

Transfers and payments Data Model and Services 4.4

# Document Status

status: Request for comment. (valid values are < Request for Comment, Preliminary Review, Public Review, Architectural Review, Final Review, Published, Deprecated)

# Authors and Change Log

|  |  |  |
| --- | --- | --- |
| Version | Date | Changes |
| 0.0.1 |  | * Initial Creation |
| 3.0 | 12/20/2013 | * Document updated based on inheritance model discussed in Denver AC group meeting. |
| 3.0 | 04/09/2014 | * Updated JSON examples for restructuring of transfer base xsds. |
| 3.0 | 05/07/2014 | * Updated JSON examples |
| 3.1 | 07/17/2015 | * Updated to release 3.1 |
| 3.2 | 05/10/2016 | * Updated to release 3.2 |
| 3.3 | 02/15/2017 | * Updated to release 3.3 |
| 4.0 | 02/19/2017 | * Updated to release 4.0, Date Range Global Update, Microsoft Global bug fix, Added occurrenceType , routing, account type to FundsTransferOccurrenceBase. Added Routing to recurring transfer base. Added definition for AccountBase. Removed and normalized common:IntervalFrequencyType into common:FrequencyType. Added finalAmount to fundsTransferReccuringBase. Added recurringFromAccountList, recurringToAccountList, recurringStatusList, date range filters for all FundsTransferReccurringBase dates to filter. |
| 4.1 | 12/10/2018 | * Updated to release 4.1, * Global update applied for minOccurs 0 maxOccurs 1. Xsd had 1 or more elements corrected to be consistent with the standard definition. * FundsTransferOccurrenceBase, FundsTransferRecurringBase - Added Ocurrence / Recurring detail amount detail lists. |
| 4.2 | 03/05/2019 | * Updated to release 4.2, \*\*\* Release 4.2 is a breaking fix release. \*\*\* Errors found in App, ArtifactFilter, and BillFilter required a breaking fix to align with the standard and prevent additional implementation difficulties going forward. |
| 4.3 | 10/07/2019 | * Updated to release 4.3, renamed file removing version as proper version control is being used in Github. |
| 4.4 | 10/20/2020 | * Updated to release 4.4. FundsTransferFilterBase - Added accountIdentificationList and includeNotesFlag to filter. * FundsTransferOccurrenceBase - Added achStandardEntryClass * FundsTransferRecurringBase - Added achStandardEntryClass * TransferOccurrence - Added noteList, achPrenoteIndicator * TransferRecurring - Added noteList, achPrenoteIndicator |

# Overview of Specification

Transfers and payments data model specification describes the services used in OLB/IVR/Mobile for creating, updating, and modifying transfers and recurring transfers.

# Any know Errors in the document

|  |  |
| --- | --- |
| **Error Description** | Status of Error |
|  |  |

Table of Contents

[Document Status 1](#_Toc54093392)

[Authors and Change Log 1](#_Toc54093393)

[Overview of Specification 2](#_Toc54093394)

[Any know Errors in the document 2](#_Toc54093395)

[Document Conventions 3](#_Toc54093396)

[CUFX API and Documentation Support 3](#_Toc54093397)

[Release 4.0 Global Update Notes 3](#_Toc54093398)

[Release 4.4 Global Update Notes 4](#_Toc54093399)

[Definitions related to the specification 4](#_Toc54093400)

[High level use cases 4](#_Toc54093401)

[Use Case 1: A simple add a transfer 4](#_Toc54093402)

[Use Case 1: A simple add a recurring transfer 5](#_Toc54093403)

[Data Elements 5](#_Toc54093404)

[transfer Services 5](#_Toc54093405)

[TransferOccurrenceService 5](#_Toc54093406)

[TransferRecurringService 5](#_Toc54093407)

[TransferOccurrenceService Details 6](#_Toc54093408)

[Service Message: Read Transfer By Occurrence Id 6](#_Toc54093409)

[Service Message: Read Transfer By Account Id 7](#_Toc54093410)

[Service Message: Create Transfers 9](#_Toc54093411)

[Service Message: Update Transfer 11](#_Toc54093412)

[Service Message: Delete Transfers 12](#_Toc54093413)

[TransferRecurringService Details 13](#_Toc54093414)

[Service Message: Read Recurring Transfers 14](#_Toc54093415)

[Service Message: Create Recurring Transfers 15](#_Toc54093416)

[Service Message: Update Recurring Transfer 17](#_Toc54093417)

[Service Message: Delete Recurring Transfers 19](#_Toc54093418)

[General Error handling For All Services 20](#_Toc54093419)

[Bibliography 20](#_Toc54093420)

# Document Conventions

List any document conventions such as what bold and italics mean and how the document is intended to be read.

“Within this specification, the key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” are to be interpreted as described in W3 Working Group (W3C)]. However, for readability, these words do not appear in all uppercase letters in this specification.

