

Return Status Codes.

When you call the Accounting Foundation Cloud Service REST APIs Resources, the Response Header returns one of the standard HTTP Status Codes.

Table 2-4 Status Code

HTTP Status Code	Description
`200 OK`	The request was successfully completed. A 200 status is returned for a successful GET or POST Method.
`201 Created`	The request has been fulfilled and resulted in a new resource being created. The response includes a Location Header containing the canonical URI for the newly created resource. A 201 status is returned from a synchronous resource creation or an asynchronous resource creation that was completed before the response was returned.

Table 2-4 (Cont.) Status Code

HTTP Status Code	Description
202 Accepted	The request has been accepted for processing, but the processing has not been completed. The request may or may not eventually be acted upon, as it may be disallowed at the time the processing takes place. When specifying an Asynchronous (__detached=true) Resource creation (for example, when deploying an application), or update (for example, when redeploying an application), a 202 is returned if the operation is still in progress. If __detached=false, a 202 may be returned if the underlying operation does not complete in a reasonable amount of time.
400 Bad Request	The request could not be processed because it contains missing or invalid information (such as a validation error on an input field, a missing required value, and so on).
401 Unauthorized	The request is not authorized. The Authentication Credentials included with this request are missing or invalid.
403 Forbidden	The user cannot be authenticated. The user does not have the authorization to perform this request.
404 Not Found	The request includes a resource URI that does not exist.
405 Method Not Allowed	The HTTP verb specified in the request (DELETE, GET, POST, PUT) is not supported for this request URI.
406 Not Acceptable	The resource identified by this request is not capable of generating a representation corresponding to one of the media types in the Accept Header of the request. For example, the client's Accept Header request XML be returned, but the resource can only return JSON.
409 Conflict	The client's ContentType Header is not correct (for example, the client attempts to send the request in XML, but the resource can only accept JSON).
415 Not Acceptable	The client's ContentType Header is not correct (for example, the client attempts to send the request in XML, but the resource can only accept JSON).
500 Internal Server Error	The server encountered an unexpected condition that prevented it from fulfilling the request.
503 Service Unavailable	The server is unable to handle the request due to temporary overloading or maintenance of the server. The REST Web Application is not currently running.

3 Posting a Real-Time Transaction SWIFT Message

Follow the sebsequent topics for posting a Real-Time Transaction SWIFT message:

- End Point Details
- Executing TF Real-time Request

3.1 End Point Details

End point details to send a request using the POST method.

Send a request using the POST method. The request must be in the following format.

- **HTTP Link Synchronous**- `https://"Host Name"/"Namespace name"/swift-transaction-service/sync/process`
- **HTTP Link Asynchronous**- `https://"Host Name"/"Namespace name"/swift-transaction-service/async/process`
- **Method** – POST
- **Content-Type** - Application/JSON

3.2 Executing TF Real-time Request

Procedure to execute the TF Real-time request.

To execute TF Real-time Request, follow the subsequent steps:

1. Go to Body Tab, select Request Form as **raw** and Request Body Format as **JSON** and paste the below script in Request Body:

```
{
    "rawMessage":
"{1:F01ABCDEFGHXXXX0000000000}{2:I101ESSESESSAXXXU3003}{3:{113:XXXX}}
{4:
:20:HMTWLTEST
:28D:1/1
:50F:/950800362384
1/XYZ
```

```
2/XYZ

3/XYZ

4/XYZ

:52A:UNLALV2X

:30:151105

:21:2015110500000001

:32B:EUR111,00

:57A:ABCDEFGH

FULLNWOTNCC

:59:/2411110000006600619015

Yury Igorevich MARTYNOV

Donetsk

Ukraine

:70:TESTY OBCIAZANIE RACH

W INNYM BANKU

:71A:SHA

-}

{5: {MAC:12345678}{CHK:123456789ABC}}",

"businessDomainCode": "GEN",

    "jurisdictionCode": "AMEA",

    "additionalData": {

    }}
```

Table 3-1 Request Parameters

Request Parameters	Descriptions
rawMessage	Specifies the format of the transaction sent by the bank for screening. This field is the primary input to the REST API.

Table 3-1 (Cont.) Request Parameters

Request Parameters	Descriptions
businessDomainCode	Indicates the code of the business domain from which the transaction originates. This field classifies records based on business types and restricts data access by linking user groups to specific domains. Users can only view records and cases within their assigned domains.
jurisdictionCode	Indicates the code of the jurisdiction from which the transaction was generated and sent for screening. This field classifies and limits data access based on organizational or geographical boundaries. Users can only access records and cases within jurisdictions assigned to their user group.
additionalData	(Optional) Represents a flexible object that can contain multiple data elements the bank wants to screen along with the input raw message. It may also be included in the response for easier identification by the bank.

2. Go to the Headers Tab and enter the details explained in the following table.

Table 3-2 Key and Value

KEY	Description	VALUE
ofs_remote_user	Used to authenticate or identify the remote user	OFS_SRV_ACCT
Content-Type	Specifies the media type of the resource sent by the client to the server.	application/json
accept-language	Specifies the preferred languages for the server response.	en-US,en-U
authorization	Used to provide authentication credentials. The Bearer token is a type of access token typically issued by an authentication server	Bearer <Access_token>

