

MessageContext Object 3.0

# Document Status

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

This version: **Assembla**.com. Files Tag = CUFX\_3.0\_RFC\_Active

Previous Version: **Assembla**.com. Files Tag = CUFX\_3.0\_RFC\_Archive

# Change Log

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Changes** |
| **0.0.01** |  | * Initial Creation |
| **0.0.02** |  | * Clarified that JSON assumes complex types within a list. Thus why user isn’t listed in example code. |
| **0.0.03** |  | * Updated to use data definitions from messageContext.xsd. * Updated UserType to Pascal case in examples |
| **0.0.04** |  | * Updated Overview of Specification |
| **0.0.05** |  | * Added use of customData example. |
| **0.0.06** |  | * Adding field to identify to return all, only updated fields or no data in the create and update |
| **3.0** | **12/16/2013** | * Versioning and format change with release CUFX 3.0 |

# Overview of Specification

The CUFX Message Context Data object specification describes the object that can be included in all data requests to aid in the applying any additional discovery of who/what made the request. This is not provided for security purposes. See Security Services for more detail on how this fits into security processes.

# Known Errors in the document

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

Table of Contents

[Document Status 1](#_Toc336963237)

[Change Log 1](#_Toc336963238)

[Overview of Specification 1](#_Toc336963239)

[Known Errors in the document 1](#_Toc336963240)

[Document Conventions 3](#_Toc336963241)

[Definitions related to the specification 3](#_Toc336963242)

[messageContext Data Elements 3](#_Toc336963243)

[REST-JSON Example 1: 4](#_Toc336963244)

[REST-JSON Example 2: 5](#_Toc336963245)

# Document Conventions

“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 with Track Changes **On** and Accepted by the Architecture Committee.

Any Quotes in code examples below should typically be coded as ASCII character decimal 034 or 022 hex.

# Definitions related to the specification

See each use object in the CUFX spec on how to include in the individual messages. See Security Services for overall conceptual use. See messageContext.xsd and messageContext.html for definitions of data elements.

## REST-JSON Example 1:

*....get rates message...*

“messageContext”:{

“vendorId”:”Facebook, Inc.”

“appId”:”Facebook finance plug-in widget”,

“fiId”:”Your Favorite Financial Institution”,

“dataSourceId”:”Rates Database”, “userIdList”:[ //\*\*\* note JSON assumes complex types within an array.. thus why user isn’t listed \*//

{

“userIdType”:”Anonymous”

}

]

}

*....get rates message...*

## REST-JSON Example 2:

*....update account info message...*

“messageContext”:{

“vendorId”:”Widget, Inc.”

“appId”:”Solutions 1 Online Banking”,

“fiId”:”Your Favorite Financial Institution”,

“dataSourceId”:”Secure financial data source”,

“returnDataFilter”:”All”,

“userIdList”:[ //\*\*\* note JSON assumes complex types within an array.. thus why user isn’t listed \*// {

“userIdType”:” FIUserId”,

“userID”:”jSmith123”

}

]

“customData”:[ //\*\*\* using custom data to pass user specific password or similar to source provider \*//

{

“name”:” userKeyToken”,

“value”:”#$f01hgbn%rthvc5”

}

]

}

*...get account info message...*