At times, this specification recommends good practice for authors and user agents. These recommendations are not normative and conformance with this specification does not depend on their realization. These recommendations contain the expression “We recommend …”, “This specification recommends …”, or some similar wording.”

All formatting in this document utilize Word Styles.

All Citations must utilize Word Citations to automatically show at the end of the document.

All updates after the initial creation must be performed using Tracking Changes turn on and Accepted by the Architecture committee.

# CUFX API and Documentation Support

CUFX is published to SwaggerHub at <https://app.swaggerhub.com/apis/dlacroix01/CUFX> . The latest default version will load automatically.

The purpose of this publication is to demonstrate the full range of CUFX messaging. Provide for complete documentation of the entire message structure and example usage.

Swaggerhub as a normal support feature also translates into several dozen of the most common and in demand client / server formats. This provides a technology specific version of the CUFX standard for essentially all platforms.

The CUFX Specification documents have been written to provide for limited examples of case usage but do not reflect the entirety of elements present in any given message. Please see the latest version of CUFX in Swaggerhub for the complete message and for superior documentation support.

# Release 4.0 Global Update Notes

CUFX Release 4.0 introduces a number modifications that significantly improves the standard and is not backward compatible with prior versions.

Messaging paradigm shift. Prior to CUFX 4.0 a Message Object would be sent and would expect the Object List to be returned or the error message. The response had to be interrogated to determine what was received. With CUFX 4.0, the Object Message that is sent is also expected to be the Object that is returned. Significant improvements have been made to the Message Context to fully support Success, Informational, Warnings and Error responses. End Points may continue to use the prior methods, but use of the Error.xsd is depreciated; all functionality has transitioned into MessageContext.xsd.

Date Range Filtering. A global update was applied across the standard to remove the pairs of date filter elements for any given range and replaced with a single Common.xsd definition DateRange complex type. This makes date range filtering completely uniform across the standard and associates the startDateTime and endDateTime together as an object set.

As example: elements transactionStartDateTime and transactionEndDateTime were replaced in the AccountFilter.xsd with transactionDateRange.

Microsoft Serialization Bug. We discovered the root cause of a serialization error impacting CUFX. A known Microsoft Serialization error from 2006 is present for single element complex types. It causes a naming error of the serialized constructs. If both endpoints are using a Microsoft compilation the error is consistent and does not present itself, the names are both wrong but pass data successfully. When one end point is not using a Microsoft compilation, the field names are in variance and fails. If both end points are using non-Microsoft compilation the serialization would be correct and match.

CUFX 4.0 has applied a global update across all list types throughout the standard. The CUFX list construct was consistently a single element complex type. For all occurrences we have applied an extension base of common:ListBase. ListBase provides pagination support and also resolves the Microsoft serialization error. No longer being a single element complex type, Microsoft compilation now generates the correct names. This will necessitate prior (Microsoft) implementations to remap to the correct serialized names.

# Release 4.4 Global Update Notes

CUFX Release 4.4 introduces a significant enhancement for complex Account identification and filter navigation. The foundational architectural design premise for account navigation is that the CUFX AccountId would be a unique value unto itself within a given institution, or that a composite unique key would be passed. With the direct support of several core system providers it was established that that later case is predominate. The AccountId is generally not a unique value unless in combination with several other values such as AccountType and AccountSubType. Hence passing a unique AccountId meant that the organization had to overload the element value making filtering implementation specific and forcing the endpoints to map overlay the accountId to unpack the value.

CUFX now fully recognizes this architectural paradigm while continuing to support the original architecture.

A global update was applied to provide the elements accountType and accountSubType in all objects that contained accountId.

A new filter list – AccountIdentificationList has been added to all account related filters so that AccountId, AccountType, and AccountSubType can be structured properly for discreet filtering of complex account keys, support filtering by the sub keys and also support inbound and outbound account filtering using the accountToFromIndicator.

By expanding the architectural paradigm to support a non-unique AccountId CUFX is now positioned to better support core adoption of the standard.

# Definitions related to the specification

Transfer

Transfer is moving of funds from one account to another account, within or outside the financial institution.

Recurring Transfer

Recurring transfers are transfers between accounts that are for the same amount or between the same accounts at specified intervals, e.g., weekly, every 2 weeks, quarterly, etc.

# High level use cases

## Use Case 1: A simple add a transfer

CUFX OLB

1

Transfer API with TransferOccurrenceMessage

Contains all the details about the Transfer

CUFX Data Provider

2

TransferOccurrenceList Would return the TransferID that was just created

## Use Case 1: A simple add a recurring transfer

CUFX OLB

1

Transfer API with TransferRecurringMessage

Contains all the details about the Transfer

CUFX Data Provider

2

TransferRecurringList Would return the RecurringTransferID that was just created

# Data Elements

All CUFX fields related to a validation are defined in

* TransferOccurrence.xsd
* TransferOccurrenceMessage.xsd
* TranferrRecurring.xsd
* TransferRecurringMessage.xsd
* TransferFilter.xsd

Base objects

* FundsTransferFilterBase.xsd
* FundsTransferOccurrenceBase.xsd
* FundsTransferRecurringBase.xsd

# transfer Services

## TransferOccurrenceService

TransactionOccurrenceService offers following services.