3. Click Send and check for the response in response body.

Sample Response:

```
 {

"status": "SUCCESS",

"statusMessage": "Message processed successfully. Matches found!",

"transactionToken": 618,

"additionalData": {},
```

```
    "feedbackData": {
        "messageReference": "HMTWLTEST",
        "finalScore": 100,
        "transactionReference": "2015110500000001",
        "caseId": "CA1181",
        "matchCount": 2,
        "matches": [
            {
                "matchedData": "Ukraine;Yury Igorevich MARTYNOV;Yury
Igorevich MARTYNOV;Donetsk",
                "blockName": "Text Block",
                "stdComments": "N/A",
                "watchlistType": "HMT",
                "identifierValue": "2015110500000001",
                "tagName": "59",
                "groupMsgID": 618,
                "score": 100,
                "userComment": "N/A",
                "matchedWatchlistId": "15139",
                "webServiceID": 1,
                "groupMsgType": "MT101",
                "inputString": "Yury Igorevich MARTYNOV\\r\\nDonetsk\\r\
\nUkraine",
                "identifierTagName": "21",
                "msgCategory": 1,
                "responseID": 6982
            },
            {
                "matchedData": "Yuriy Igorevich",
```

```
                    "blockName": "Text Block",
                    "stdComments": "N/A",
                    "watchlistType": "OFAC",
                    "identifierValue": "2015110500000001",
                    "tagName": "59",
                    "groupMsgID": 618,
                    "score": 64,
                    "userComment": "N/A",
                    "matchedWatchlistId": "38076",
                    "webServiceID": 1,
                    "groupMsgType": "MT101",
                    "inputString": "Yury Igorevich MARTYNOV\\r\\nDonetsk\\r\
\nUkraine",
                    "identifierTagName": "21",
                    "msgCategory": 1,
                    "responseID": 6983
                }
            ],
            "status": "HOLD"
        }
    }
```

Table 3-3 Response Parameters

Response Parameters	Description
First Status	Indicates the result of the API request. **Possible Values**: • SUCCESS: The request completed successfully. • FAILED: The request is not completed due to errors.
statusMessage	Describes the outcome of the message processing.
transactionToken	A unique group message ID generated by the application for each input raw message.
additionalData	A copy of the data provided in the request.
messageReference	Refers to the specific SWIFT message being processed, including its type (e.g., MT101, MT103) and header information.
finalScore	Indicates the match score, showing how closely the message matched records in the watchlist.
transactionReference	A 36-character unique identifier for a specific transaction or payment instruction.
caseId	A unique alphanumeric ID generated for each message that results in a match and requires case creation.
matchCount	Indicates the total number of hits for the current input raw message.
matchedData	Represents the message content that matched an entry in the watchlist.
blockName	Specifies the section or block of the message where the matched data was found.
watchlistType	Specifies the type of watchlist that was used for matching.
identifierValue	The actual value contained in that tag, which provides identifying information such as financial institution, country, and etc.
tagName	Identifies the tag from which the matched data was extracted.
groupMsgId	A unique identifier for a single message.
score	The score assigned to each match indicating its strength.
userComment	A comment field that allows the user to provide additional context.
matchedWatchlistId	Identifies the specific watchlist from which a match was found.
webServiceID	Indicates the ID of the watchlist service where the match was found.
groupMsgType	Specifies the SWIFT message format type.
inputString	The message string that was evaluated against the watchlist.
identifierTagName	Tag which Displays the unique value to identify a financial institution in a transaction.

Table 3-3 (Cont.) Response Parameters

Response Parameters	Description
`msgCategory`	Numerical code indicating the message type. For example, 1 represents SWIFT. Other types include 2 represents FEDWIRE, and 3 represents ISO.
`responseID`	A unique identifier generated for each match found in the current message. Referred to as the response ID.
`Second Status`	Indicates the result of message processing. This parameter value appears only when `First Status` is `SUCCESS`. **Possible Values**: • `HOLD`: Indicates the input message contains suspicious data that requires review by analysts. • `CLEAN`: Indicates the input message is clean and contains no suspicious data.

4

Posting a Real-Time Transaction Fedwire Message

Follow the sebsequent topics for posting a Real-Time Transaction Fedwire message:

- End Point Details
- Executing TF Real-time Request

4.1 End Point Details

End point details to send a request using the POST method.

Send a request using the POST method. The request must be in the following format.

- **HTTP Link Synchronous**- `https://"Host Name"/"Namespace name"/fedwire-transaction-service/sync/process`
- **HTTP Link Asynchronous**- `https://"Host Name"/"Namespace name"/fedwire-transaction-service/async/process`
- **Method** – POST
- **Content-Type** - Application/JSON

4.2 Executing TF Real-time Request

Procedure to execute the TF Real-time request.

To execute TF Real-time Request, follow the subsequent steps:

1. Go to Body Tab, select Request Form as **raw** and Request Body Format as **JSON** and paste the below script in Request Body:

```
{

    "rawMessage" :
"{1510}1002{3320}FEDWIRE{3500}REPALL{3600}BTR{4000}B6437804*YOU{4320}pigs",

    "jurisdictionCode":"AMEA",

    "businessDomainCode":"GEN",

    "additionalData" : {

                        "extKey1":"extVal01",

                        "extKey2":"extVal178"
```

```
}
```

Table 4-1 Request Parameters

Request Parameters	Descriptions
`rawMessage`	Specifies the format of the transaction sent by the bank for screening. This field is the primary input to the REST API.
`businessDomainCode`	Indicates the code of the business domain from which the transaction originates. This field classifies records based on business types and restricts data access by linking user groups to specific domains. Users can only view records and cases within their assigned domains.
`jurisdictionCode`	Indicates the code of the jurisdiction from which the transaction was generated and sent for screening. This field classifies and limits data access based on organizational or geographical boundaries. Users can only access records and cases within jurisdictions assigned to their user group.
`additionalData`	(Optional) Represents a flexible object that can contain multiple data elements the bank wants to screen along with the input raw message. It may also be included in the response for easier identification by the bank.

