

Account Opening and Initial Funding USE CASE 5.0

# 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 |
| 1.0 | **07/14/2015** | * Initial Draft |
| 3.2 | **05/10/2016** | * Updated to version 3.2 |
| 3.3 | **02/15/2017** | * Updated to version 3.3 |
| 4.0 | **02/19/2018** | * Updated to version 4.0, Date Range Global Update, Microsoft Global bug fix |
| 4.1 | **12/10/2018** | * Updated to version 4.1, Add enum Other to AccountType, 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 version 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 | **10/14/2020** | * Updated to release 4.4, Deprecated type. Use accountType in accountBase. Documented meta usage. Added interestPostingFrequency, withholdingsInfoList, overDraftProtectionAccountId, overDraftProtectionAccountType, overDraftProtectionAccountSubType. Added enums to accountType for Installment, CommercialLoan, CommercialLineOfCredit, GeneralLedger. OtherLoan.   DeprecatedAccountAccountTypeList and AccountAccountTypePair.  Added Type AccountIdentificationList, AccountIdentification, AccountSubType. Added enums ChargedOff, ChargeOffClosed to DepositAccountStatus" |
| 4.5 | **04/02/2021** | * Updated to release 4.4 |
| 5.0 | **06/07/2021** | * Updated to release 5.0 * Account –Update PartyId to reference Common.xsd. * Added elements statementCode, statementGroup, statementDate from Loan. * Removed deprecation – element subtype, and types AccountAccountTypeList, AccountAccountTypePair. * AccountFilter - Update PartyId to reference Common.xsd. * SimpleValidationRequestFilter - Update PartyIdList to reference Common.xsd. |

# Overview of Account Opening USE CASE EXAMPLE

The following use case example provides a coherent set of transactions to initiate the creation of an account and provide the initial funding. The process starts with a SimpleValidationRequest to verify the initial information. The next step is to create the membership using the Relationship message. Once the membership exists then create the account using the Account message. The last step of the example funds the account using the Deposit Message. This example provides a complete sample set of messages and responses to simulate the function for a single account opening.

The CUFX architecture supports additional complexities not represented in this example. Depending on the systems involved a different order of transactions may be required and is also supported by CUFX.

# Any know Errors in the document

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

# Table of Contents

[Document Status 1](#_Toc73691971)

[Change Log 1](#_Toc73691972)

[Overview of Account Opening USE CASE EXAMPLE 2](#_Toc73691973)

[Any know Errors in the document 2](#_Toc73691974)

[Table of Contents 2](#_Toc73691975)

[Document Conventions 3](#_Toc73691976)

[CUFX API and Documentation Support 3](#_Toc73691977)

[Release 4.0 Global Update Notes 3](#_Toc73691978)

[Release 4.4 Global Update Notes 4](#_Toc73691979)

[Release 5.0 Global Update Notes 4](#_Toc73691980)

[Definitions related to the specification 4](#_Toc73691981)

[Data Elements 5](#_Toc73691982)

[Filters used when accessing the Document data 5](#_Toc73691983)

[Variables used when accessing the account object (but not part of the object) 5](#_Toc73691984)

[Account Object attributes 5](#_Toc73691985)

[Deposit Object attributes 5](#_Toc73691986)

[Validation services Data attributes 6](#_Toc73691987)

[General Best Practice 6](#_Toc73691988)

[Use Case 7](#_Toc73691989)

[SimpleValidationRequestMessage 8](#_Toc73691990)

[Overview 8](#_Toc73691991)

[Authenticaton 8](#_Toc73691992)

[simpleValidationRequest Details 8](#_Toc73691993)

[REST-JSON CREATE REQUEST-RESPONSE Example 9](#_Toc73691994)

[RelationshipMessage 12](#_Toc73691995)

[REST-JSON CREATE REQUEST-RESPONSE Example 12](#_Toc73691996)

[AccountMessage 17](#_Toc73691997)