1. Service to add, update and delete a single transfer
2. Service to read list of past transfers.

## TransferRecurringService

TransactionOccurrenceService offers following services.

1. Service to add, update and delete a recurring transfer
2. Service to Read, Update, Delete list of recurring transfers.

## TransferOccurrenceService Details

### Service Message: Read Transfer By Occurrence Id

When a TransferFilter is provided the TransferOccurrence service will return a list of transfer objects that match the filter. For example, the TransferFilter when containing a fromData and ToDate would return all the transfer objects for that time period.

|  |  |
| --- | --- |
| INPUTS | cufx: TransferOccurrenceMessage (which includes)   * cufx:MessageContext * cufx:TransferFilter (for read, update) * cufx:transferOccurrenceList (for create, update, delete) |
| Outputs | cufx: TransferOccurrenceMessage (which includes)   * cufx:MessageContext * cufx:transferOccurrenceList |
| Return Values | cufx: TransferOccurrenceMessage (which includes)   * cufx:MessageContext   + statusList |
| Side Effects | No data is being manipulated, only a lookup is being performed. |
| Dependencies | Security Services for authentication and security. |
| CUFX Rest Link | https://api.dataprovider.com/tranferoccurrencemessage |

REST-JSON read transfer

**REQUEST:**

Headers:

**<security related header parameters... see Security Services>**

Accept: application/json

Accept-Charset: utf-8

Accept-Language: en-us (IANA – language codes)(W3C, HTTP Protocols)

Content-type: application/json; charset=utf-8

**X**-**HTTP**-**Method**-**Override**: **GET**

X-API-Version: >=4.3.0

**POST** [**https://api.dataprovider.com/transferoccurrencemessage**](https://api.dataprovider.com/transferoccurrencemessage)

{

"transferOccurrenceMessage": {

"@xmlns:xsi": "http://www.w3.org/2001/XMLSchema-instance",

"messageContext": {

<See messageContext.xsd>

},

"transferFilter": {

"occurrenceIdList": {

"occurrenceId": [

"567654"

]

}

...

}

}

RESPONSE

Headers:

Status Code: 200 Ok

Content-type: application/json; charset=utf-8

Content-Language: en-us

Payload:

{

"transferOccurrenceMessage": {

"@xmlns:xsi": "http://www.w3.org/2001/XMLSchema-instance",

"messageContext": {

<See messageContext.xsd>

},

"transferOccurrenceList": {

"@xmlns:xsi": "http://www.w3.org/2001/XMLSchema-instance",

"transferOccurrence": [

{

"occurrenceId": "567654",

"recurringId": "recurringId1",

"occurrenceAmount": {

"value": "1",

"currencyCode": "USD"

},

"occurrenceFromAccountId": "occurrenceFromAccountId1",

"occurrenceToAccountId": "occurrenceToAccountId1",

"occurrenceStatus": "Scheduled",

"fundsWithdrawalType": "WhenRequested",

"occurrenceMemo": "occurrenceMemo1",

"estimatedProcessDateTime": "1900-01-01T01:01:01Z",

"createdDateTime": "1900-01-01T01:01:01Z",

"processingStartedDateTime": "1900-01-01T01:01:01Z",

"processedDateTime": "1900-01-01T01:01:01Z",

"completedDateTime": "1900-01-01T01:01:01Z",

"lastUpdatedDateTime": "1900-01-01T01:01:01Z",

"occurrenceConfirmationCode": "occurrenceConfirmationCode1",

"queuedForPosting": "true",

"note": "note1",

"batchId": "batchId1",

"customData": {

"valuePair": [

{

"name": "name1",

"value": "value1"

}

]

}

}

]

}

}

### Service Message: Read Transfer By Account Id

When a TransferFilter is provided the TransferOccurrence service will return a list of transfer objects that match the filter. For example, the TransferFilter when containing a fromData and ToDate would return all the transfer objects for that time period.

|  |  |
| --- | --- |
| INPUTS | cufx: TransferOccurrenceMessage (which includes)   * cufx:MessageContext * cufx:TransferFilter (for read, update) * cufx:transferOccurrenceList (for create, update, delete) |
| Outputs | cufx: TransferOccurrenceMessage (which includes)   * cufx:MessageContext * cufx:transferOccurrenceList |
| Return Values | cufx: TransferOccurrenceMessage (which includes)   * cufx:MessageContext   + statusList |
| Side Effects | No data is being manipulated, only a lookup is being performed. |
| Dependencies | Security Services for authentication and security. |
| CUFX Rest Link | https://api.dataprovider.com/tranferoccurrencemessage |

