

Product offering Data and Services 4.1

# 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\_4.1\_RFC\_Active

Previous Version: **Assembla**.com. Files Tag = CUFX\_4.0\_RFC\_Archive

# Change Log

|  |  |  |
| --- | --- | --- |
| Version | Date | Changes |
| 0.0.01 |  | * Initial Release |
| 0.0.02 |  | * Cleaned up examples and verbiage |
| 0.0.03 |  | * Cleaned up typos, formatting etc. |
| 0.0.04 |  | * Update id lists to be consistent with XSD’s. * Fixed interest date filter. |
| 3.0 | **10/29/2013** | * Switch to use X-HTTP-METHOD-OVERRIDE standard rather than subMethod non-Standard method for overriding request types. * Create a productOfferingMessage wrapper for every message to increase ability for infrastructure to serialize the data * Versioning and format with release CUFX 3.0 |
| 3.0 | **12/16/2013** | * Update examples X-API-Version to >=3.0.0 |
| 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 | **02/19/2018** | * Updated to release 4.0, Date Range Global Update, Microsoft Global bug fix, Added interestRateEffective and interestRateExpiration DateRange filters. |
| 4.1 | **12/10/2018** | * Updated to release 4.1, ProductOffering - Corrected productId, productType, and productSubType to minOccurs 0. InterestRateList set to minOccurs="0" |

# Overview of Specification

Defines the ProductOffering data object for use by all specifications. ProductOffering reference information is used by a financial institution to identify all products available. ProductOffering has rates associated to the product based on date range, rewards programs, product brand, risk rating, etc. See the ProductOffering.xsd for reference data that can be stored for the ProductOffering.

# Any Known Errors in the Document

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

# Table of Contents

[Document Status 1](#_Toc532217019)

[Change Log 1](#_Toc532217020)

[Overview of Specification 1](#_Toc532217021)

[Any Known Errors in the Document 2](#_Toc532217022)

[Table of Contents 2](#_Toc532217023)

[Document Conventions 2](#_Toc532217024)

[Release 4.0 Global Update Notes 3](#_Toc532217025)

[Definitions Related to the Specification 3](#_Toc532217026)

[Data Elements 3](#_Toc532217027)

[Related Data Elements 3](#_Toc532217028)

[ProductOffering Object Attributes 3](#_Toc532217029)

[Services 4](#_Toc532217030)

[ProductOffering 4](#_Toc532217031)

[Overview 4](#_Toc532217032)

[Party Resource Based Create, Read, Update, Delete Services 4](#_Toc532217033)

[Service Message: Read Product Offering 5](#_Toc532217034)

[REST-JSON Read Request-Response (Example: Known Product Id’s) 5](#_Toc532217035)

[REST-JSON Read Request-Response (Example: Known Product Type) 6](#_Toc532217036)

[REST-JSON Read Request-Response (Example: Known Product Sub Type) 7](#_Toc532217037)

[REST-JSON Read Request-Response (Example: Known Product Interest Date Time) 8](#_Toc532217038)

[REST-JSON Read Request-Response (Example: Known Product Points Rewards Program) 9](#_Toc532217039)

[REST-JSON Read Request-Response (Example: Known Product Brand) 10](#_Toc532217040)

[REST-JSON Read Request-Response (Example: Known Product Risk Rate Classification) 11](#_Toc532217041)

[General Error Handling For All Services 12](#_Toc532217042)

[Bibliography 12](#_Toc532217043)

# 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

productoffering Data

List of possible products and interest rates associated to the products.

# Data Elements

## Related Data Elements

Refer to Security Services documentation to understand what may be contained in 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 **ProductOffering.xsd.**

The filters are used to filter based on associated data type for the product offering such as product offering Id, type, subtype, etc. Combining filters can be used to get list of product offerings for a product offering for a specific product brand, risk ranges, etc.

## ProductOffering Object Attributes

All CUFX fields related to a product offering are defined in **productOffering.xsd**.

Note: Fields not listed in the calling specification are not to be returned to the calling specification. i.e. If the field brand 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.