[Service Message: Create account 17](#_Toc73691998)

[REST-JSON CREATE REQUEST-RESPONSE 17](#_Toc73691999)

[depositMessage 20](#_Toc73692000)

[Deposit Resource Based Create, Read, Update, Delete Deposit Account data 20](#_Toc73692001)

[REST-JSON CREATE DEPOSIT 20](#_Toc73692002)

[General Error handling For All Services 23](#_Toc73692003)

[Bibliography 23](#_Toc73692004)

# 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.

# Release 5.0 Global Update Notes

CUFX Release 5.0 Moved PartyId and PartyIdList into the Common.xsd. Namespaces references to Party were removed if there were no other references to Party in the specification. PartyId and PartyIdList type references were updated.

# Definitions related to the specification

AccountData

The CUFX Account Data model and Services defines the account data model and Services used by all specifications. Accounts are an abstract summary of loan, share and investment balances at a high level. Any financial product stored in a financial services platform optionally including any transactions associated with that account.

Deposit Account

A liability for a financial institution to its clients.

**Party**

Any person or entity which may be in the process of researching products and services at a financial institution or may have an existing account relationship with the financial institution and stored in a financial services platform. Typically this is limited to entities that have an SSN or TIN created for them.

# Data Elements

## Filters used when accessing the Document 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. Refer to recent CUFX messageContext Data and CUFX Security Services for use of MessageContext.xsd.

Include any filter variables related to the request.

## Variables used when accessing the account object (but not part of the object)

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 **AccountFilter.xsd.** Note that the fields for filtering for account detail and account are identical.

The filters are used to filter based on associated data type for the account detail such as accounts for an account id, account type, or transaction range. etc.. Combining filters can be used to get list of accounts for a specific relationship, party, etc..

## Account Object attributes

All CUFX fields related to an account are defined in **Account.xsd**. Note that some fields are variable based on the accountType. These variations are defined in **Meta.xsd**. Further definitions for the account will be in the specific product type definition file such as **Loan.xsd** and **Deposit.xsd**. ***Note that Loan.xsd and Deposit.xsd inherit from Account.xsd and therefore all the fields within Account.xsd are available to the Loan.xsd and deposit.xsd as well***.

## Deposit Object attributes

* Entity Relationships: Refer to CUFX Entity relationship Diagram for complete list of entities that relate to the deposit entity

Deposit

Party Relationship (see party relationship services)

ATM/Debit Card (See card services)

Overdraft Protection (with priority)

Funding

## Validation services Data attributes

All CUFX fields related to a validation are defined in

* SimpleValidationRequest.xsd
* CreditReportRequest.xsd
* CreditReport.xsd
* CreditReportFIlter.xsd
* QuestionList.xsd
* Answer.xsd
* ValidationStatus.xsd

## General Best Practice

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 API

1

SimpleValidationRequest Message

2

[ValidationStatus](file:///C:\Users\brianl.STCU\Downloads\ValidationStatus.html)

contains confirmation or error

3

Relationship Message

4

Relationship Creation Data Response contains confirmation or error

CUFX Core Data Provider

5

Account Message

6

Account Creation Data Response

contains confirmation or error

7

Deposit Funding Message

8

Deposit Funding Data Response

contains confirmation or error

## SimpleValidationRequestMessage

### Overview

|  |  |
| --- | --- |
| Definition | Provides simple pass/fail response for the requested validation type. |
| Overview of Capabilities | Verify party information or qualifications. This may be for example checking against the OFAC list, retrieving Credit Bureau etc. and returning simple pass/fail response. Multiple services may implement this interface. |
| Dependencies | *Security Services* for authentication and security. *Party Services* for party data definition. *Error* specifications. |
| CUFX REST LINK | https://api.serviceprovider.com/simplevalidationrequestmessage |
| CUFX WSDL LINK |  |

### Authenticaton

The service provider will use a mechanism from the security services specification to authenticate the application.

