

Billpay Model and Services 4.5

# Document Status

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

# Change Log

|  |  |  |
| --- | --- | --- |
| Version | Date | Changes |
| 3.0 | 12/17/2013 | * Initial Creation |
| 3.0 | 12/24/2013 | * Create a billPaymentMessage wrapper for every message to increase ability for infrastructure to serialize the data. |
| 3.0 | 01/28/2014 | * Updated JSON Examples for XSD modifications. Added Use Case Diagram. |
| 3.0 | 02/10/2014 | * Updated known errors in the document |
| 3.0 | 04/10/2014 | * Updated JSON examples, corrected known errors, updated diagram. |
| 3.0 | 07/28/2014 | * Updated Diagram in Services. Corrected minor typos, added content on filtering. |
| 3.1 | 07/17/2015 | * Updated to release 3.1. Added FundsTransferCommonBase . Updated diagram. |
| 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, date/amounts fields added to Payee, Change minOccurs to zero for isElectronic Added payeeAddedDateRange and payeeUpdatedDateRange, Added recurringFromAccountList, recurringToAccountList, recurringStatusList, date range filters for all FundsTransferReccurringBase dates to filter. Added bill status and date payment made. |
| 4.1 | 12/10/2018 | * Updated to release 4.1, * BillFilter - Added billStatusList to Filter * BillPaymentReccurringMessage - Add BillPaymentReccurringMessage as correction to BillPaymentReoccurringMessage, Corrected spelling error in xsd name |
| 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. * BillFilter - Removed duplicated billStatusList element. |
| 4.3 | 10/07/2019 | * Updated to release 4.3, renamed file removing version as proper version control is being used in Github. BillStatus added "Scheduled" and “Other” as an enum value. Added otherBillStatus element to Bill. |
| 4.4 | 10/20/2020 | * Updated to release 4.4, Bill - Added accountType and accountSubType to Bill. * BillFilter - Added accountIdentificationList to filter. * BillPayee - Added accountType and accountSubType to BillPayee. * Added eBillEnrollmentStatus, and otherEBillEnrollmentStatus * Added EBillEnrollmentStatusList * BillPayeeFilter - Added accountIdentificationList, EBillEnrollmentStatusList to filter. * BillPaymentFilter.xsd - Added Account to Namespace. Added accountIdentificationList to filter. * BillPaymentOccurrence - Removed transaction namespace reference. Updated checkNumber to common type reference. |
| 4.5 | 04/02/2020 | * Updated to release 4.5 |

# Overview of Specification

The CUFX Bill Payment Service provides a standard description for payment definitions and instances of those payments. The CUFX Bill Payment Service provide the information necessary to pass create, read, update, and delete bill payment transactions between credit union data systems.

# Any know Errors in the document

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

Table of Contents

[Document Status 1](#_Toc68097567)

[Change Log 1](#_Toc68097568)

[Overview of Specification 2](#_Toc68097569)

[Any know Errors in the document 2](#_Toc68097570)

[Document Conventions 3](#_Toc68097571)

[CUFX API and Documentation Support 3](#_Toc68097572)

[Release 4.0 Global Update Notes 3](#_Toc68097573)

[Release 4.4 Global Update Notes 4](#_Toc68097574)

[Definitions related to the specification 4](#_Toc68097575)

[Data Elements 5](#_Toc68097576)

[Filters used when accessing the Bill Payment data 5](#_Toc68097577)

[Use Case 6](#_Toc68097578)

[Services 7](#_Toc68097579)

[Authentication 9](#_Toc68097580)

[Bill Read 9](#_Toc68097581)

[REST-JSON QUERY INDIVIDUAL BILL TRANSACTION 9](#_Toc68097582)

[Bill PAYMENT OCCURRENCE CREATE, READ, UPDATE, DELETE 11](#_Toc68097583)

[REST-JSON CREATE BILL PAYMENT OCCURRENCE EXAMPLE 11](#_Toc68097584)

[REST-JSON READ BILL PAYMENT OCCURRENCE EXAMPLE 12](#_Toc68097585)

[REST-JSON UPDATE BILL PAYMENT OCCURRENCE EXAMPLE 14](#_Toc68097586)

[REST-JSON DELETE BILL PAYMENT OCCURRENCE EXAMPLE 15](#_Toc68097587)

[Bill PAYEE CREATE, READ, UPDATE, DELETE 17](#_Toc68097588)

[REST-JSON CREATE BILL PAYEE EXAMPLE 17](#_Toc68097589)

[REST-JSON READ BILL PAYEE EXAMPLE 19](#_Toc68097590)

[REST-JSON UPDATE BILL PAYEE EXAMPLE 21](#_Toc68097591)