REST-JSON read transfer

**REQUEST:**

Headers:

**<security related header parameters... see Security Services>**

Accept: application/json

Accept-Charset: utf-8

Accept-Language: en-us (IANA – language codes)(W3C, HTTP Protocols)

Content-type: application/json; charset=utf-8

**X**-**HTTP**-**Method**-**Override**: **GET**

X-API-Version: >=4.0.0

**POST** [**https://api.dataprovider.com/transferoccurrence**](https://api.dataprovider.com/transferoccurrence)**message**

{

"transferOccurrenceMessage": {

"@xmlns:xsi": "http://www.w3.org/2001/XMLSchema-instance",

"messageContext": {

<See messageContext.xsd>

},

"transferFilter": {

"startCompletedDateTime": "1900-01-01T01:01:01Z",

"endCompletedDateTime": "1900-01-01T01:01:01Z",

"accountIDList": {

"accountId": [

"accountId1"

]

}

...

}

}

RESPONSE

Headers:

Status Code: 200 Ok

Content-type: application/json; charset=utf-8

Content-Language: en-us

Payload:

{

"transferOccurrenceMessage": {

"@xmlns:xsi": "http://www.w3.org/2001/XMLSchema-instance",

"messageContext": {

<See messageContext.xsd>

},

"transferOccurrenceList": {

"@xmlns:xsi": "http://www.w3.org/2001/XMLSchema-instance",

"transferOccurrence": [

{

"occurrenceId": "567654",

"recurringId": "recurringId1",

"occurrenceAmount": {

"value": "1",

"currencyCode": "USD"

},

"occurrenceFromAccountId": "occurrenceFromAccountId1",

"occurrenceToAccountId": "occurrenceToAccountId1",

"occurrenceStatus": "Scheduled",

"fundsWithdrawalType": "WhenRequested",

"occurrenceMemo": "occurrenceMemo1",

"estimatedProcessDateTime": "1900-01-01T01:01:01Z",

"createdDateTime": "1900-01-01T01:01:01Z",

"processingStartedDateTime": "1900-01-01T01:01:01Z",

"processedDateTime": "1900-01-01T01:01:01Z",

"completedDateTime": "1900-01-01T01:01:01Z",

"lastUpdatedDateTime": "1900-01-01T01:01:01Z",

"occurrenceConfirmationCode": "occurrenceConfirmationCode1",

"queuedForPosting": "true",

"note": "note1",

"batchId": "batchId1",

"customData": {

"valuePair": [

{

"name": "name1",

"value": "value1"

}

]

}

}

]

}

}

### Service Message: Create Transfers

The CreateTransfer service message will create the transfer record on the host system and respond with a list of TransferOccurrenceIdList.

|  |  |
| --- | --- |
| INPUTS | cufx: TransferOccurrenceMessage (which includes)   * cufx:MessageContext * cufx:TransferFilter (for read, update) * cufx:transferOccurrenceList (for create, update, delete) |
| Outputs | cufx: TransferOccurrenceMessage (which includes)   * cufx:MessageContext * cufx:transferOccurrenceList |
| Return Values | cufx: TransferOccurrenceMessage (which includes)   * cufx:MessageContext   + statusList |
| Side Effects | Creates a new transfer record on the core system if successful. |
| Dependencies |  |
| CUFX REST LINK | https://api.dataprovider.com/transferoccurrencemessage |

REST-JSON Example

**REQUEST:**

Headers:

**<Security related header parameters... see Security Services>**

Accept: application/json

Accept-Charset: utf-8

Accept-Language: en-us (IANA – language codes)(W3C, HTTP Protocols)

Content-type: application/json; charset=utf-8

X-API-Version: >=4.3.0

**POST** [**h**ttps://api.dataprovider.com/transferoccurrence](https://api.dataprovider.com/transferoccurrence)message

{

"transferOccurrenceMessage": {

"@xmlns:xsi": "http://www.w3.org/2001/XMLSchema-instance",

"messageContext": {

<See messageContext.xsd>

},

"transferOccurrenceList": {

"transferOccurrence": [

{

"occurrenceAmount": {

"value": "123.00",

"currencyCode": "USD"

},

"occurrenceFromAccountId": "occurrenceFromAccountId1",

"occurrenceToAccountId": "occurrenceToAccountId1",

"occurrenceStatus": "Scheduled",

"fundsWithdrawalType": "WhenRequested",

"occurrenceMemo": "occurrenceMemo1",

"estimatedProcessDateTime": "1900-01-01T01:01:01Z",

"createdDateTime": "1900-01-01T01:01:01Z",

"queuedForPosting": "true",

"note": "note1",

"batchId": "batchId1"

}

]

}

}

}

RESPONSE

Headers:

Status Code: 200 Ok

Content-type: application/json; charset=utf-8

Content-Language: en-us

Payload:

