

Activity Data Model and Services 4.3

# 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 |
| 4.0 | **02/19/2018** | * Initial Creation |
| 4.1 | **12/10/2018** | * Updated to release 4.1, Added enum OnHold to ActivityStatus, Global update applied for minOccurs 0 maxOccurs 1. Xsd had 1 or more elements corrected to be consistent with the standard definition. |
| 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. |

# Overview of Specification

This specification defines the features to create, read, update or delete an activity.

# Any know Errors in the document

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

Table of Contents

[Document Status 1](#_Toc21343273)

[Authors and Change Log 1](#_Toc21343274)

[Overview of Specification 1](#_Toc21343275)

[Any know Errors in the document 1](#_Toc21343276)

[Document Conventions 2](#_Toc21343277)

[Release 4.0 Global Update Notes 2](#_Toc21343278)

[Definitions related to the specification 3](#_Toc21343279)

[Data Elements 3](#_Toc21343280)

[Filters Used When Accessing the Activity Data 3](#_Toc21343281)

[Investment Data Attributes 3](#_Toc21343282)

[Use Case 3](#_Toc21343283)

[Services 4](#_Toc21343284)

[Authentication 4](#_Toc21343285)

[Investment Resource Based Create, Read, Update, Delete investment Account data 4](#_Toc21343286)

[REST-JSON READ activty 4](#_Toc21343287)

[General Error handling For All Services 7](#_Toc21343288)

[Bibliography 7](#_Toc21343289)

# 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.

# 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.

# Definitions related to the specification

Activity

General purpose activity object to create, read, update and delete activity data.

# Data Elements

## Filters Used When Accessing the Activity Data

Refer to Security Services documentation to understand what may be contained the header and processed by security procedures. When accessing the data include **messageContext.xsd** so that the service can determine the scope of the request. Include any filter variables related to the request. See **activityFilter.xsd.**

The filters are used to filter based on associated activity id data. Combining filters can be used to get a response or affect specific activity data.

## Investment Data Attributes

All CUFX fields related to an investment are defined in **activity.xsd**.

Note: Fields not listed in the calling specification are not to be returned to the calling specification. i.e. If the field transaction type is not listed in the calling specification, then do not return the data field to alleviate issues with unexpected information and bloat of information being returned to light weight applications.

# Use Case

CUFX Compliant Application

1

ActivityMessage Request

2

ActivityMessage contains

messageContext and activityList

CUFX Core Data Provider

# Services

|  |  |
| --- | --- |
| Definition | Activity Message |
| Overview of Capabilities | Service to read, update, create or deleted activity data. |
| Dependencies | Security Services, messageContext |
| CUFX REST LINK | https://api.dataprovider.com/activitymessage |
| CUFX WSDL LINK |  |

## Authentication

See Security Services and messageContext for more detail.

## Investment Resource Based Create, Read, Update, Delete investment Account data

|  |  |
| --- | --- |
| INPUTS | cufx:activityMessage (which includes)   * [cufx:MessageContext](file:///\\files2\users\CMarjaniemi\Projects\CUFX\MessageContext.html) * Cufx:activitytFilter (for read, update) * cufx:actvityList (for create, update, delete) |
| Outputs | cufx: investmentMessage (which includes)   * [cufx:MessageContext](file:///\\files2\users\CMarjaniemi\Projects\CUFX\MessageContext.html) * cufx:activityList |
| Return Values | cufx: investmentMessage (which includes)   * [cufx:MessageContext](file:///\\files2\users\CMarjaniemi\Projects\CUFX\MessageContext.html)   + statuslist |
| Side Effects | Creation, update or deletion of an activity. |
| Dependencies | Security Services for authentication and security. |
| Fields used | Message Headers : See security services  messageContext: See messageContext.xsd  Filters: See ActivityFilter.xsd  Attributes: activityList : See Activity.xsd |
| Testing Procedures for Certification |  |

### REST-JSON READ activty

This example shows reading an activity record filtering on FI ID, Party Id, and Account Id. Note: Not all fields are listed for simplicity of the example.

REQUEST:

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

Accept: application/json

Accept-Charset: utf-8

Accept-Language: en-us

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

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

X-API-Version: >=4.3.0

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

{

"activityMessage": {

"messageContext": {

"cufxVersion": "4.3.0",

"requestId": "requestId1",

"vendorId": "vendorId1",

"appId": "appId1",

"fiId": "77393",

},

},

"activityFilter": {

"activityIdList": {

},

"activityStatusList": {

"activityStatus": {

"@xsi:nil": "true"

},

"partyIdList": {

"partyId": "47463"

},

"fiUserIdList": {

"fiUserId": "77393"

},

}

},

"accountIdList": {

"accountId": "663893"

}

}

}

}

}

RESPONSE

Headers:

Status Code: 200 Ok

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

Content-Language: en-us

Payload:

{

"activityMessage": {

"messageContext": {

"cufxVersion": "4.3.0",

"requestId": "requestId1",

"vendorId": "vendorId1",

"appId": "appId1",

"fiId": "77393",

"dataSourceId": "dataSourceId1",

"environment": "Development",

"returnDataFilter": "All",

"includeBlankFields": "true",

"includeZeroNumerics": "true",

"statusList": {

"statusList": {

"status": {

"statusType": "Success"

}

},

},

"activityList": {

"activity": {

"activityId": {

"activityUniqueId": "activityUniqueId1"

},

"previousActivityId": "previousActivityId1",

"externalActivityId": "externalActivityId1",

"activityRelatedTo": null,

"activityStatus": "New",

"activityType": "activityType1",

"activityName": "activityName1",

"activityShortDescription": "activityShortDescription1",

"activityDescription": "activityDescription1",

"activityFieldImpactedList": {

"activityFieldImpacted": {

"activityFieldId": "activityFieldId1",

"activityFieldName": "activityFieldName1",

"activityFieldDataType": "activityFieldDataType1",

"activityFieldPreviousValue": "activityFieldPreviousValue1",

"activityFieldNewValue": "activityFieldNewValue1"

}

},

"creator": "creator1",

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

},

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

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

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

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

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

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

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

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

"activityIpAddress": "activityIpAddress1",

"activityScheduled": "true",

"activityCustomData": {

"valuePair": null

}

}

}

}

}

# 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.