[REST-JSON DELETE BILL PAYEE EXAMPLE 22](#_Toc68097592)

[Bill PAYMEnT RECURRING CREATE, READ, UPDATE, DELETE 24](#_Toc68097593)

[REST-JSON CREATE BILL PAYMENT RECURRING EXAMPLE 24](#_Toc68097594)

[REST-JSON READ BILL PAYMENT RECURRING EXAMPLE 26](#_Toc68097595)

[REST-JSON UPDATE BILL PAYMENT RECURRING EXAMPLE 27](#_Toc68097596)

[REST-JSON DELETE BILL PAYMENT RECURRING EXAMPLE 28](#_Toc68097597)

[General Error handling For All Services 29](#_Toc68097598)

[Bibliography 30](#_Toc68097599)

# 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

**Bill Pay Service Definitions**

**BillPaymentOccurence** – This object represents an instance of the bill payment.

**BillPaymentRecurring** – This object contains the bill payment specific information needed to create an instance of a recurring bill payment.

**FundsTransferOccurenceBase** – This object contains the details about an occurrence of a funds transfer. It may contain a recurring Id as reference to a recurring payment.

**FundsTransferRecurringBase** – This object contains the details about a recurring funds transfer. The details of this object are linked to a funds transfer occurrence through the recurring Id.

**FundsTransferCommonBase –** This object contains the common type definitions used in the Base objects and filters. Definition was added in release 3.1 to resolve circular reference conditions.

**Bill Payee Service Definitions**

**Bill** – This object contains details about a bill item.

**BillPayee** – This object contains the details about a payee or service provider (e.g., utility, phone provider, etc.).

# Data Elements

## Filters used when accessing the Bill Payment data

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. Refer to recent CUFX messageContext Data and CUFX Security Services for use of MessageContext.xsd. Include any filter variables related to the request.

See **BillFilter.xsd, BillPaymentFilter.xsd,** and **BillPayeeFilter.xsd.**

# Use Case

1

READ a Bill with BillMessage

CUFX OLB

2

Return Bill Detail or error code

3

READ for BillPayee with BillPayeeMessage

14

Return confirmation or error code

13

Delete BillPayment with billPaymentOccurrenceMessage

12

Return confirmation or error code

11

Update BillPayment with billPaymentOccurrenceMessage

9

READ for BillPayment with billPaymentOccurrenceMessage

10

Return the list of bill payments or error code

8

Return confirmation or error code

7

Delete BillPayee with BillPayeeMessage

5

Update BillPayee with BillPayeeMessage

6

Return confirmation or error code

4

Return the list of payees or error code

CUFX Data Provider

Note: for Recurring Bill Payments use cases replace billPaymentOccurrence with billPaymentRecurring as noted above.

# Services

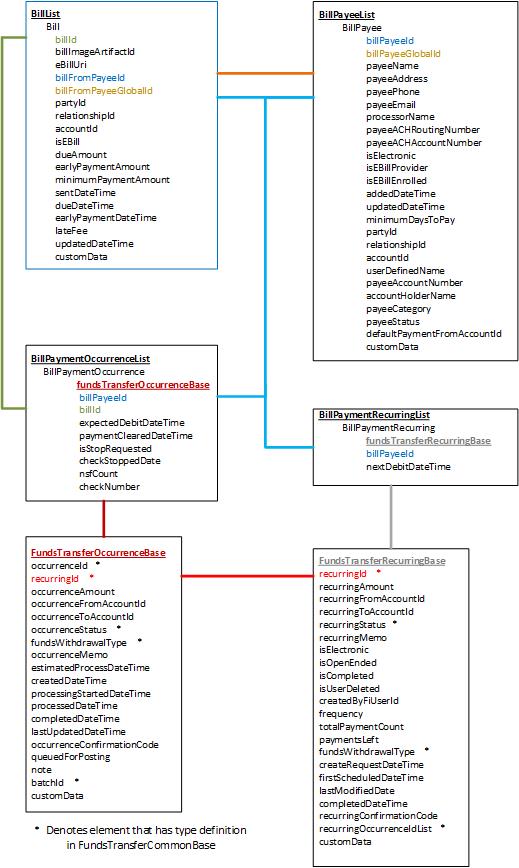
|  |  |
| --- | --- |
| Definition | Bill Payments |
| Overview of Capabilities | Service to create payments and set dependent data. |
| Dependencies | Security Services, messageContext |
| CUFX REST LINK | [https://api.dataprovider.com/](https://api.dataprovider.com/deposit)billmessage  https://api.dataprovider.com/billpaymentoccurrencemessage  [https://api.dataprovider.com/](https://api.dataprovider.com/deposit)billpaymentrecurringmessage  [https://api.dataprovider.com/](https://api.dataprovider.com/deposit)billpayeemessage |
| CUFX WSDL LINK |  |