2. Go to the Headers Tab and enter the details explained in the following table.

Table 4-2 Key and Value

KEY	Description	VALUE
`ofs_remote_user`	Used to authenticate or identify the remote user.	OFS_SRV_ACCT
`Content-Type`	Specifies the media type of the resource sent by the client to the server.	application/json
`accept-language`	Specifies the preferred languages for the server response.	en-US,en-U
`authorization`	Used to provide authentication credentials. The Bearer token is a type of access token typically issued by an authentication server.	Bearer <Access_token>

3. Click Send and check for the response in response body.

Sample Response:

```
{

"status": "SUCCESS",
```

```
"statusMessage": "Message processed successfully. Matches found!",
"transactionToken": 341,
"additionalData": {
    "extKey1": "extVal01",
    "extKey2": "extVal178"
},
"feedbackData": {
    "messageReference": "",
    "finalScore": 100,
    "transactionReference": "",
    "caseId": "CA1182",
    "matchCount": 3,
    "matches": [
        {
            "matchedData": "YOU",
            "stdComments": "N/A",
            "watchlistType": "OFAC",
            "identifierValue": "1002",
            "tagName": "4000",
            "groupMsgID": 341,
            "score": 100,
            "userComment": "N/A",
            "matchedWatchlistId": "30968",
            "webServiceID": 4,
            "groupMsgType": "FDBTR1002",
            "inputString": "B6437804\\r\\nYOU",
            "identifierTagName": "1510",
            "msgCategory": 2,
```

```
        "responseID": 6984
    },
    {
        "matchedData": "YOU",
        "stdComments": "N/A",
        "watchlistType": "OFAC",
        "identifierValue": "1002",
        "tagName": "4000",
        "groupMsgID": 341,
        "score": 70,
        "userComment": "N/A",
        "matchedWatchlistId": "30968",
        "webServiceID": 1,
        "groupMsgType": "FDBTR1002",
        "inputString": "B6437804\\r\\nYOU",
        "identifierTagName": "1510",
        "msgCategory": 2,
        "responseID": 6985
    },
    {
        "matchedData": "REPALL",
        "stdComments": "N/A",
        "watchlistType": "GOODS",
        "identifierValue": "1002",
        "tagName": "3500",
        "groupMsgID": 341,
        "score": 100,
        "userComment": "N/A",
```

```
                "matchedWatchlistId": "1",
                "webServiceID": 4,
                "groupMsgType": "FDBTR1002",
                "inputString": "REPALL",
                "identifierTagName": "1510",
                "msgCategory": 2,
                "responseID": 6986
            }
        ],
        "status": "HOLD"
    }
}
```

Table 4-3 Response Parameters

Response Parameters	Description
First Status	Indicates the result of the API request. **Possible Values**: • SUCCESS: The request completed successfully. • FAILED: The request is not completed due to errors.
statusMessage	Describes the outcome of the message processing.
transactionToken	A unique group message ID generated by the application for each input raw message.
additionalData	A copy of the data provided in the request.
messageReference	Refers to the specific FEDWIRE message being processed, including its proprietary message type and associated header information.
finalScore	Indicates the match score, showing how closely the message matched records in the watchlist.
transactionReference	A 36-character unique identifier for a specific transaction or payment instruction.
caseId	A unique alphanumeric ID generated for each message that results in a match and requires case creation.
matchCount	Indicates the total number of hits for the current input raw message.

Table 4-3 (Cont.) Response Parameters

Response Parameters	Description
matchedData	Represents the message content that matched an entry in the watchlist.
stdComments	Contains standard comments related to the provided data.
watchlistType	Specifies the type of watchlist that was used for matching.
identifierValue	The actual value contained in that tag, which provides identifying information such as financial institution, country, and etc.
tagName	Identifies the tag from which the matched data was extracted.
groupMsgId	A unique identifier for a single message.
score	The score assigned to each match indicating its strength.
userComment	A comment field that allows the user to provide additional context.
matchedWatchlistId	Identifies the specific watchlist from which a match was found.
webServiceID	Indicates the ID of the watchlist service where the match was found.
groupMsgType	Specifies the FEDWIRE message format type.
inputString	The message string that was evaluated against the watchlist.
identifierTagName	Tag which Displays the unique value to identify a financial institution in a transaction.
msgCategory	Numerical code indicating the message type. For example, 2 represents FEDWIRE. Other types include 1 represents SWIFT, and 3 represents ISO.
responseID	A unique identifier generated for each match found in the current message. Referred to as the response ID.
Second Status	Indicates the result of message processing. This parameter value appears only when First Status is SUCCESS. **Possible Values**: • HOLD: Indicates the input message contains suspicious data that requires review by analysts. • CLEAN: Indicates the input message is clean and contains no suspicious data.

5

Posting a Real-Time Transaction ISO Message

Follow the sebsequent topics for posting a Real-Time Transaction ISO message:

- End Point Details
- Executing TF Real-time Request

5.1 End Point Details

End point details to send a request using the POST method.

Send a request using the POST method. The request must be in the following format.

- **HTTP Link Synchronous**- `https://"Host Name"/"Namespace name"/iso20022-transaction-service/sync/process`
- **HTTP Link Asynchronous**- `https://"Host Name"/"Namespace name"/iso20022-transaction-service/async/process`
- **Method** – POST
- **Content-Type** - Application/JSON

5.2 Executing TF Real-time Request

Procedure to execute the TF Real-time request.