# Services

## ProductOffering

To READ high level ProductOffering data, clients access the ProductOffering data services.

* Specifications depend on the Security service for single-sign-on and/or authentication.

## Overview

|  |  |
| --- | --- |
| Definition | Collection of services to read productOffering reference data. |
| Overview of Capabilities | Read a product offering. |
| Dependencies | Security Services, messageContext |
| Sample CUFX REST LINK | https://api.dataprovider.com/productOffering |
| CUFX SOAP LINK |  |
| CUFX WaDL LINK |  |

## Party Resource Based Create, Read, Update, Delete Services

|  |  |
| --- | --- |
| INPUTS | cufx:productOfferingMessage (which includes)   * [cufx:MessageContext](file:///\\files2\users\CMarjaniemi\Projects\CUFX\MessageContext.html) * cufx:productOfferingFilter (for read, update) * cudf:productOfferingList (for create, update, delete) |
| Outputs | cufx:productOfferingMessage (which includes)   * [cufx:MessageContext](file:///\\files2\users\CMarjaniemi\Projects\CUFX\MessageContext.html) * cudf:productOfferingList |
| Return Values | cufx:productOfferingMessage (which includes)   * [cufx:MessageContext](file:///\\files2\users\CMarjaniemi\Projects\CUFX\MessageContext.html)   + statusList |
| Side Effects | Read has no side effects |
| Dependencies | Security Services for authentication and security. |
| Fields used | Message Headers : See security services  messageContext: See messageContext.xsd  Filters: See productOfferingFilter.xsd  Attributes: productOffering: See productOffering.xsd |
| Testing Procedures for Certification | Access the CUFX Product Offering data test service and verify that test data is returned appropriately. |

## Service Message: Read Product Offering

### REST-JSON Read Request-Response (Example: Known Product Id’s)

Example: Read a list of product offerings based on known Product Offering Id’s.

Clients submit a product offering data request to the Data Provider’s URL. For all CUFX RESTful-JSON requests, the client must set the appropriate headers. This example may not include all fields and is for example only.

REQUEST:

Headers:

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

Accept: application/json

Accept-Charset: utf-8

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

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

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

X-API-Version: >=4.0.0

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

{

“productOfferingMessage”:{

“messageContext”: { <see messageContext.xsd>

},

“productOfferingFilter”: {

“productIdList”:[ ”86753” , ”90246” ]

}

}

}

The Data Provider responds to requests with a JSON-formatted message:

**RESPONSE:**

Headers:

Status Code: 200 Ok

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

Content-Language: en-us

**Payload:**

{

“productOfferingMessage”:{

“messageContext”: { <see messageContext.xsd>

},

“productOfferingList”: [

{

“productID”: “86753”,

“productType”:”LineOfCredit”,

“productSubType”:”Premier”,

“description”: “Home Equity 500k bal loan offer”,

“interestRateList”: [

{

“interestRateID”:”05”,

“interestPointsRewardsProgram”:”true”,

“interestRate”:3.500

},

{

“interestRateID”:”06”,

“interestPointsRewardsProgram”:”false”,

“interestRate”:4.500

},

]

},

{

“productID”: “90246”,

“productType”:”Mortgage”,

“productSubType”:”Gold”,

“interestRateList”: [

{

“interestRateID”:”10”,

“riskRate”:”true”,

“ExpirateDateTime,”2013-03-31”,

“interestRate”:5.000

}

]

}

]

}

### REST-JSON Read Request-Response (Example: Known Product Type)

Example: Read a list of product offerings based on known product types.

Clients submit a product offering data request to the Data Provider’s URL. For all CUFX RESTful-JSON requests, the client must set the appropriate headers. This example may not include all fields and is for example only.