{

"transferOccurrenceMessage": {

"@xmlns:xsi": "http://www.w3.org/2001/XMLSchema-instance",

"messageContext": {

<See messageContext.xsd>

},

"transferOccurrenceList": {

"transferOccurrence": [

{

"occurrenceId": "occurrenceId1",

"occurrenceAmount": {

"value": "123.00",

"currencyCode": "USD"

},

"occurrenceFromAccountId": "occurrenceFromAccountId1",

"occurrenceToAccountId": "occurrenceToAccountId1",

"occurrenceStatus": "Scheduled",

"fundsWithdrawalType": "WhenRequested",

"occurrenceMemo": "occurrenceMemo1",

"estimatedProcessDateTime": "1900-01-01T01:01:01Z",

"createdDateTime": "1900-01-01T01:01:01Z",

"processingStartedDateTime": "1900-01-01T01:01:01Z",

"lastUpdatedDateTime": "1900-01-01T01:01:01Z",

"occurrenceConfirmationCode": "occurrenceConfirmationCode1",

"queuedForPosting": "true",

"note": "note1",

"batchId": "batchId1"

}

]

}

}

}

### Service Message: Update Transfer

The Update Transfer service message is used to update the upcoming transfers on the host system. A valid TransferOccurrenceMessage filter must be passed in that indicate the upcoming transfer or transfers to be updated.

|  |  |
| --- | --- |
| INPUTS | cufx: TransferOccurrenceMessage (which includes)   * cufx:MessageContext * cufx:TransferFilter (for read, update) * cufx:transferOccurrenceList (for create, update, delete) |
| Outputs | cufx: TransferOccurrenceMessage (which includes)   * cufx:MessageContext * cufx:transferOccurrenceList |
| Return Values | cufx: TransferOccurrenceMessage (which includes)   * cufx:MessageContext   + statusList |
| Side Effects | Updates the existing transfer record on the core system if successful. |
| Dependencies | The transfer should be a schedule transfer not past transfer. |
| CUFX REST LINK | https://api.dataprovider.com/transferoccurrencemessage |

REST-JSON Example

**REQUEST:**

Headers:

**<security related header parameters... see Security Services>**

Accept: application/json

Accept-Charset: utf-8

Accept-Language: en-us (IANA – language codes)(W3C, HTTP Protocols)

Content-type: application/json; charset=utf-8

X-API-Version: >=4.3.0

**PUT** [**h**ttps://api.dataprovider.com/transferoccurrence](https://api.dataprovider.com/transferoccurrence)message

{

"transferOccurrenceMessage": {

"@xmlns:xsi": "http://www.w3.org/2001/XMLSchema-instance",

"messageContext": {

<See messageContext.xsd>

},

"transferFilter": {

"occurrenceIdList": {

"occurrenceId": [

"567654"

]

},

"transferOccurrenceList": {

"transferOccurrence": [

{

"occurrenceId": "567654",

"occurrenceAmount": {

"value": "78.00",

"currencyCode": "USD"

},

"occurrenceFromAccountId": "occurrenceFromAccountId1",

"occurrenceToAccountId": "occurrenceToAccountId1",

"occurrenceStatus": "Scheduled",

"fundsWithdrawalType": "WhenRequested",

"occurrenceMemo": "occurrenceMemo1",

"estimatedProcessDateTime": "1900-01-01T01:01:01Z",

"createdDateTime": "1900-01-01T01:01:01Z",

"queuedForPosting": "true",

"note": "note1",

"batchId": "batchId1"

}

]

}

}

}

RESPONSE

Headers:

Status Code: 200 Ok

Content-type: application/json; charset=utf-8

Content-Language: en-us

Payload:

{

"transferOccurrenceMessage": {

"@xmlns:xsi": "http://www.w3.org/2001/XMLSchema-instance",

"messageContext": {

<See messageContext.xsd>

},

"transferOccurrenceList": {

"transferOccurrence": [

{

"occurrenceId": "567654",

"occurrenceAmount": {

"value": "78.00",

"currencyCode": "USD"

},

"occurrenceFromAccountId": "occurrenceFromAccountId1",

"occurrenceToAccountId": "occurrenceToAccountId1",

"occurrenceStatus": "Scheduled",

"fundsWithdrawalType": "WhenRequested",

"occurrenceMemo": "occurrenceMemo1",

"estimatedProcessDateTime": "1900-01-01T01:01:01Z",

"createdDateTime": "1900-01-01T01:01:01Z",

"processingStartedDateTime": "1900-01-01T01:01:01Z",

"lastUpdatedDateTime": "1900-01-01T01:01:01Z",

"occurrenceConfirmationCode": "occurrenceConfirmationCode1",

"queuedForPosting": "true",

"note": "note1",

"batchId": "batchId1"

}

]

}

}

}

### Service Message: Delete Transfers

When a TransferFilter is provided the TransferOccurrence service will delete a list of upcoming transfer objects that match the filter. For example, the TransferFilter when containing a fromData and ToDate would delete all the transfer objects for that time period.