To execute TF Real-time Request, follow the subsequent steps:

1. Go to Body Tab, select Request Form as **raw** and Request Body Format as **JSON** and paste the below script in Request Body:

```
{

    "rawMessage" : "<Document
xmlns=\"urn:iso:std:iso:20022:tech:xsd:sct:pacs.008.001.02\"
xsi:schemaLocation=\"urn:iso:std:iso:20022:tech:xsd:sct:pacs.008.001.02
schema.xsd\" xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\">

  <FIToFICstmrCdtTrf>

    <GrpHdr>

      <MsgId>LKJOIUY0XXX20210112010004000001</MsgId>

      <CreDtTm>2021-01-12T15:15:28</CreDtTm>

      <NbOfTxs>1</NbOfTxs>
```

```
<TtlIntrBkSttlmAmt Ccy=\"EUR\">20000.00</TtlIntrBkSttlmAmt>
<IntrBkSttlmDt>2021-01-13</IntrBkSttlmDt>
<SttlmInf>
  <SttlmMtd>CLRG</SttlmMtd>
  <ClrSys>
    <Prtry>SCL</Prtry>
  </ClrSys>
</SttlmInf>
<InstgAgt>
  <FinInstnId>
    <BIC>INKKRUM1</BIC>
  </FinInstnId>
</InstgAgt>
</GrpHdr>
<CdtTrfTxInf>
  <PmtId>
    <InstrId>PE15E9THW2</InstrId>
    <EndToEndId>16864318-3050015655</EndToEndId>
    <TxId>EZ/SGP/101120007</TxId>
  </PmtId>
  <PmtTpInf>
    <SvcLvl>
      <Cd>SEPA</Cd>
    </SvcLvl>
    <CtgyPurp>
      <Cd>SUPP</Cd>
    </CtgyPurp>
  </PmtTpInf>
```

```
        <IntrBkSttlmAmt Ccy=\"EUR\">10000.00</IntrBkSttlmAmt>
        <ChrgBr>SLEV</ChrgBr>
        <UltmtDbtr>
            <Nm>Pedro</Nm>
  <Id>
<PrvtId>
<DtAndPlcOfBirth>
<BirthDt>1963</BirthDt>
<CtryOfBirth>Mexico</CtryOfBirth>
</DtAndPlcOfBirth>
</PrvtId>
          </Id>
       </UltmtDbtr>
        <Dbtr>
        <Nm>GHASHGHAVI     Hamed</Nm>
        <PstlAdr>
          <Ctry>NL</Ctry>
        </PstlAdr>
        <Id>
         <PrvtId>
            <DtAndPlcOfBirth>
            <BirthDt>1989-06-30</BirthDt>
<CityOfBirth>Chennai</CityOfBirth>
<CtryOfBirth>IN</CtryOfBirth>
            </DtAndPlcOfBirth>
          </PrvtId>


        </Id>
```

```
</Dbtr>
 <DbtrAcct>
   <Id>
     <IBAN>IR231</IBAN>
   </Id>
 </DbtrAcct>
 <DbtrAgt>
   <FinInstnId>
     <BIC>INKKRUM1XXX</BIC>
   </FinInstnId>
 </DbtrAgt>
 <CdtrAgt>
   <FinInstnId>
     <BIC>IRANRUM1</BIC>
   </FinInstnId>
 </CdtrAgt>
 <Cdtr>
   <Nm>ENA LGUNLUG ARDELAMANCHA DECU YONO</Nm>
   <PstlAdr>
     <AdrLine>FELVOIEJVKSJ 26-45</AdrLine>
     <AdrLine>13245 AFNLFEIVNOEN</AdrLine>
   </PstlAdr>
 </Cdtr>
 <CdtrAcct>
   <Id>
     <IBAN>GB29NWBK64816516465461</IBAN>
   </Id>
 </CdtrAcct>
```

```
      <RmtInf>
        <Ustrd>On Behalf of:Jhond Oecompanyss,Ref:TESTING</Ustrd>
      </RmtInf>
    </CdtTrfTxInf>
  </FIToFICstmrCdtTrf>
</Document>",
    "businessDomainCode":"GEN",
    "jurisdictionCode":"AMEA",
    "messageDirection":"INBOUND",
    "additionalData" : {
    }
}
```

Table 5-1 Request Parameters

Request Parameters	Descriptions
rawMessage	Specifies the format of the transaction sent by the bank for screening. This field is the primary input to the REST API.
businessDomainCode	Indicates the code of the business domain from which the transaction originates. This field classifies records based on business types and restricts data access by linking user groups to specific domains. Users can only view records and cases within their assigned domains.
jurisdictionCode	Indicates the code of the jurisdiction from which the transaction was generated and sent for screening. This field classifies and limits data access based on organizational or geographical boundaries. Users can only access records and cases within jurisdictions assigned to their user group.
additionalData	(Optional) Represents a flexible object that can contain multiple data elements the bank wants to screen along with the input raw message. It may also be included in the response for easier identification by the bank.

Table 5-1 (Cont.) Request Parameters

Request Parameters	Descriptions
messageDirection	(Mandatory) This field indicates the direction of the ISO message. Users must enter either INBOUND or OUTBOUND. • INBOUND: A message coming into your system from an external source. • OUTBOUND: A message going out of your system to an external party.