The Bill Pay Service is a logical grouping of four services. The services are bill, billPaymentOccurrence, billPaymentRecurring, and billPayee. These four services support the creation, viewing, and maintenance of bill payments between different systems. The bill service primarily exists to read bill information. The current model while extensible does not currently consider the creation and maintenance of bills using CUFX as a number of organizations creating Bills may lie outside of the Credit Union Industry. Capture of bill information is used to populate a bill payment occurrence or recurring transaction. For each payment type, occurring and recurring, a base xsd has been established to contain all of the elements common for that transaction and included as an extension base. The funds transfer base xsds are also used with transfers, and wire transactions.

The funds transfer base XSDs have been designed to include all common elements necessary for the payment transactions. Due to the nature of payments types, not all fields contained in the xsd may be applicable for the type and may be disregarded needed. This is an intentional decision to allow flexibility for the implementation and to keep the structures simplified.

The diagram below show the relationships between the bill, payee, payment occurrence and a payment recurring.

The field and the corresponding line has been color coded to enhance readability and to reflect keys between the structures.



## Authentication

See Security Services and messageContext for more detail.

## Bill Read

|  |  |
| --- | --- |
| INPUTS | cufx:billMessage (which includes)   * cufx:messageContext * cufx:billFilter (for read, update) |
| Outputs | cufx:billMessage (which includes)   * cufx:MessageContext * cufx:billist |
| Return Values | cufx: billMessage (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. |
| Fields used | Message Headers : See security services  messageContext: See messageContext.xsd  Errors: See MessageContext.xsd |
| Testing Procedures for Certification | Send details for a new account and verify that it has been created, with the id returned in the deposit object. |

### REST-JSON QUERY INDIVIDUAL BILL TRANSACTION

In some cases the remote system may need to request the details for a specific instance of a bill. This example shows call that reads an existing instance of a bill. The bill filter is used as search criterion.

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: >=4.5.0

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

{

"billMessage": {

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

"messageContext": {

<see messageContext.xsd>

},

"billFilter": {

"billIdList": {

"billId": [

"439987"

]

},

"accountIdList": {

"accountId": [

"accountId1"

]

},

"billFromPayeeIdList": {

"billPayeeId": [

"billPayeeId1"

]

}

}

}

}

RESPONSE

Headers:

Status Code: 200 Ok

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

Content-Language: en-us

Payload:

{ "billMessage": {

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

"messageContext": {

<see messageContext.xsd>

},

"billList": {

"bill": [

{

"billId": "439987",

"billImageArtifactId": {

"artifactUniqueId": "artifactUniqueId1"

},

"eBillUri": "http://uri1",

"billFromPayeeId": "billFromPayeeId1",

"billFromPayeeGlobalId": "billFromPayeeGlobalId1",

"partyId": "partyId1",

"relationshipId": "relationshipId1",

"accountId": "accountId1",

"isEBill": "true",

"dueAmount": {

"value": "125.32",

"currencyCode": "USD"

},

"earlyPaymentAmount": {

"value": "125.32",

"currencyCode": "USD"

},

"minimumPaymentAmount": {

"value": "20.00",

"currencyCode": "USD"

},

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

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

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

"lateFee": {

"value": "5.00",

"currencyCode": "USD"

},

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

}

]

}

}

}

## Bill PAYMENT OCCURRENCE CREATE, READ, UPDATE, DELETE

|  |  |
| --- | --- |
| INPUTS | cufx: billPaymentOccurrenceMessage (which includes)   * cufx:messageContext * cufx:billPaymentFilter (read, update,) * cufx:billPaymentList (create,update, delete) |
| Outputs | cufx: billPaymentOccurrenceMessage (which includes)   * cufx:messageContext * cufx:billPaymentList |
| Return Values | cufx: billPaymentOccurrenceMessage (which includes)   * cufx:messageContext   + statusList |
| Side Effects | Creation, update or deletion of bill payment on system of record. Read has no side effects. |
| Dependencies | Security Services for authentication and security. |
| Fields used | Message Headers : See security services  messageContext: See messageContext.xsd  Errors: See MessageContext.xsd |
| Testing Procedures for Certification | Send details for a new account and verify that it has been created, with the id returned in the deposit object. |

### REST-JSON CREATE BILL PAYMENT OCCURRENCE EXAMPLE

This example shows call that creates a new instance of a bill payment.