|  |  |
| --- | --- |
| INPUTS | cufx: TransferOccurrenceMessage (which includes)   * cufx:MessageContext * cufx:TransferFilter (for read, update) * cufx:transferOccurrenceList (for create, update, delete) |
| Outputs | cufx: TransferOccurrenceMessage (which includes)   * cufx:MessageContext * cufx:transferOccurrenceList |
| Return Values | cufx: TransferOccurrenceMessage (which includes)   * cufx:MessageContext   + statusList |
| Side Effects | Transfer is deleted from the system of record. |
| Dependencies | Security Services for authentication and security. |
| CUFX Rest Link | https://api.dataprovider.com/tranferoccurrencemessage |

REST-JSON Delete transfer

**REQUEST:**

Headers:

**<security related header parameters... see Security Services>**

Accept: application/json

Accept-Charset: utf-8

Accept-Language: en-us (IANA – language codes)(W3C, HTTP Protocols)

Content-type: application/json; charset=utf-8

**X**-**HTTP**-**Method**-**Override**: **DELETE**

X-API-Version: >=4.3.0

**POST** [**https://api.dataprovider.com/transferoccurrence**](https://api.dataprovider.com/transferoccurrence)**message**

{

"transferOccurrenceMessage": {

"@xmlns:xsi": "http://www.w3.org/2001/XMLSchema-instance",

"messageContext": {

<See messageContext.xsd>

},

"transferFilter": {

"occurrenceIdList": {

"occurrenceId": [

"567654"

]

}

}

...

}

}

RESPONSE

Headers:

Status Code: 200 Ok

Content-type: application/json; charset=utf-8

Content-Language: en-us

Payload:

{

"transferOccurrenceMessage": {

"@xmlns:xsi": "http://www.w3.org/2001/XMLSchema-instance",

"messageContext": {

<See messageContext.xsd>

}

}

}

## TransferRecurringService Details

### Service Message: Read Recurring Transfers

When a TransferFilter is provided the TransferRecurring service will return a list of recurring transfer objects that match the filter. For example, the TransferFilter when containing a fromData and ToDate would return all the transfer objects for that time period.

|  |  |
| --- | --- |
| INPUTS | cufx: TransferRecurringMessage (which includes)   * cufx:MessageContext * cufx:TransferFilter (for read, update) * cufx:transferRecurringList (for create, update, delete) |
| Outputs | cufx: TransferRecurringMessage (which includes)   * cufx:MessageContext * cufx:transferRecurringList |
| Return Values | cufx: TransferRecurringMessage (which includes)   * cufx:MessageContext   + statusList |
| Side Effects | No data is being manipulated, only a lookup is being performed. |
| Dependencies | Security Services for authentication and security. |
| CUFX Rest Link | https://api.dataprovider.com/tranferrecurringmessage |

REST-JSON read Recurringtransfer

**REQUEST:**

Headers:

**<security related header parameters... see Security Services>**

Accept: application/json

Accept-Charset: utf-8

Accept-Language: en-us (IANA – language codes)(W3C, HTTP Protocols)

Content-type: application/json; charset=utf-8

**X**-**HTTP**-**Method**-**Override**: **GET**

X-API-Version: >=4.3.0

**POST https://api.dataprovider.com/transferrecurringmessage**

{

"transferRecurringMessage": {

"@xmlns:xsi": "http://www.w3.org/2001/XMLSchema-instance",

"messageContext": {

<See messageContext.xsd>

},

"transferFilter": {

"recurringIdList": {

"recurringId": [

"874456"

]

},

"accountIDList": {

"accountId": [

"accountId1"

]

},

"minAmount": {

"value": "1.00",

"currencyCode": "USD"

},

"maxAmount": {

"value": "500.00",

"currencyCode": "USD"

}

}

}

}

RESPONSE

Headers:

Status Code: 200 Ok

Content-type: application/json; charset=utf-8

Content-Language: en-us

Payload:

{

"transferRecurringMessage": {

"@xmlns:xsi": "http://www.w3.org/2001/XMLSchema-instance",

"messageContext": {

<See messageContext.xsd>

},

"transferRecurringList": {

"transferRecurring": [

{

"recurringId": "874456",

"recurringAmount": {

"value": "220.23",

"currencyCode": "USD"

},

"recurringFromAccountId": "recurringFromAccountId1",

"recurringToAccountId": "recurringToAccountId1",

"recurringStatus": "Requested",

"recurringMemo": "recurringMemo1",

"isElectronic": "true",

"createdByFiUserId": "createdByFiUserId1",

"frequency": "Monthly",

"totalPaymentCount": "1",

"paymentsLeft": "1",

"fundsWithdrawalType": "WhenRequested",

"createRequestDateTime": "1900-01-01T01:01:01Z",

"firstScheduledDateTime": "1900-01-01T01:01:01Z",

"lastModifiedDate": "1900-01-01T01:01:01Z",

"recurringConfirmationCode": "recurringConfirmationCode1",

"recurringOccurrenceIdList": {

"occurrenceId": [

"123456"