REQUEST:

Headers:

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

Accept: application/json

Accept-Charset: utf-8

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

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

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

X-API-Version: >=4.0.0

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

{

“productOfferingMessage”:{

“messageContext”: { <see messageContext.xsd>

},

“productOfferingFilter”: {

“productTypeList”:[ ”LineOfCredit”, ”Mortgage” ]

}

}

}

}

**RESPONSE:**

Headers:

Status Code: 200 Ok

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

Content-Language: en-us

**Payload:**

{

“productOfferingMessage”:{

“messageContext”: { <see messageContext.xsd>

},

“productOfferingList”: [

{

“productID”: “86753”,

“productType”:”LineOfCredit”,

“productSubType”:”Premier”,

“description”: “Home Equity 500k bal loan offer”,

“interestRateList”: [

{

“interestRateID”:”05”,

“interestPointsRewardsProgram”:”true”,

“interestRate”:3.500

}

]

},

{

“productID”: “90246”,

“productType”:”Mortgage”,

“productSubType”:”Gold”,

“interestRateList”: [

{

“interestRateID”:”10”,

“riskRate”:”false”,

“ExpirateDateTime,”2013-03-31”,

“interestRate”:5.000

}

]

},

{

“productID”: “90357”,

“productType”:”Mortgage”,

“productSubType”:”Basic”,

“interestRateList”: [

{

“interestRateID”:”08”,

“riskRate”:”true”,

“interestRate”:5.000

}

]

}

]

}

### REST-JSON Read Request-Response (Example: Known Product Sub Type)

Example: Read a list of product offerings based on known product sub types.

Clients submit a product offering data request to the Data Provider’s URL. For all CUFX RESTful-JSON requests, the client must set the appropriate headers. This example may not include all fields and is for example only.

REQUEST:

Headers:

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

Accept: application/json

Accept-Charset: utf-8

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

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

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

X-API-Version: >=4.0.0

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

{

“productOfferingMessage”:{

“messageContext”: { <see messageContext.xsd>

},

“productOfferingFilter”: {

“productSubTypeList”:[ ”Premier”, ”Gold” ]

}

}

}

**RESPONSE:**

Headers:

Status Code: 200 Ok

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

Content-Language: en-us

**Payload:**

{

“productOfferingMessage”:{

“messageContext”: { <see messageContext.xsd>

},

“productOfferingList”: [

{

“productID”: “90246”,

“productType”:”Mortgage”,

“productSubType”:”Gold”,

“interestRateList”: [

{

“interestRateID”:”10”,

“riskRate”:”true”,

“ExpirateDateTime,”2013-03-31”,

“interestRate”:5.000

}

]

},

{

“productID”: “90357”,

“productType”:”Mortgage”,

“productSubType”:”Premier”,

“interestRateList”: [

{

“interestRateID”:”08”,

“riskRate”:”false”,

“interestRate”:5.000

}

]

}

]

}

### REST-JSON Read Request-Response (Example: Known Product Interest Date Time)

Example: Read a list of product offerings based on date and time.

Clients submit a product offering data request to the Data Provider’s URL. For all CUFX RESTful-JSON requests, the client must set the appropriate headers. This example may not include all fields and is for example only.

REQUEST:

Headers:

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

Accept: application/json

Accept-Charset: utf-8

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

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

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

X-API-Version: >=4.0.0

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

{

“productOfferingMessage”:{

“messageContext”: { <see messageContext.xsd>

},

“productOfferingFilter”: {

“interestDateTime”:”2012-12-31”

}

}

}

The Data Provider responds to requests with a JSON-formatted message:

**RESPONSE:**

Headers:

Status Code: 200 Ok

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

Content-Language: en-us

**Payload:**

{

“productOfferingMessage”:{

“messageContext”: { <see messageContext.xsd>

},

“productOfferingList”: [

{

“productID”: “90246”,

“productType”:”Mortgage”,

“productSubType”:”Gold”,

“interestRateList”: [

{

“interestRateID”:”10”,

“riskRate”:”true”,

“ExpirateDateTime,”2013-03-31”,

“interestRate”:5.000

}

]

}

]

}

### REST-JSON Read Request-Response (Example: Known Product Points Rewards Program)

Example: Read a list of product offerings based on points rewards program filter.

Clients submit a product offering data request to the Data Provider’s URL. For all CUFX RESTful-JSON requests, the client must set the appropriate headers. This example may not include all fields and is for example only.