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-API-Version: >=4.5.0

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

{

"billPaymentOccurrenceMessage": {

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

"messageContext": {

<see messageContext.xsd>

},

"billPaymentList": {

"billPaymentOccurrence": [

{

"occurrenceId": "occurrenceId1",

"occurrenceAmount": {

"value": "150.00",

"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",

"billPayeeId": "billPayeeId1",

"billId": "billId1",

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

"nsfCount": "1",

"checkNumber": "checkNumber1"

}

}

}

}

}

RESPONSE

Headers:

Status Code: 200 Ok

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

Content-Language: en-us

Payload:

{{

"billPaymentOccurrenceMessage": {

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

"messageContext": {

<see messageContext.xsd>

},

"billPaymentList": {

"billPaymentOccurrence": [

{

"occurrenceId": "occurrenceId1",

"occurrenceAmount": {

"value": "150.00",

"currencyCode": "USD"

},

"occurrenceFromAccountId": "occurrenceFromAccountId1",

"occurrenceToAccountId": "occurrenceToAccountId1",

"occurrenceStatus": "Scheduled",

"fundsWithdrawalType": "WhenRequested",

"occurrenceMemo": "occurrenceMemo1",

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

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

"occurrenceConfirmationCode": "occurrenceConfirmationCode1",

"queuedForPosting": "true",

"note": "note1",

"batchId": "batchId1",

"billPayeeId": "billPayeeId1",

"billId": "billId1",

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

"nsfCount": "1",

"checkNumber": "checkNumber1"

}

}

}

}

### REST-JSON READ BILL PAYMENT OCCURRENCE EXAMPLE

This example shows call that reads an existing instance of a bill payment(s) matching the billPaymentOccurrenceId in the BillPaymentFilter.

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: >=4.5.0

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

{

"billPaymentOccurrenceMessage": {

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

"messageContext": {

<see messageContext.xsd>

},

"billPaymentFilter": {

"occurrenceIdList": {

"occurrenceId": [

"1234567"

]

},

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

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

"accountIDList": {

"accountId": [

"accountId1"

]

},

"minAmount": {

"value": "1",

"currencyCode": "USD"

},

"maxAmount": {

"value": "1000",

"currencyCode": "USD"

},

"batchIdList": {

"batchId": [

"batchId1"

]

},

"billPayeeIdList": {

"billPayeeId": [

"billPayeeId1",

]

},

"billIdList": {

"billId": [

"billId1"

]

}

}

}

}

RESPONSE

Headers:

Status Code: 200 Ok

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

Content-Language: en-us

Payload:

{

"billPaymentOccurrenceMessage": {

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

"messageContext": {

<see messageContext.xsd>

},

"billPaymentList": {

"billPaymentOccurrence": [

{

"occurrenceId": "1234567",

"occurrenceAmount": {

"value": "143.33",

"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",

"billPayeeId": "billPayeeId1",

"billId": "billId1",

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

"nsfCount": "1",

"checkNumber": "checkNumber1"

}

}

}

}

}

### REST-JSON UPDATE BILL PAYMENT OCCURRENCE EXAMPLE

This example shows call that updates an existing instance of a bill payment.

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-API-Version: >=4.5.0

**PUT** https://api.dataprovider.com/billpaymentoccurrencemessage

{

"billPaymentOccurrenceMessage": {

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

"messageContext": {

<see messageContext.xsd>

},

"billPaymentFilter": {

"occurrenceIdList": {

"occurrenceId": [

"654353"

]

},

"billPayeeIdList": {

"billPayeeId": [

"1234567",

]

},

"billIdList": {

"billId": [

"12345"

]

}

},

"billPaymentList": {

"billPaymentOccurrence": [

{

"occurrenceId": "654353",

"occurrenceAmount": {

"value": "85,32",

"currencyCode": "USD"

},

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

"batchId": "batchId1",

"billPayeeId": "1234567",

"billId": "billId1",

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

}

}

}

}

}

RESPONSE

Headers:

Status Code: 200 Ok

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

Content-Language: en-us

Payload:

{{

"billPaymentOccurrenceMessage": {

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

"messageContext": {

<see messageContext.xsd>

},

"billPaymentList": {

"billPaymentOccurrence": [

{

"occurrenceId": "654353",

"occurrenceAmount": {

"value": "85,32",

"currencyCode": "USD"

},

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

"occurrenceConfirmationCode": "occurrenceConfirmationCode1",

"batchId": "batchId1",

"billPayeeId": "1234567",

"billId": "billId1",

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

}

}

}

}

}

}

### REST-JSON DELETE BILL PAYMENT OCCURRENCE EXAMPLE

This example shows call that deletes an existing instance of a bill payment.