2. Go to the Headers Tab and enter the details explained in the following table.

Table 5-2 Key and Value

KEY	Description	VALUE
ofs_remote_user	Used to authenticate or identify the remote user.	OFS_SRV_ACCT
Content-Type	Specifies the media type of the resource sent by the client to the server.	application/json
accept-language	Specifies the preferred languages for the server response.	en-US,en-U
authorization	Used to provide authentication credentials. The Bearer token is a type of access token typically issued by an authentication server.	Bearer <Access_token>

3. Click Send and check for the response in response body.

Sample SUCCESS Response:


```
{
    "status": "SUCCESS",
    "statusMessage": "Message processed successfully. Matches found!",
    "transactionToken": 121,
    "additionalData": {
        "MESSAGE_DIRECTION": "INBOUND"
    },
    "feedbackData": {
        "messageReference": "",
        "finalScore": 100,
```

```
                "transactionReference": "",
                "caseId": "CA1183",
                "matchCount": 9,
                "matches": [
                    {
                        "score": 75,
                        "userComment": "N/A",
                        "matchedData": "Pedro",
                        "matchedWatchlistId": "21585",
                        "webServiceID": 1,
                        "stdComments": "N/A",
                        "watchlistType": "OFAC",
                        "identifierValue": "PE15E9THW2",
                        "inputString": "Pedro",
                        "identifierTagName": "Document/FIToFICstmrCdtTrf/
CdtTrfTxInf[1]/UltmtDbtr[1]/Nm",
                        "groupMsgID": 121,
                        "responseID": 6991
                    },
                    {
                        "score": 75,
                        "userComment": "N/A",
                        "matchedData": "Pedro",
                        "matchedWatchlistId": "41657",
                        "webServiceID": 1,
                        "stdComments": "N/A",
                        "watchlistType": "OFAC",
                        "identifierValue": "PE15E9THW2",
                        "inputString": "Pedro",
```

```
                "identifierTagName": "Document/FIToFICstmrCdtTrf/
CdtTrfTxInf[1]/UltmtDbtr[1]/Nm",

                "groupMsgID": 121,

                "responseID": 6989

            },

            {

                "score": 73,

                "userComment": "N/A",

                "matchedData": "Petro;Petr Anatolyevich",

                "matchedWatchlistId": "16715",

                "webServiceID": 1,

                "stdComments": "N/A",

                "watchlistType": "OFAC",

                "identifierValue": "PE15E9THW2",

                "inputString": "Pedro",

                "identifierTagName": "Document/FIToFICstmrCdtTrf/
CdtTrfTxInf[1]/UltmtDbtr[1]/Nm",

                "groupMsgID": 121,

                "responseID": 6987

            },

            {

                "score": 80,

                "userComment": "N/A",

                "matchedData": "GHASHGHAVI Hamed;1989",

                "matchedWatchlistId": "26387",

                "webServiceID": 1,

                "stdComments": "N/A",

                "watchlistType": "OFAC",

                "identifierValue": "PE15E9THW2",
```

```
                "inputString": "GHASHGHAVI     Hamed\\r\\nChennai\\r\
\nNL,IN",

                "identifierTagName": "Document/FIToFICstmrCdtTrf/
CdtTrfTxInf[1]/Dbtr[1]/Nm,Document/FIToFICstmrCdtTrf/CdtTrfTxInf[1]/
Dbtr[1]/Id/PrvtId/DtAndPlcOfBirth/CityOfBirth,Document/FIToFICstmrCdtTrf/
CdtTrfTxInf[1]/Dbtr[1]/PstlAdr/Ctry,Document/FIToFICstmrCdtTrf/
CdtTrfTxInf[1]/Dbtr[1]/Id/PrvtId/DtAndPlcOfBirth/CtryOfBirth,Document/
FIToFICstmrCdtTrf/CdtTrfTxInf[1]/Dbtr[1]/Id/PrvtId/DtAndPlcOfBirth/
BirthDt",

                "groupMsgID": 121,

                "responseID": 6988

            },

            {

                "score": 73,

                "userComment": "N/A",

                "matchedData": "Pyotr;Petro",

                "matchedWatchlistId": "22411",

                "webServiceID": 1,

                "stdComments": "N/A",

                "watchlistType": "OFAC",

                "identifierValue": "PE15E9THW2",

                "inputString": "Pedro",

                "identifierTagName": "Document/FIToFICstmrCdtTrf/
CdtTrfTxInf[1]/UltmtDbtr[1]/Nm",

                "groupMsgID": 121,

                "responseID": 6990

            },

            {

                "score": 100,

                "userComment": "N/A",

                "matchedData": "IR;IR",

                "matchedWatchlistId": "2",
```

```
                    "webServiceID": 3,

                    "stdComments": "N/A",

                    "watchlistType": "COUNTRY",

                    "identifierValue": "PE15E9THW2",

                    "inputString": "IRANRUM1",

                    "identifierTagName": "Document/FIToFICstmrCdtTrf/
CdtTrfTxInf[1]/CdtrAgt/FinInstnId[1]/BIC",

                    "groupMsgID": 121,

                    "responseID": 6994

                },

                {

                    "score": 91,

                    "userComment": "N/A",

                    "matchedData": "IRQ;IRQ",

                    "matchedWatchlistId": "1",

                    "webServiceID": 3,

                    "stdComments": "N/A",

                    "watchlistType": "COUNTRY",

                    "identifierValue": "PE15E9THW2",

                    "inputString": "IR231",

                    "identifierTagName": "Document/FIToFICstmrCdtTrf/
CdtTrfTxInf[1]/DbtrAcct/Id[1]/IBAN",

                    "groupMsgID": 121,

                    "responseID": 6995

                },

                {

                    "score": 75,

                    "userComment": "N/A",

                    "matchedData": "CHENARI;1984",
```

```
                "matchedWatchlistId": "27159",

                "webServiceID": 1,

                "stdComments": "N/A",

                "watchlistType": "OFAC",

                "identifierValue": "PE15E9THW2",

                "inputString": "GHASHGHAVI     Hamed\\r\\nChennai\\r\
\nNL,IN",

                "identifierTagName": "Document/FIToFICstmrCdtTrf/
CdtTrfTxInf[1]/Dbtr[1]/Nm,Document/FIToFICstmrCdtTrf/CdtTrfTxInf[1]/
Dbtr[1]/Id/PrvtId/DtAndPlcOfBirth/CityOfBirth,Document/FIToFICstmrCdtTrf/
CdtTrfTxInf[1]/Dbtr[1]/PstlAdr/Ctry,Document/FIToFICstmrCdtTrf/
CdtTrfTxInf[1]/Dbtr[1]/Id/PrvtId/DtAndPlcOfBirth/CtryOfBirth,Document/
FIToFICstmrCdtTrf/CdtTrfTxInf[1]/Dbtr[1]/Id/PrvtId/DtAndPlcOfBirth/
BirthDt",

                "groupMsgID": 121,

                "responseID": 6992

            },

            {

                "score": 75,

                "userComment": "N/A",

                "matchedData": "Pedro",

                "matchedWatchlistId": "13361",

                "webServiceID": 1,

                "stdComments": "N/A",

                "watchlistType": "OFAC",

                "identifierValue": "PE15E9THW2",

                "inputString": "Pedro",

                "identifierTagName": "Document/FIToFICstmrCdtTrf/
CdtTrfTxInf[1]/UltmtDbtr[1]/Nm",

                "groupMsgID": 121,

                "responseID": 6993

            }
```

```
        ],

        "status": "HOLD"

    }

}
```