### simpleValidationRequest Details

|  |  |
| --- | --- |
| Inputs | cufx: simpleValidationRequestMessage (which includes)   * cufx:MessageContext * cufx: simpleValidationRequestList |
| Outputs | cufx: simpleValidationRequestMessage (which includes)   * cufx:MessageContext   indeterminate signals an automated decision could not be made |
| Return Values | cufx: simpleValidationRequestMessage (which includes)   * cufx:MessageContext   + statusList |
| Service Specific Error Conditions | If the service provider requires a field in applicant that is not provided, the service provider should include the name of all missing required fields in the error. |
| Side Effects | None |
| Dependencies | Security Services for authentication and security.  Member Services for party data definition |
| CUFX REST LINK | https://api.serviceprovider.com/simplevalidationrequestmessage |
| CUFX WSDL LINK |  |
| Testing Procedures for Certification | Access the CUFX test service with a test data set that should pass and verify that it passed. Access the CUFX test service with a test data set that should fail and verify that it failed. Access the CUFX test service with a test data set that should return an error and verify that it returns an error and all missing fields are returned in the error object. |
| Derivative Services | None. |
| Fields Used | Message Headers : See security services  messageContext: See messageContext.xsd  Attributes: simpleValidationRequest : See simpleValidationRequest.xsd  Errors: See MessageContext.xsd |

### REST-JSON CREATE REQUEST-RESPONSE Example

Consumers submit validation request to the service providers validation url. For all CUFX RESTful-JSON requests, the client must set these headers:

**REQUEST:**

Headers:

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

Accept: application/json

Accept-Charset: utf-8

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

X-API-Version: >=5.0.0

**PUT** https://api.serviceprovider.com/simplevalidationrequestmessage

{

"simpleValidationRequestMessage": {

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

"messageContext": {

"requestId": "requestId1",

"vendorId": "vendorId1",

"appId": "appId1",

"fiId": "fiId1",

"dataSourceId": "dataSourceId1",

"environment": "Development",

"returnDataFilter": "All",

"includeBlankFields": "true",

"includeZeroNumerics": "true",

"user": [

{

"userId": "userId1",

"userType": "SecurityToken"

},

{

"userId": "NewMemberAdmin",

"userType": "SystemAccountId"

}

],

},

"simpleValidationRequestList": {

"simpleValidationRequest": [

{

"validationType": "IDRiskCheck",

"applicant": {

"role": "Primary",

"party": {

"irs": {

"taxId": "123456728",

"taxIdEncrypted": "false",

"taxIdType": "SocialSecurityNumber",

"reportingFlag": "true",

"verifiedTaxIdFlag": "false",

"backupWithholdingFlag": "false”

},

"type": "Individual",

"characteristics": {

“individual”:

{“title”:”General”,

“firstName”:”Thomas”,

“middleName”:”R”,

“lastName”:”Thumb”,

“mothersMaidenName”:”Toe”,

“nickname”:”Tom”,

“birthdate”:”1938-01-04”

“citizenshipList”:[

{“citizenship”:”USA”}

},

“employmentStatus”=”Contract”,

“employmentList:[

{“employerName”:”Barnum Circus”

“employmentStartDate”:”1953-04-01”,

“employeeOccupation”:”Entertainer”,

“incomeDetail”:{

“grossIncomeData”:{

“amount”:{“value”:15.00},

“frequency”:“Weekly”

}

}

]

}

}

},

"identificationDocumentList": {

"identificationDocument": [

{

"idDocumentIdentifer": "idDocumentIdentifer1",

"idDocumentType": {

"individualDocument": "DriversLicense"

},

"idIssuedBy": "idIssuedBy1",

"idIssueDate": "1954-01-01",

"idExpirationDate": "2017-01-01",

"idDisplayOrder": "0",

"idVerifyDateTime": "1954-01-01T01:01:01Z",

"documentId": "documentId1"

}

]