]

}

}

]

}

}

}

### Service Message: Create Recurring Transfers

The CreateRecurringTransfer service message will create the transfer record on the host system and respond with a list of TransferRecurringIdList.

|  |  |
| --- | --- |
| INPUTS | cufx: TransferRecurringMessage (which includes)   * cufx:MessageContext * cufx:TransferFilter (for read, update) * cufx:transferRecurringList (for create, update, delete) |
| Outputs | cufx: TransferRecurringMessage (which includes)   * cufx:MessageContext * cufx:transferRecurringList |
| Return Values | cufx: TransferRecurringMessage (which includes)   * cufx:MessageContext   + statusList |
| Side Effects | Creates a new transfer record on the core system if successful. |
| Dependencies |  |
| CUFX REST LINK | https://api.dataprovider.com/tranferrecurringmessage |

REST-JSON Example

**REQUEST:**

Headers:

**<Security related header parameters... see Security Services>**

Accept: application/json

Accept-Charset: utf-8

Accept-Language: en-us (IANA – language codes)(W3C, HTTP Protocols)

Content-type: application/json; charset=utf-8

X-API-Version: >=4.3.0

**POST h**ttps://api.dataprovider.com/transferrecurringmessage

{

"transferRecurringMessage": {

"@xmlns:xsi": "http://www.w3.org/2001/XMLSchema-instance",

"messageContext": {

<See messageContext.xsd>

},

"transferRecurringList": {

"transferRecurring": [

{

"recurringAmount": {

"value": "145.12",

"currencyCode": "USD"

},

"recurringFromAccountId": "recurringFromAccountId1",

"recurringToAccountId": "recurringToAccountId1",

"recurringStatus": "Requested",

"recurringMemo": "recurringMemo1",

"isElectronic": "true",

"isOpenEnded": "true",

"createdByFiUserId": "createdByFiUserId1",

"frequency": "OneTime",

"totalPaymentCount": "1",

"paymentsLeft": "14",

"fundsWithdrawalType": "WhenRequested",

"createRequestDateTime": "1900-01-01T01:01:01Z",

"firstScheduledDateTime": "1900-01-01T01:01:01Z",

"recurringOccurrenceIdList": {

"occurrenceId": [

"occurrenceId1"

]

}

}

]

}

}

}

RESPONSE

Headers:

Status Code: 200 Ok

Content-type: application/json; charset=utf-8

Content-Language: en-us

Payload:

{

"transferRecurringMessage": {

"@xmlns:xsi": "http://www.w3.org/2001/XMLSchema-instance",

"messageContext": {

<See messageContext.xsd>

},

"transferRecurringList": {

"transferRecurring": [

{

"recurringId": "recurringId1",

"recurringAmount": {

"value": "145.12",

"currencyCode": "USD"

},

"recurringFromAccountId": "recurringFromAccountId1",

"recurringToAccountId": "recurringToAccountId1",

"recurringStatus": "Requested",

"recurringMemo": "recurringMemo1",

"isElectronic": "true",

"isOpenEnded": "true",

"createdByFiUserId": "createdByFiUserId1",

"frequency": "Monthly",

"totalPaymentCount": "1",

"paymentsLeft": "14",

"fundsWithdrawalType": "WhenRequested",

"createRequestDateTime": "1900-01-01T01:01:01Z",

"firstScheduledDateTime": "1900-01-01T01:01:01Z",

"recurringConfirmationCode": "recurringConfirmationCode1",

"recurringOccurrenceIdList": {

"occurrenceId": [

"occurrenceId1"

]

}

}

]

}

}

### Service Message: Update Recurring Transfer

The Update Transfer service message is used to update the upcoming recurring transfers on the host system. A valid TransferRecurringMessage filter must be passed in that indicate the upcoming recurring transfers to be updated.

|  |  |
| --- | --- |
| INPUTS | cufx: TransferRecurringMessage (which includes)   * cufx:MessageContext * cufx:TransferFilter (for read, update) * cufx:transferRecurringList (for create, update, delete) |
| Outputs | cufx: TransferRecurringMessage (which includes)   * cufx:MessageContext * cufx:transferRecurringList |
| Return Values | cufx: TransferRecurringMessage (which includes)   * cufx:MessageContext   + statusList |
| Side Effects | Updates the existing transfer record on the core system if successful. |
| Dependencies | The transfer should be a schedule transfer not past transfer. |
| CUFX REST LINK | https://api.dataprovider.com/transferrecurringmessage |

REST-JSON Example

**REQUEST:**

Headers:

**<security related header parameters... see Security Services>**

Accept: application/json

Accept-Charset: utf-8

Accept-Language: en-us (IANA – language codes)(W3C, HTTP Protocols)