Sample CLEAN Response:

```
{
  "status": "SUCCESS",
  "statusMessage": "Message processed successfully.No Matches Found",
  "transactionToken": 161,
  "additionalData": {
    "MESSAGE_DIRECTION": "INBOUND"
  },
  "feedbackData": {
    "messageReference": "SEPA_2_BATCH_MULTI_TRXN",
    "finalScore": 0,
    "comments": "No Matches Found",
    "transactionReference": "trxn05",
    "matches": [],
    "status": "CLEAN"
  }
}
```

Table 5-3 Response Parameters

Response Parameters	Description
First Status	Indicates the result of the API request. **Possible Values**: • SUCCESS: The request completed successfully. • FAILED: The request is not completed due to errors.
statusMessage	Describes the outcome of the message processing.
transactionToken	A unique group message ID generated by the application for each input raw message.
additionalData	A copy of the data provided in the request.
messageReference	Refers to the specific ISO 20022 message being processed, including its message type and associated header information.
finalScore	Indicates the match score, showing how closely the message matched records in the watchlist.
transactionReference	A 36-character unique identifier for a specific transaction or payment instruction.

Table 5-3 (Cont.) Response Parameters

Response Parameters	Description
caseId	A unique alphanumeric ID generated for each message that results in a match and requires case creation.
matchCount	Indicates the total number of hits for the current input raw message.
score	The score assigned to each match indicating its strength.
userComment	A comment field that allows the user to provide additional context.
matchedData	Represents the message content that matched an entry in the watchlist.
matchedWatchlistId	Identifies the specific watchlist from which a match was found.
webServiceID	Indicates the ID of the watchlist service where the match was found.
stdComments	Contains standard comments related to the provided data.
watchlistType	Specifies the type of watchlist that was used for matching.
identifierValue	The actual value contained in that tag, which provides identifying information such as financial institution, country, and etc.
inputString	The message string that was evaluated against the watchlist.
identifierTagName	Tag which Displays the unique value to identify a financial institution in a transaction.
groupMsgId	A unique identifier for a single message.
responseID	A unique identifier generated for each match found in the current message. Referred to as the response ID.
Second Status	Indicates the result of message processing. This parameter value appears only when First Status is SUCCESS. **Possible Values**: • HOLD: Indicates the input message contains suspicious data that requires review by analysts. • CLEAN: Indicates the input message is clean and contains no suspicious data.

6

Using Real-Time TF Transaction Feedback

Feedback API is used to get the transaction feedback using the transaction token number.

You can get the feedback response for a particular transaction using the Feedback API. For synchronous API, the feedback response is displayed in the postman during the execution. For asynchronous API, only the transaction token is generated during the execution. Using this transaction token, you can create the feedback response at any instance using the GET method.

Follow the subsequent topics:

- End Point Details
- Executing TF Real-time Feedback Request

6.1 End Point Details

End point details to send a feedback request using GET method.

The request must be in the following format:

- **HTTP Link** - `https://"Host Name"/"Namespace name"/feedback-service/Feedback/findByID?messageType=1&transactionToken="Transaction Token Number"`
- **Method** – GET
- **Content-Type** - Application/JSON

6.2 Executing TF Real-time Feedback Request

Procedure to execute the TF Real-time Feedback request.

To execute TF Real-time Feedback Request, follow the subsequent steps:

1. Open Postman or a relevant tool.
2. Copy only the Authorization token that you got from the Authentication request.
3. Go to the Header tab.
4. Enter KEY as Authorization and Description as bearer (Authorization token) `<eyJ4NXQjUzI1NiI6Ikk3cWxndm1Kal...>. .`
5. Send the request using the GET method.

   Use the request ID generated while posting a Real-Time Transaction Message to send the request.

   You will get a response as Real-time TF Tranasaction Feedback for the particular Tranasaction Token ID.