},

"eligibilityRequirementMetList": {

"eligibilityRequirementMet": [

{

"requirementId": "123",

"referenceDescription": "Barnum Employee",

},

]

},

"contactList": {

"contact": [

{

"contactType": "Address",

"address": {

"line1": "123 Main St",

"line2": "Apt B",

"city": "New Haven",

"stateProvince": "CT",

"postalCode": "06501",

"countryCode": "US",

"country": "USA",

"type": "Home",

"timeAtAddress": "P396DT1H1M1S",

"addressVerifyDateTime": "2014-01-21T01:01:01Z"

},

"timeOfDay": "Afternoon",

"demonstratedAccess": {

"dateTime": "2014-01-21T01:01:01Z",

"fullName": "Tom Thumb",

"userName": "PTomThumb1"

},

"badContactPoint": "false",

},

{

"contactType": "Phone",

"phone": {

"number": "1-303-456-6763",

"planFormat": “Nanp”,

"description": "Home Phone",

"smsRegistered": “false”

}

}

},

{

"contactType": "Email",

"email": {

"type": "Home",

"address": "tthumb@comcast.net"

}

}

]

}

}

}

}

]

}

}

}

RESPONSE:

Headers:

Status Code: 200 Ok

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

Content-Language: en-us

Payload:

{

"validationStatus": {

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

"#text": "Pass"

}

}

## RelationshipMessage

|  |  |
| --- | --- |
| INPUTS | cufx:relationshipMessage (which includes)   * cufx:messageContext * cufx:relationshipFilter (read, update) * cufx:relationshipList (for create, update, delete) |
| Outputs | cufx:relationshipMessage (which includes)   * cufx:messageContext * cufx:relationshipList (for create, update, delete) |
| Return Values | cufx:relationshipMessage (which includes)   * cufx:MessageContext   + 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 RelationshipFilter.xsd  **Attributes:** relationshipList: See Relationship.xsd  **Errors:** See MessageContext.xsd |

### REST-JSON CREATE REQUEST-RESPONSE Example

Accept: application/json

Accept-Charset: utf-8

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

X-API-Version: >=5.0.0

PUT https://api.serviceprovider.com/relationshipmessage

{

"relationshipMessage": {

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

"messageContext": {

"requestId": "requestId1",

"vendorId": "vendorId1",

"appId": "appId1",

"fiId": "fiId1",

"dataSourceId": "dataSourceId1",

"environment": "Development",

"returnDataFilter": "All",

"includeBlankFields": "true",

"includeZeroNumerics": "true",

"user": [

{

"userId": "userId1",

"userType": "SecurityToken"

},

{

"userId": "NewMemberAdmin",

"userType": "SystemAccountId"

}

],

"customData": {

"valuePair": [

{

"name": "name1",

"value": "value1"

}

]

}

},

"relationshipList": {

"relationship": [

{

"dateCreated": "2015-05-01",

"rewardsCode": "rewardsCode1",

"relationshipPartyList": {

"relationshipParty": [

{

"irs": {

"taxId": "123456728",

"taxIdEncrypted": "false",

"taxIdType": "SocialSecurityNumber",

"reportingFlag": "true",

"verifiedTaxIdFlag": "false",

"backupWithholdingFlag": "false"

},

"type": "Individual",

"characteristics": {

“individual”:

{“title”:”General”,

“firstName”:”Thomas”,

“middleName”:”R”,

“lastName”:”Thumb”,

“mothersMaidenName”:”Toe”,

“nickname”:”Tom”,

“birthdate”:”1838-01-04”

“citizenshipList”:[

{“citizenship”:”USA”}

},

“employmentStatus”=”Contract”,

“employmentList:[

{“employerName”:”Barnum Circus”

“employmentStartDate”:”1843-04-01”,

“employeeOccupation”:”Entertainer”,

“incomeDetail”:{

“grossIncomeData”:{

“amount”:{“value”:15.00},

“frequency”:“Weekly”

}

}

]