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: DELETE**

X-API-Version: >=4.5.0

**PUT** https://api.dataprovider.com/billpaymentoccurrencemessage

{

"billPaymentOccurrenceMessage": {

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

"messageContext": {

<see messageContext.xsd>

},

"billPaymentFilter": {

"occurrenceIdList": {

"occurrenceId": [

"654353"

]

},

"billPayeeIdList": {

"billPayeeId": [

"1234567",

]

},

"billIdList": {

"billId": [

"12345"

]

}

}

}

}

RESPONSE

Headers:

Status Code: 200 Ok

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

Content-Language: en-us

Payload:

{ "billPaymentOccurrenceMessage": {

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

"messageContext": {

"cufxVersion": "4.5.0",

"requestId": "requestId1",

"vendorId": "vendorId1",

"appId": "appId1",

"fiId": "fiId1",

"dataSourceId": "dataSourceId1",

"environment": "Development",

"returnDataFilter": "All",

"includeBlankFields": "true",

"includeZeroNumerics": "true",

"user": {

"userId": "userId1",

"processorSessionId": "processorSessionId1",

"userType": "EmployeeId"

},

"statusList": {

"status": {

"statusType": "Success"

}

},

},

},

}

}

## Bill PAYEE CREATE, READ, UPDATE, DELETE

|  |  |
| --- | --- |
| INPUTS | cufx:billPayeeMessage (which includes)   * cufx:messageContext * cufx:billPayeeFilter (read, update) * cufx:billPayeeList (create, update, delete) |
| Outputs | cufx:billPayeeMessage (which includes)   * cufx:messageContext * cufx:billPayeeList |
| Return Values | cufx:billPayeeMessage (which includes)   * cufx:MessageContext   + statusList |
| Side Effects | Creation, update or deletion of deposit account, in some cases party, contact, Read has no side effects. |
| Dependencies | Security Services for authentication and security. |
| Fields used | **Message Headers** : See security services  **messageContext**: See messageContext.xsd  **TBD** |
| Testing Procedures for Certification | Send details for a new account and verify that it has been created, with the id returned in the deposit object. |

### REST-JSON CREATE BILL PAYEE EXAMPLE

This example shows call that creates a new instance of a bill payee.

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-API-Version: >=4.5.0

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

{

"billPayeeMessage": {

“messageContext”: { <see messageContext.xsd>

},

"billPayeeList": {

"billPayee": [

{

"billPayeeId": "billPayeeId1",

"billPayeeGlobalId": "billPayeeGlobalId1",

"payeeName": "payeeName1",

"payeeAddress": {

"line1": "line11",

"line2": "line21",

"line3": "line31",

"city": "city1",

"stateProvince": "stateProvince1",

"postalCode": "postalCode1",

"countryCode": "AD",

"country": "country1",

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

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

"timeAtAddress": "P396DT1H1M1S",

"addressVerifyDateTime": "1900-01-01T01:01:01Z"

},

"payeePhone": {

"type": "Home",

"number": "number1",

"extension": "extension1",

"planFormat": "Nanp",

"description": "description1",

"smsRegistered": "true"

},

"payeeEmail": {

"type": "Home",

"address": "address1"

},

"processorName": "processorName1",

"payeeACHRoutingNumber": "1234567",

"payeeACHAccountNumber": "7654321",

"isElectronic": "true",

"isEBillProvider": "true",

"isEBillEnrolled": "true",

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

"minimumDaysToPay": "1",

"partyId": "partyId1",

"relationshipId": "relationshipId1",

"accountId": "accountId1",

"userDefinedName": "userDefinedName1",

"payeeAccountNumber": "payeeAccountNumber1",

"accountHolderName": "accountHolderName1",

"payeeCategory": "payeeCategory1",

"defaultPaymentFromAccountId": "1234567",

}

]

}

}

}

RESPONSE

Headers:

Status Code: 200 Ok

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

Content-Language: en-us

Payload:

{

"billPayeeMessage": {

“messageContext”: { <see messageContext.xsd>

},

"billPayeeList": {

"billPayee": [

{

"billPayeeId": "billPayeeId1",

"billPayeeGlobalId": "billPayeeGlobalId1",

"payeeName": "payeeName1",

"payeeAddress": {

"line1": "line11",

"line2": "line21",

"line3": "line31",

"city": "city1",

"stateProvince": "stateProvince1",

"postalCode": "postalCode1",

"countryCode": "AD",

"country": "country1",

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

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

"timeAtAddress": "P396DT1H1M1S",

"addressVerifyDateTime": "1900-01-01T01:01:01Z"

},

"payeePhone": {

"type": "Home",

"number": "number1",

"extension": "extension1",

"planFormat": "Nanp",

"description": "description1",

"smsRegistered": "true"