Content-type: application/json; charset=utf-8

X-API-Version: >=4.3.0

**PUT h**ttps://api.dataprovider.com/transferrecurringmessage

{

"transferRecurringMessage": {

"@xmlns:xsi": "http://www.w3.org/2001/XMLSchema-instance",

"messageContext": {

<See messageContext.xsd>

},

"transferFilter": {

"recurringIdList": {

"recurringId": [

"874456"

]

}

}

"transferRecurringList": {

"transferRecurring": [

{

"recurringId": "874456",

"recurringAmount": {

"value": "145.12",

"currencyCode": "USD"

},

"recurringFromAccountId": "recurringFromAccountId1",

"recurringToAccountId": "recurringToAccountId1",

"recurringStatus": "Requested",

"recurringMemo": "recurringMemo1",

"isElectronic": "true",

"isOpenEnded": "true",

"createdByFiUserId": "createdByFiUserId1",

"frequency": "OneTime",

"totalPaymentCount": "1",

"paymentsLeft": "6",

"fundsWithdrawalType": "WhenRequested",

"createRequestDateTime": "1900-01-01T01:01:01Z",

"firstScheduledDateTime": "1900-01-01T01:01:01Z",

"recurringOccurrenceIdList": {

"occurrenceId": [

"occurrenceId1"

]

}

}

]

}

}

}

RESPONSE

Headers:

Status Code: 200 Ok

Content-type: application/json; charset=utf-8

Content-Language: en-us

Payload:

{

"transferRecurringMessage": {

"@xmlns:xsi": "http://www.w3.org/2001/XMLSchema-instance",

"messageContext": {

<See messageContext.xsd>

},

"transferRecurringList": {

"transferRecurring": [

{

"recurringId": "recurringId1",

"recurringAmount": {

"value": "145.12",

"currencyCode": "USD"

},

"recurringFromAccountId": "recurringFromAccountId1",

"recurringToAccountId": "recurringToAccountId1",

"recurringStatus": "Requested",

"recurringMemo": "recurringMemo1",

"isElectronic": "true",

"isOpenEnded": "true",

"createdByFiUserId": "createdByFiUserId1",

"frequency": "Monthly",

"totalPaymentCount": "1",

"paymentsLeft": "6",

"fundsWithdrawalType": "WhenRequested",

"createRequestDateTime": "1900-01-01T01:01:01Z",

"firstScheduledDateTime": "1900-01-01T01:01:01Z",

"lastModifiedDate": "1900-01-01T01:01:01Z",

"recurringConfirmationCode": "recurringConfirmationCode1",

"recurringOccurrenceIdList": {

"occurrenceId": [

"occurrenceId1"

]

}

}

]

}

}

### Service Message: Delete Recurring Transfers

When a TransferFilter is provided the TransferOccurrence service will delete a list of upcoming transfer objects that match the filter. For example, the TransferFilter when containing a fromData and ToDate would delete all the transfer objects for that time period.

|  |  |
| --- | --- |
| INPUTS | cufx: TransferRecurringMessage (which includes)   * cufx:MessageContext * cufx:TransferFilter (for read, update) * cufx:transferRecurringList (for create, update, delete) |
| Outputs | cufx: TransferRecurringMessage (which includes)   * cufx:MessageContext * cufx:transferRecurringList |
| Return Values | cufx: TransferRecurringMessage (which includes)   * cufx:MessageContext   + statusList |
| Side Effects | Transfer is deleted from the system of record. |
| Dependencies | Security Services for authentication and security. |
| CUFX Rest Link | https://api.dataprovider.com/tranferrecurringmessage |

REST-JSON Delete transfer

**REQUEST:**

Headers:

**<security related header parameters... see Security Services>**

Accept: application/json

Accept-Charset: utf-8

Accept-Language: en-us (IANA – language codes)(W3C, HTTP Protocols)

Content-type: application/json; charset=utf-8

**X**-**HTTP**-**Method**-**Override**: **DELETE**

X-API-Version: >=4.3.0

**POST https://api.dataprovider.com/transferrecurringmessage**

{

"transferRecurringMessage": {

"@xmlns:xsi": "http://www.w3.org/2001/XMLSchema-instance",

"messageContext": {

<See messageContext.xsd>

},

"transferFilter": {

"recurringIdList": {

"recurringId": [

"874456"

]

}

}

}

}

RESPONSE

Headers:

Status Code: 200 Ok

Content-type: application/json; charset=utf-8

Content-Language: en-us

Payload:

{

"transferRecurringMessage": {

"@xmlns:xsi": "http://www.w3.org/2001/XMLSchema-instance",

"messageContext": {

<See messageContext.xsd>

}

}

# General Error handling For All Services

Refer to latest CUFX documentation *Error Mapping*.

# Bibliography

W3C. (n.d.). *Key words for use in RFCs to Indicate Requirement Levels [RFC2119].* Retrieved Sept. 8th, 2011, from W3C.