}

}

},

"identificationDocumentList": {

"identificationDocument": [

{

"idDocumentIdentifer": "idDocumentIdentifer1",

"idDocumentType": {

"individualDocument": "DriversLicense"

},

"idIssuedBy": "idIssuedBy1",

"idIssueDate": "1954-01-01",

"idExpirationDate": "2017-01-01",

"idDisplayOrder": "0",

"idVerifyDateTime": "1954-01-01T01:01:01Z",

"documentId": "documentId1"

}

]

},

"eligibilityRequirementMetList": {

"eligibilityRequirementMet": [

{

"requirementId": "123",

"referenceDescription": "Barnum Employee",

},

]

},

"contactList": {

"contact": [

{

"contactType": "Address",

"address": {

"line1": "123 Main St",

"line2": "Apt B",

"city": "New Haven",

"stateProvince": "CT",

"postalCode": "06501",

"countryCode": "US",

"country": "USA",

"type": "Home",

"timeAtAddress": "P396DT1H1M1S",

"addressVerifyDateTime": "2014-01-21T01:01:01Z"

},

"timeOfDay": "Afternoon",

"demonstratedAccess": {

"dateTime": "2014-01-21T01:01:01Z",

"fullName": "Tom Thumb",

"userName": "PTomThumb1"

},

"badContactPoint": "false",

},

{

"contactType": "Phone",

"phone": {

"number": "1-303-456-6763",

"planFormat": “Nanp”,

"description": "Home Phone",

"smsRegistered": “false”

}

}

},

{

"contactType": "Email",

"email": {

"type": "Home",

"address": "tthumb@comcast.net"

}

}

]

}

}

},

}

},

}

]

}

}

}

RESPONSE:

Headers:

Status Code: 200 Ok

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

Content-Language: en-us

Payload:{

{

"relationshipMessage": {

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

"messageContext": {

"requestId": "requestId1",

"vendorId": "vendorId1",

"appId": "appId1",

"fiId": "fiId1",

"dataSourceId": "dataSourceId1",

"environment": "Development",

"returnDataFilter": "All",

"includeBlankFields": "true",

"includeZeroNumerics": "true",

"user": [

{

"userId": "userId1",

"userType": "SecurityToken"

},

{

"userId": "NewMemberAdmin",

"userType": "SystemAccountId"

}

],

"customData": {

"valuePair": [

{

"name": "name1",

"value": "value1"

}

]

}

},

"relationshipList": {

"relationship": [

{

"dateCreated": "2015-05-01",

"rewardsCode": "rewardsCode1",

"relationshipPartyList": {

"relationshipParty": [

{

"irs": {

"taxId": "123456728",

"taxIdEncrypted": "false",

"taxIdType": "SocialSecurityNumber",

"reportingFlag": "true",

"verifiedTaxIdFlag": "false",

"backupWithholdingFlag": "false"

},

"type": "Individual",

"characteristics": {

“individual”:

{“title”:”General”,

“firstName”:”Thomas”,

“middleName”:”R”,

“lastName”:”Thumb”,

“mothersMaidenName”:”Toe”,

“nickname”:”Tom”,

“birthdate”:”1838-01-04”

“citizenshipList”:[

{“citizenship”:”USA”}

},

“employmentStatus”=”Contract”,

“employmentList:[

{“employerName”:”Barnum Circus”

“employmentStartDate”:”1843-04-01”,

“employeeOccupation”:”Entertainer”,

“incomeDetail”:{

“grossIncomeData”:{

“amount”:{“value”:15.00},

“frequency”:“Weekly”

}

}

]

}

}

},

"identificationDocumentList": {

"identificationDocument": [

{

"idDocumentIdentifer": "idDocumentIdentifer1",

"idDocumentType": {

"individualDocument": "DriversLicense"