},

"payeeEmail": {

"type": "Home",

"address": "address1"

},

"processorName": "processorName1",

"payeeACHRoutingNumber": "1234567",

"payeeACHAccountNumber": "7654321",

"isElectronic": "true",

"isEBillProvider": "true",

"isEBillEnrolled": "true",

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

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

"minimumDaysToPay": "1",

"partyId": "partyId1",

"relationshipId": "relationshipId1",

"accountId": "accountId1",

"userDefinedName": "userDefinedName1",

"payeeAccountNumber": "payeeAccountNumber1",

"accountHolderName": "accountHolderName1",

"payeeCategory": "payeeCategory1",

"payeeStatus": "Active",

"defaultPaymentFromAccountId": "1234567",

}

}

]

}

}

}

### REST-JSON READ BILL PAYEE EXAMPLE

This example shows call that reads an existing instance of a bill payee.

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: >=4.5.0

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

{

"billPayeeMessage": {

"messageContext": { <see messageContext.xsd>

},

"billPayeeFilter": {

"billPayeeGlobalIdList": {

"billPayeeGlobalId": [

"2998765"

]

},

"accountIdList": {

"accountId": [

"accountId1"

]

},

"payeeName": "payeeName1",

"isElectronic": "true",

"payeeCategory": "payeeCategory1",

"payeeStatusList": {

"payeeStatus": [

"Active"

]

}

}

}

}

RESPONSE

Headers:

Status Code: 200 Ok

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

Content-Language: en-us

Payload:

{"billPayeeMessage": {

"messageContext": { <see messageContext.xsd>

},

"billPayeeList": {

"billPayee": [

{

"billPayeeId": "124775",

"billPayeeGlobalId": "2998765",

"payeeName": "payeeName1",

"payeeAddress": {

"line1": "line11",

"line2": "line21",

"line3": "line31",

"city": "city1",

"stateProvince": "stateProvince1",

"postalCode": "postalCode1",

"countryCode": "AD",

"country": "country1",

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

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

"timeAtAddress": "P396DT1H1M1S",

"addressVerifyDateTime": "1900-01-01T01:01:01Z"

},

"payeePhone": {

"type": "Home",

"number": "number1",

"extension": "extension1",

"planFormat": "Nanp",

"description": "description1",

"smsRegistered": "true"

},

"payeeEmail": {

"type": "Home",

"address": "address1"

},

"processorName": "processorName1",

"payeeACHRoutingNumber": "1234567",

"payeeACHAccountNumber": "7654321",

"isElectronic": "true",

"isEBillProvider": "true",

"isEBillEnrolled": "true",

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

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

"minimumDaysToPay": "1",

"partyId": "partyId1",

"relationshipId": "relationshipId1",

"accountId": "accountId1",

"userDefinedName": "userDefinedName1",

"payeeAccountNumber": "payeeAccountNumber1",

"accountHolderName": "accountHolderName1",

"payeeCategory": "payeeCategory1",

"payeeStatus": "Active",

"defaultPaymentFromAccountId": "1234567"

}

}

]

}

}

}

### REST-JSON UPDATE BILL PAYEE EXAMPLE

This example shows call that updates an existing instance of a bill payee.

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-API-Version: >=4.5.0

**PUT** https://api.dataprovider.com/billpayeemessage

{

"billPayeeMessage": {

“messageContext”: { <see messageContext.xsd>

},

"billPayeeFilter": {

"billPayeeGlobalIdList": {

"billPayeeGlobalId": [

"2998765"

]

},

"accountIdList": {

"accountId": [

"accountId1"

]

},

"payeeName": "payeeName1",

"isElectronic": "true",

"payeeCategory": "payeeCategory1",

"payeeStatusList": {

"payeeStatus": [

"Active"

]

}

},

"billPayeeList": {

"billPayee": [

{

"billPayeeId": "123456",

"billPayeeGlobalId": "2998765",

"payeeName": "payeeName2",

"payeeAddress": {

"line1": "New Address",

"city": "New City",

"stateProvince": "NY",

"postalCode": "10001",

"countryCode": "US",

}

]

}

}

}

RESPONSE

Headers:

Status Code: 200 Ok

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

Content-Language: en-us

Payload:

{"billPayeeMessage": {

“messageContext”: { <see messageContext.xsd>

},

"billPayeeList": {

"billPayee": [

{

"billPayeeId": "124775",

"billPayeeGlobalId": "2998765",

"payeeName": "payeeName2",

"payeeAddress": {

"line1": "New Address",

"city": "New City",

"stateProvince": "NY",

"postalCode": "10001",

"countryCode": "US",

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

"payeeStatus": "Active"

}

}

]

}

}

}

### REST-JSON DELETE BILL PAYEE EXAMPLE

This example shows call that deletes an existing instance of a bill payee.