Table 6-1 Key and Value

KEY	VALUE
messageType	1 (SWIFT) 2 (Fedwire) 3 (ISO)
transactionToken	Tranasaction Token ID

Sample Response:


```
 {
"transactionToken": "Tranasaction Token ID",
"score": 67,
"caseId": "CA660",
"feedbackMessage": {
    "finalScore": 67,
    "caseId": "CA660",
    "matchCount": 5,
    "matches": [
        {
            "matchedData": "S. M.;MOHAMMED;Sheik Mohammed;",
            "blockName": "Text Block",
            "stdComments": "N/A",
            "watchlistType": "HMT",
            "identifierValue": "Namecountrycity",
            "tagName": "50F[Format1]",
            "groupMsgID": 1,
            "score": 68,
            "userComment": "N/A",
            "matchedWatchlistId": "7517",
            "webServiceID": 1,
            "groupMsgType": "MT101",
            "inputString": "mohemmed",
            "identifierTagName": "20",
            "msgCategory": 1,
            "responseID": 12
        },
        {
            "matchedData":
"Mohammad;Mohammad;Mohammed;Mohammed;Mohammad Saber;Mohammad Saber;",
            "blockName": "Text Block",
            "stdComments": "N/A",
            "watchlistType": "HMT",
            "identifierValue": "Namecountrycity",
            "tagName": "50F[Format1]",
            "groupMsgID": 1,
            "score": 68,
            "userComment": "N/A",
            "matchedWatchlistId": "11933",
            "webServiceID": 1,
            "groupMsgType": "MT101",
            "inputString": "mohemmed",
            "identifierTagName": "20",
            "msgCategory": 1,
            "responseID": 13
```

```
                },
                {
                    "matchedData": ";Mohammed;Mohammed;MOHAMMAD AMAN;Mad Aman
Ustad;Mohammad Aman Ustad;",
                    "blockName": "Text Block",
                    "stdComments": "N/A",
                    "watchlistType": "HMT",
                    "identifierValue": "2015110500000001",
                    "tagName": "59",
                    "groupMsgID": 1,
                    "score": 68,
                    "userComment": "N/A",
                    "matchedWatchlistId": "12456",
                    "webServiceID": 1,
                    "groupMsgType": "MT101",
                    "inputString": "mohemmed",
                    "identifierTagName": "21",
                    "msgCategory": 1,
                    "responseID": 15
                },
                {
                    "matchedData":
"SHEHU;SHAYKU;SHEKAU;TAUHID;TAWHEED;SHEKAU;SHEKAU;MOHAMMED;MOHAMMED;",
                    "blockName": "Text Block",
                    "stdComments": "N/A",
                    "watchlistType": "HMT",
                    "identifierValue": "Namecountrycity",
                    "tagName": "50F[Format1]",
                    "groupMsgID": 1,
                    "score": 68,
                    "userComment": "N/A",
                    "matchedWatchlistId": "13006",
                    "webServiceID": 1,
                    "groupMsgType": "MT101",
                    "inputString": "mohemmed",
                    "identifierTagName": "20",
                    "msgCategory": 1,
                    "responseID": 14
                },
                {
                    "matchedData": "Ahmad;Mohammed;Nazir Ahmad;",
                    "blockName": "Text Block",
                    "stdComments": "N/A",
                    "watchlistType": "HMT",
                    "identifierValue": "2015110500000001",
                    "tagName": "59",
                    "groupMsgID": 1,
                    "score": 68,
                    "userComment": "N/A",
                    "matchedWatchlistId": "13753",
                    "webServiceID": 1,
                    "groupMsgType": "MT101",
                    "inputString": "mohemmed",
                    "identifierTagName": "21",
                    "msgCategory": 1,
                    "responseID": 16
```

```
            },
        ],
        "status": "HOLD"
    },
    "msgCategory": "Transaction Type",
    "status": "SUCCESS"
}
```

Table 6-2 Response Parameters

Response Parameters	Description
transactionToken	A unique group message ID generated by the application for each input raw message.
finalScore	Indicates the match score, showing how closely the message matched records in the watchlist.
caseId	A unique alphanumeric ID generated for each message that results in a match and requires case creation.
matchCount	Indicates the total number of hits for the current input raw message.
matchedData	Represents the message content that matched an entry in the watchlist.
blockName	Specifies the section or block of the message where the matched data was found.
watchlistType	Specifies the type of watchlist that was used for matching.
identifierValue	The actual value contained in that tag, which provides identifying information such as financial institution, country, and etc.
tagName	Identifies the tag from which the matched data was extracted.
groupMsgId	A unique identifier for a single message.
score	The score assigned to each match indicating its strength.
userComment	A comment field that allows the user to provide additional context.
matchedWatchlistId	Identifies the specific watchlist from which a match was found.
webServiceID	Indicates the ID of the watchlist service where the match was found.
groupMsgType	Specifies the SWIFT message format type.
inputString	The message string that was evaluated against the watchlist.
identifierTagName	Tag which Displays the unique value to identify a financial institution in a transaction.
msgCategory	Numerical code indicating the message type. For example, 1 represents SWIFT. Other types include 2 represents FEDWIRE, and 3 represents ISO.

Table 6-2 (Cont.) Response Parameters

Response Parameters	Description
`responseID`	A unique identifier generated for each match found in the current message. Referred to as the response ID.
`feedbackMessage status`	Indicates the current status of the alert in the case management system. This status reflects the alert's current stage in the analyst review process and is updated as the alert progresses through its review lifecycle. **Sample Possible Values**: • `HOLD`: Indicates the input message contains suspicious data that requires review by analysts and the transaction is on `HOLD` • `CLEAN`: Indicates the input message is clean and contains no suspicious data and the transaction is completed. • `BLOCKED`: Indicates the alert contains confirmed suspicious data. The transaction is `BLOCKED` from further processing. • `RELEASED`: Indicates the alert is approved for release by an analyst after it was on `HOLD` or under review.
`Status`	Indicates the result of the API request. **Possible Values**: • `SUCCESS`: The request completed successfully. • `FAILED`: The request is not completed due to errors.