},

"idIssuedBy": "idIssuedBy1",

"idIssueDate": "1954-01-01",

"idExpirationDate": "2017-01-01",

"idDisplayOrder": "0",

"idVerifyDateTime": "1954-01-01T01:01:01Z",

"documentId": "documentId1"

}

]

},

"eligibilityRequirementMetList": {

"eligibilityRequirementMet": [

{

"requirementId": "123",

"referenceDescription": "Barnum Employee",

},

]

},

"contactList": {

"contact": [

{

"contactType": "Address",

"address": {

"line1": "123 Main St",

"line2": "Apt B",

"city": "New Haven",

"stateProvince": "CT",

"postalCode": "06501",

"countryCode": "US",

"country": "USA",

"type": "Home",

"timeAtAddress": "P396DT1H1M1S",

"addressVerifyDateTime": "2014-01-21T01:01:01Z"

},

"timeOfDay": "Afternoon",

"demonstratedAccess": {

"dateTime": "2014-01-21T01:01:01Z",

"fullName": "Tom Thumb",

"userName": "PTomThumb1"

},

"badContactPoint": "false",

},

{

"contactType": "Phone",

"phone": {

"number": "1-303-456-6763",

"planFormat": “Nanp”,

"description": "Home Phone",

"smsRegistered": “false”

}

}

},

{

"contactType": "Email",

"email": {

"type": "Home",

"address": "tthumb@comcast.net"

}

}

]

}

}

},

}

},

}

"relationshipStatus" : “Active”

]

}

}

}

## AccountMessage

|  |  |
| --- | --- |
| INPUTS | Cufx:AccountMessage.xsd which includes   * [cufx:MessageContext](file:///\\files2\users\CMarjaniemi\Projects\CUFX\MessageContext.html).xsd * cufx:AccountFilter.xsd (for read) |
| Outputs | Cufx:AccountMessage.xsd which includes   * [cufx:MessageContext](file:///\\files2\users\CMarjaniemi\Projects\CUFX\MessageContext.html).xsd * cufx:AccountList.xsd (for read) |
| Return Values | cufx: accountMessage (which includes)   * cufx:MessageContext   + 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 AccountFilter.xsd  **Attributes: account : See Account.xsd**  **Errors:** See MessageContext.xsd |
| Testing Procedures for Certification | Access the CUFX account data test service and verify that test data is returned appropriately. |

### Service Message: Create account

### REST-JSON CREATE REQUEST-RESPONSE

Clients submit an account data request to the Data Provider’s URL. For all CUFX RESTful-JSON requests, the client must set the appropriate headers. This example does 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-API-Version: >=5.0.0

Post https://api.dataprovider.com/accountmessage

{

"accountMessage": {

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

"messageContext": {

"requestId": "requestId1",

"vendorId": "vendorId1",

"appId": "appId1",

"fiId": "fiId1",

"dataSourceId": "dataSourceId1",

"environment": "Development",

"returnDataFilter": "All",

"includeBlankFields": "true",

"includeZeroNumerics": "true",

"user": [

{

"userId": "userId1",

"userType": "SecurityToken"

},

{

"userId": "NewMemberAdmin",

"userType": "SystemAccountId"

}

],

}

"accountList": {

"account": [

{

"idType": "Reserved",

"description": "DefaultShare",

"openDate": "2015-03-18",

"accountNickName": "My Money",

"branch": "branch1",

"relationshipId": "relationshipId1",

}

]

}

}

}

RESPONSE:

Headers:

Status Code: 200 Ok

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

Content-Language: en-us

Payload:{

"accountMessage": {

"@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"

}

},

},

"accountList": {

"account": [

{

"accountId": "98765432334500",

"idType": "Reserved",

"description": "DefaultShare",

"openDate": "2015-03-18",

"accountNickName": "Thomas Savings Acct",

"branch": "branch1",

"relationshipId": "relationshipId1",

}

]