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: DELETE**

X-API-Version: >=4.5.0

**PUT** https://api.dataprovider.com/billpayeeMmessage

{

"billPayeeMessage": {

“messageContext”: { <see messageContext.xsd>

},

"billPayeeFilter": {

"billPayeeGlobalIdList": {

"billPayeeGlobalId": [

"2998765"

]

},

"accountIdList": {

"accountId": [

"236684"

]

},

"payeeName": "payeeName1",

"payeeStatusList": {

"payeeStatus": [

"Active"

]

}

},

RESPONSE

Headers:

Status Code: 200 Ok

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

Content-Language: en-us

Payload:

{

"billPayeeMessage": {

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

"messageContext": {

"cufxVersion": "4.0.0",

"requestId": "requestId1",

"vendorId": "vendorId1",

"appId": "appId1",

"fiId": "fiId1",

"dataSourceId": "dataSourceId1",

"environment": "Development",

"returnDataFilter": "All",

"includeBlankFields": "true",

"includeZeroNumerics": "true",

"user": {

"userId": "userId1",

"processorSessionId": "processorSessionId1",

"userType": "EmployeeId"

},

"statusList": {

"status": {

"statusType": "Success"

}

},

},

},

}

}

## Bill PAYMEnT RECURRING CREATE, READ, UPDATE, DELETE

|  |  |
| --- | --- |
| INPUTS | cufx: billPaymentRecurringMessage (which includes)   * cufx:messageContext * cufx:billPaymentFilter (read, update,) * cufx:billPaymentRecurringList (create, update, delete) |
| Outputs | cufx: billPaymentRecurringMessage (which includes)   * cufx:messageContext * cufx:billPaymentRecurringList |
| Return Values | cufx: billPaymentRecurringMessage (which includes)   * cufx:MessageContext   + statusList |
| Side Effects | Creation, update or deletion of recurring bill payment. Read has no side effects. |
| Dependencies | Security Services for authentication and security. |
| Fields used | **Message Headers** : See security services  **messageContext**: See messageContext.xsd  **Errors:** See MessageContext.xsd |
| Testing Procedures for Certification | Send details for a new account and verify that it has been created, with the id returned in the deposit object. |

### REST-JSON CREATE BILL PAYMENT RECURRING EXAMPLE

This example shows call that creates a new instance of a recurring bill payment.

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-API-Version: >=4.5.0

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

{

"billPaymentRecurringMessage": {

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

"messageContext": {

“messageContext”: { <see messageContext.xsd>

},

"billPaymentRecurringList": {

"billPaymentRecurring": [

{

"recurringId": "recurringId1",

"recurringAmount": {

"value": "234.12",

"currencyCode": "USD"

},

"recurringFromAccountId": "recurringFromAccountId1",

"recurringToAccountId": "recurringToAccountId1",

"recurringStatus": "Requested",

"recurringMemo": "recurringMemo1",

"isElectronic": "true",

"isOpenEnded": "true",

"createdByFiUserId": "createdByFiUserId1",

"frequency": "Monthly",

"totalPaymentCount": "1",

"paymentsLeft": "10",

"fundsWithdrawalType": "WhenRequested",

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

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

"recurringOccurrenceIdList": {

"occurrenceId": [

"occurrenceId1"

]

},

"billPayeeId": "billPayeeId1",

"nextDebitDateTime": "1900-01-01T01:01:01Z"

}

]

}

}

}

RESPONSE

Headers:

Status Code: 200 Ok

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

Content-Language: en-us

Payload:

{

"billPaymentRecurringMessage": {

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

"messageContext": {

“messageContext”: { <see messageContext.xsd>

},

"billPaymentRecurringList": {

"billPaymentRecurring": [

{

"recurringId": "recurringId1",

"recurringAmount": {

"value": "234.12",

"currencyCode": "USD"

},

"recurringFromAccountId": "recurringFromAccountId1",

"recurringToAccountId": "recurringToAccountId1",

"recurringStatus": "Requested",

"recurringMemo": "recurringMemo1",

"isElectronic": "true",

"isOpenEnded": "true",

"createdByFiUserId": "createdByFiUserId1",

"frequency": "Monthly",

"totalPaymentCount": "1",

"paymentsLeft": "10",

"fundsWithdrawalType": "WhenRequested",

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

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

"recurringConfirmationCode": "recurringConfirmationCode1",

"recurringOccurrenceIdList": {

"occurrenceId": [

"occurrenceId1"

]

},

"billPayeeId": "billPayeeId1",

"nextDebitDateTime": "1900-01-01T01:01:01Z"

}

]

}

}

}

### REST-JSON READ BILL PAYMENT RECURRING EXAMPLE

This example shows call that reads an existing instance of a recurring bill payment.