REQUEST:

Headers:

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

Accept: application/json

Accept-Charset: utf-8

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

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

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

X-API-Version: >=4.0.0

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

{

“productOfferingMessage”:{

“messageContext”: { <see messageContext.xsd>

},

“productOfferingFilter”: {

“pointsRewardsProgram”:”true”

}

}

}

}

The Data Provider responds to requests with a JSON-formatted message:

**RESPONSE:**

Headers:

Status Code: 200 Ok

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

Content-Language: en-us

**Payload:**

{

“productOfferingMessage”:{

“messageContext”: { <see messageContext.xsd>

},

“productOfferingList”: [

{

“productID”: “86753”,

“productType”:”LineOfCredity”,

“productSubType”:”Premier”,

“description”: “Home Equity 500k bal loan offer”,

“interestRateList”: [

{

“interestRateID”:”05”,

“interestPointsRewardsProgram”:”true”,

“interestRate”:3.500

}

]

}

]

}

### REST-JSON Read Request-Response (Example: Known Product Brand)

Example: Read a list of product offering based on known product brand.

Clients submit a product offering data request to the Data Provider’s URL. For all CUFX RESTful-JSON requests, the client must set the appropriate headers. This example may not include all fields and is for example only.

REQUEST:

Headers:

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

Accept: application/json

Accept-Charset: utf-8

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

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

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

X-API-Version: >=4.0.0

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

{

“productOfferingMessage”:{

“messageContext”: { <see messageContext.xsd>

},

“productOfferingFilter”: {

“brand”:”Visa”

}

}

}

The Data Provider responds to requests with a JSON-formatted message:

**RESPONSE:**

Headers:

Status Code: 200 Ok

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

Content-Language: en-us

**Payload:**

{

“productOfferingMessage”:{

“messageContext”: { <see messageContext.xsd>

},

“productOfferingList”: [

{

“productID”: “31415”,

“productType”:”CreditCard”,

“productSubType”:”Premier”,

“interestRateList”: [

{

“interestRateID”:”05”,

“interestPointsRewardsProgram”:”true”,

“interestRate”:3.500

}

],

“brand”:”Visa”

},

{

“productID”: “90246”,

“productType”:”CreditCard”,

“productSubType”:”Gold”,

“description”:”Visa Classic Gold”,

“interestRateList”: [

{

“interestRateID”:”10”,

“riskRate”:”true”,

“ExpirateDateTime,”2013-03-31”,

“interestRate”:5.000

}

],

“brand”:”Visa”

}

]

}

### REST-JSON Read Request-Response (Example: Known Product Risk Rate Classification)

Example: Read a list of product offerings based on known risk rate classification.

Clients submit a product offering data request to the Data Provider’s URL. For all CUFX RESTful-JSON requests, the client must set the appropriate headers. This example may not include all fields and is for example only.

REQUEST:

Headers:

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

Accept: application/json

Accept-Charset: utf-8

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

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

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

X-API-Version: >=4.0.0

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

{

“productOfferingMessage”:{

“messageContext”: { <see messageContext.xsd>

},

“productOfferingFilter”: {

“riskRateClassification”:“10”

}

}

}

The Data Provider responds to requests with a JSON-formatted message:

**RESPONSE:**

Headers:

Status Code: 200 Ok

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

Content-Language: en-us

**Payload:**

{

“productOfferingMessage”:{

“messageContext”: { <see messageContext.xsd>

},

“productOfferingList”: [

{

“productID”: “90246”,

“productType”:”Mortgage”,

“productSubType”:”Gold”,

“interestRateList”: [

{

“interestRateID”:”10”,

“riskRate”:”true”,

“ExpirateDateTime,”2013-03-31”,

“interestRate”:5.000

}

]

},

{

“productID”: “90357”,

“productType”:”Mortgage”,

“productSubType”:”Basic”,

“interestRateList”: [

{

“interestRateID”:”08”,

“riskRate”:”false”,

“interestRate”:5.000

}

]

}

]

}

# 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.