7

Get TF Batch Feedback Download URL

The Feedback Download URL API is used to get the TF Batch feedback file download URL using the batch date, run number, and batch type.

Follow the sebsequent topics for the Get TF Batch Feedback Download URL:

- End Point Details
- Executing Get TF Batch Feedback Download URL Request

7.1 End Point Details

End point details to send a request using the GET method.

Send a request using the GET method. The request must be in the following format.

- **HTTP Link** - `https://"Host Name"/"Namespace name"/feedback-service/batch/getFeedbackParURL?runNumber=”RUN_NUMBER”&batchDate=”BATCH_DATE”&batchType=”BATCH_TYPE”`
- **Method** – GET
- **Content-Type** - Application/JSON

7.2 Executing Get TF Batch Feedback Download URL Request

Procedure to execute the TF Batch feedback download URL request.

To execute TF Batch feedback download URL Request, follow the subsequent steps:

1. Open Postman or a relevant tool.
2. Copy only the Authorization token that you got from the Authentication request.
3. Go to the Header tab.
4. Enter KEY as Authorization and Description as bearer (Authorization token)<`eyJ4NXQjUzI1NiI6Ikk3cWxndm1Kal...`> .
5. Send the request using the GET method.

   Use the batch date and run number used while executing the ISO20022BatchScreening/ NACHABatchScreening.

   You will get a response as Feedback file download URL with expiry date for the batch date and run number.

Table 7-1 Key and Value

KEY	VALUE
batchDate	Batch Date(MIS-DATE)
runNumber	Run Number
batchType	ISOBATCH/ACHBATCH

6. Click Send and check for the response in response body.

   Sample Response:


```
{
    "STATUS": "SUCCESS",
    "EXPIRY_TIME": "2024-11-21 07:12:20 UTC",
    "IS_EXPIRED": "false",
    "CREATED_TIME": "2024-11-06 07:12:20 UTC",
    "DOWNLOAD_URL": "https://objectstorage.us-phoenix-1.oraclecloud.com/p/
kaAdxKiWmfdtimwMW3f9gh1SB7kk84-Wjl3fZxc1tsDi_0oxhyds8un7UOtR3nKm/n/
oraclegbudevcorp/b/fsgbu_aml_cndevcorp_devregresspro-prd/o/
20241106_RUN3_ISO_FEEDBACK.zip",
    "FILENAME": "20241106_RUN3_ISO_FEEDBACK.zip"
}
```

8

Generate New TF Batch Feedback Download URL

Generate New Feedback Download URL API is used to get the new TF Batch feedback file download URL using the batch date, run number, and batch type when the existing download URL gets expired or any authorization issue.

Follow the sebsequent topics for the Generate New TF Batch Feedback Download URL:

- End Point Details
- Executing Generate New TF Batch Feedback Download URL Request

8.1 End Point Details

End point details to send a request using the POST method.

Send a request using the POST method. The request must be in the following format.

- **HTTP Link** - `https://"Host Name"/"Namespace name"/feedback-service/batch/uploadFeedbackToObjectStorage?runNumber="RUN_NUMBER"&batchDate="BATCH_DATE"&batchType="BATCH_TYPE"`
- **Method** – POST
- **Content-Type** - Application/JSON

8.2 Executing Generate New TF Batch Feedback Download URL Request

Procedure to execute the Generate new TF Batch feedback download URL request.

To execute generate new TF Batch feedback download URL Request, follow the subsequent steps:

1. Open Postman or a relevant tool.
2. Copy only the Authorization token that you got from the Authentication request.
3. Go to the Header tab.
4. Enter KEY as Authorization and Description as bearer (Authorization token)<eyJ4NXQjUzI1NiI6Ikk3cWxndm1Kal...>. .
5. Send the request using the POST method.

   Use the batch date and run number used while executing the ISO20022BatchScreening/NACHABatchScreening.

   You will get a response as new Feedback file download URL with expiry date for the batch date, run number, and batch type.

Table 8-1 Key and Value

KEY	VALUE
batchDate	Batch Date(MIS-DATE)
runNumber	Run Number
batchType	ISOBATCH/ACHBATCH

6. Click Send and check for the response in response body.

Sample Response:


```
{
    "STATUS": "SUCCESS",
    "EXPIRY_TIME": "2024-11-26 08:15:49 UTC",
    "IS_EXPIRED": "false",
    "CREATED_TIME": "2024-11-11 08:15:49 UTC",
    "DOWNLOAD_URL": "https://objectstorage.us-phoenix-1.oraclecloud.com/p/
49dRVmtptjrHpdmq1i--
    zlLxO_rGCkkXjVnbYNOPkS3z2vydysH1xyYHhUwAtxqw/n/oraclegbudevcorp/b/
fsgbu_aml_cndevcorp_devregresspro-prd/o/20241106_RUN3_ISO_FEEDBACK.zip",
    "FILENAME": "20241106_RUN3_ISO_FEEDBACK.zip"
}
```

9

API Specification Files

This section provides a list of Transaction Filter (TF) API specification files available for download. You can import these files into tools such as Swagger UI, Postman, or any OpenAPI-compatible platform to explore the API, modify parameters, and execute requests interactively.

You can download following API specification files from the MOS page:

- Authorization Token Generation API
- Swift Real Time Screening API
- Fedwire Real Time Screening API
- ISO20022 Real Time Screening API
- Get Feedback API