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: >=4.5.0

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

{

"billPaymentRecurringMessage": {

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

"messageContext": {

“messageContext”: { <see messageContext.xsd>

},

"billPaymentFilter": {

"recurringIdList": {

"recurringId": [

"1243256"

]

},

"accountIDList": {

"accountId": [

"987654"

]

},

"minAmount": {

"value": "1",

"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:

{

"billPaymentRecurringMessage": {

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

"messageContext": {

“messageContext”: { <see messageContext.xsd>

},

"billPaymentRecurringList": {

"billPaymentRecurring": [

{

"recurringId": "1243256",

"recurringAmount": {

"value": "234.12",

"currencyCode": "USD"

},

"recurringFromAccountId": "987654",

"recurringToAccountId": "recurringToAccountId1",

"recurringStatus": "Requested",

"recurringMemo": "recurringMemo1",

"isElectronic": "true",

"isOpenEnded": "true",

"createdByFiUserId": "createdByFiUserId1",

"frequency": "Monthly",

"totalPaymentCount": "1",

"paymentsLeft": "10",

"fundsWithdrawalType": "WhenRequested",

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

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

"recurringConfirmationCode": "recurringConfirmationCode1",

"recurringOccurrenceIdList": {

"occurrenceId": [

"occurrenceId1"

]

},

"billPayeeId": "billPayeeId1",

"nextDebitDateTime": "1900-01-01T01:01:01Z"

}

]

}

}

}

### REST-JSON UPDATE BILL PAYMENT RECURRING EXAMPLE

This example shows call that updates an existing instance of a recurring bill payment.

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-API-Version: >=4.5.0

**PUT** https://api.dataprovider.com/billpaymentrecurringmessage

{

"billPaymentRecurringMessage": {

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

"messageContext": {

“messageContext”: { <see messageContext.xsd>

},

"billPaymentFilter": {

"recurringIdList": {

"recurringId": [

"4553346"

]

},

"accountIDList": {

"accountId": [

"7876654"

]

}

},

"billPaymentRecurringList": {

"billPaymentRecurring": [

{

"recurringId": "4553346",

"recurringAmount": {

"value": "114.12",

"currencyCode": "USD"

},

"recurringOccurrenceIdList": {

"occurrenceId": [

"occurrenceId1"

]

},

"billPayeeId": "1234633",

}

]

}

}

}

RESPONSE

Headers:

Status Code: 200 Ok

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

Content-Language: en-us

Payload:

{

"billPaymentRecurringMessage": {

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

"messageContext": {

“messageContext”: { <see messageContext.xsd>

},

"billPaymentRecurringList": {

"billPaymentRecurring": [

{

"recurringId": "4553346",

"recurringAmount": {

"value": "114.12",

"currencyCode": "USD"

},

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

"recurringConfirmationCode": "recurringConfirmationCode1",

"recurringOccurrenceIdList": {

"occurrenceId": [

"occurrenceId1"

]

},

"billPayeeId": "1234633",

}

]

}

}

}

### REST-JSON DELETE BILL PAYMENT RECURRING EXAMPLE

This example shows call that deletes an existing instance of a recurring bill payment.

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: DELETE**

X-API-Version: >=4.5.0

**PUT** https://api.dataprovider.com/billpaymentrecurringmessage

{

"billPaymentRecurringMessage": {

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

"messageContext": {

“messageContext”: { <see messageContext.xsd>

},

"billPaymentFilter": {

"recurringIdList": {

"recurringId": [

"4553346"

]

},

"accountIDList": {

"accountId": [

"7876654"

]

}

}

}

}

RESPONSE

Headers:

Status Code: 200 Ok

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

Content-Language: en-us

Payload:

{

"billPaymentRecurringMessage": {

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

"messageContext": {

“messageContext”: {

"cufxVersion": "4.5.0",

"requestId": "requestId1",

"vendorId": "vendorId1",

"appId": "appId1",

"fiId": "fiId1",

"dataSourceId": "dataSourceId1",

"environment": "Development",

"returnDataFilter": "All",

"includeBlankFields": "true",

"includeZeroNumerics": "true",

"user": {

"userId": "userId1",

"processorSessionId": "processorSessionId1",

"userType": "EmployeeId"

},

"statusList": {

"status": {

"statusType": "Success"

}

},

},

},

}

}

# 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.