}

}

}

}

## depositMessage

## Deposit Resource Based Create, Read, Update, Delete Deposit Account data

|  |  |
| --- | --- |
| INPUTS | cufx: depositMessage (which includes)   * [cufx:MessageContext](file:///\\files2\users\CMarjaniemi\Projects\CUFX\MessageContext.html) * cufx:DepositFilter (for read, update, delete) * cufx:DepositList (for create, update) |
| Outputs | cufx: depositMessage (which includes)   * [cufx:MessageContext](file:///\\files2\users\CMarjaniemi\Projects\CUFX\MessageContext.html) * cufx:DepositList (for create, update) |
| Return Values | ufx: depositMessage (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  **Filters**: See DepositFilter.xsd  **Attributes: DepositList : See Deposit.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 DEPOSIT

In some cases the new membership application will require an account ID for use in the documentation for the member to sign, prior to the creation of the account on the core system.

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: >=5.0.0

POST https://api.serviceprovider.com/depositmessage

{

"depositMessage": {

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

"messageContext": {

"requestId": "requestId1",

"vendorId": "vendorId1",

"appId": "appId1",

"fiId": "fiId1",

"dataSourceId": "dataSourceId1",

"environment": "Development",

"returnDataFilter": "All",

"includeBlankFields": "true",

"includeZeroNumerics": "true",

"user": [

{

"userId": "userId1",

"userType": "SecurityToken"

},

{

"userId": "NewMemberAdmin",

"userType": "SystemAccountId"

}

],

"customData": {

"valuePair": [

{

"name": "name1",

"value": "value1"

}

]

}

},

"depositList": {

"deposit": [

{

"idType": "Actual",

"description": "SAVINGS ACCOUNT",

"type": "Savings",

"subType": "subType1",

"accountNickName": "Thomas Savings Acct",

"externalAccountFlag": "false",

"externalTransferFrom": "true",

"externalTransferTo": "true",

"transferFrom": "true",

"transferTo": "true",

"sourceCode": "NewMembership",

"partyIdList": {

"partyId": [

"partyId1"

]

},

"relationshipId": "relationshipId1",

},

"transactionList": {

"transaction": [

{

"transactionId": "transactionId1",

"accountId": "98765432334500",

"type": "Credit",

"amount": {

"value": "1492.38",

"currencyCode": "USD

},

"description": "Pay Check Deposit",

"checkNumber": "3456",

"dateTimeEffective": "2015-06-03T01:01:01Z",

"source": "Check",

}

}

}

}

]

}

}

}

RESPONSE:

Headers:

Status Code: 200 Ok

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

Content-Language: en-us

Payload:

{"depositMessage": {

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

"messageContext": {

"requestId": "requestId1",

"vendorId": "vendorId1",

"appId": "appId1",

"fiId": "fiId1",

"dataSourceId": "dataSourceId1",

"environment": "Development",

"returnDataFilter": "All",

"includeBlankFields": "true",

"includeZeroNumerics": "true",

"user": [

{

"userId": "userId1",

"userType": "SecurityToken"

},

{

"userId": "NewMemberAdmin",

"userType": "SystemAccountId"

}

],

},

"depositList": {

"deposit": [

{

"idType": "Actual",

"description": "SAVINGS ACCOUNT",

"type": "Savings",

"subType": "subType1",

"accountNickName": "Thomas Savings Acct",

"externalAccountFlag": "false",

"externalTransferFrom": "true",

"externalTransferTo": "true",

"transferFrom": "true",

"transferTo": "true",

"sourceCode": "NewMembership",

"partyIdList": {

"partyId": [

"partyId1"

]

},

"relationshipId": "relationshipId1",

},

"transactionList": {

"transaction": [

{

"transactionId": "87654",

"accountId": "98765432334500",

"type": "Credit",

"amount": {

"value": "1492.38",

"currencyCode": "USD

},

"description": "Pay Check Deposit",

"checkNumber": "3456",

"dateTimeEffective": "2015-06-03T01:01:01Z",

"source": "Check",

}

}

}

}

]

}

}

}

# 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.