

Deposit Data 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)

# Authors and Change Log

|  |  |  |
| --- | --- | --- |
| Version | Date | Changes |
| 0.0.01 |  | * Initial Creation |
| 0.0.02 |  | * Added visual diagram |
| 0.0.03 |  | * Deleted references to appVendorToken and added messageContext |
| 0.0.04 |  | * Updated versioning |
| 0.0.05 |  | * Deleted remaining instance of appVendorToken |
| 0.0.06 |  | * Reference XSDs, add Overview, remove redundant errors table |
| 0.0.07 |  | * Removed data elements that have been implemented in XSDs |
| 0.0.08 |  | * Added links to elements that have been implemented in XSDs * Added services methods for Card and DepositFunding |
| 0.0.09 |  | * Cleaned up general formatting * Switched URI from account to deposit * Added read examples with Filters and Lists being returned for deposit, Overdraft Priority and Deposit Funding examples. |
| 0.0.10 |  | * Added nextDepositId service to reserve account IDs prior to account creation to facilitate generating legal documents required to open the account |
| 0.0.11 |  | * Change in account id reservation process. |
| 0.0.12 |  | * Added missing subMethods. |
| 0.0.13 |  | * Cleaned up JSON. |
| 0.0.14 |  | * Fixed filter list iteration. |
| 0.0.15 |  | * Added new overdraft filter, new overdraft ID, and some JSON example fixes. |
| 0.0.16 |  | * Added page footer and changed TOC format. * Changed id to accountId based on field in Account.xsd * Changed examples for id lists and fixed sourceOfFunds * Reordered some fields to match XSD order. |
| 0.0.17 |  | * Switch to use X-HTTP-METHOD-OVERRIDE standard rather than subMethod non-Standard method for overriding request types. * Create a depositMessage wrapper for every message to increase ability for infrastructure to serialize the data * Converted reservation and creates to a POST (had an override of PUT) |
| 3.0 | **10/29/2013** | * Versioning and format change with release CUFX 3.0 |
| 3.0 | **12/13/2013** | * 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, maturityDateRange maxOccurs set to unbounded. Restructed AccountStatus to choice of Deposit,Loan, and Investment. Consolidated to Acccount.xsd. |
| 4.1 | **12/10/2018** | * Updated to release 4.1, * Global removal of choice constructs to eliminate cross language serialization issues. * Global update applied for minOccurs 0 maxOccurs 1. Xsd had 1 or more elements corrected to be consistent with the standard definition. |
| 4.2 | **03/05/2019** | * Updated to release 4.2, \*\*\* Release 4.2 is a breaking fix release. \*\*\* Errors found in App, ArtifactFilter, and BillFilter required a breaking fix to align with the standard and prevent additional implementation difficulties going forward. |
| 4.3 | **10/07/2019** | * Updated to release 4.3, renamed file removing version as proper version control is being used in Github. |
| 4.4 | **10/20/2020** | * Account - 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" * Account Filter - includeNotesFlag type to common:IncludeNotesFlag. Added accountIdentificationList * Deposit - Standalone element Deposit marked as deprecated. * Added maturityTransferAmount, maturityTransferPercent, serviceChargeWaivedReason. * Added enum TransferAndRenew to MaturityPostCodeType * Added customData to depositParty. |
| 4.5 | **04/02/2021** | * Updated to release 4.5 |

# Overview of Specification

This specification defines the features for creating share (deposit) accounts, including the ability to:

1. Create a new share, draft or certificate account
2. Specify overdraft priority
3. Specify a relationship between an account and a party
4. Order a Debit/ATM card
5. Fund a new account

# Any know Errors in the document

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

Table of Contents

[Document Status 1](#_Toc68098588)

[Authors and Change Log 1](#_Toc68098589)

[Overview of Specification 2](#_Toc68098590)

[Any know Errors in the document 3](#_Toc68098591)

[Document Conventions 3](#_Toc68098592)

[CUFX API and Documentation Support 4](#_Toc68098593)

[Release 4.0 Global Update Notes 4](#_Toc68098594)

[Release 4.4 Global Update Notes 4](#_Toc68098595)

[Definitions related to the specification 5](#_Toc68098596)

[Data Elements 5](#_Toc68098597)

[Use Case 6](#_Toc68098598)

[Services 7](#_Toc68098599)

[Authentication 7](#_Toc68098600)

[Deposit Resource Based Create, Read, Update, Delete Deposit Account data 7](#_Toc68098601)

[REST-JSON RESERVE DEPOSIT ACCOUNT ID 7](#_Toc68098602)

[REST-JSON CREATE DEPOSIT ACCOUNT 8](#_Toc68098603)

[REST-JSON READ DEPOSIT ACCOUNT 9](#_Toc68098604)

[Overdraft priority Resource Based Create, Read, Update, Delete Data 11](#_Toc68098605)

[REST-JSON Create Overdraft priority 11](#_Toc68098606)

[REST-JSON READ Overdraft priority 12](#_Toc68098607)

[ATM/Debit Card Resource Based Create, Read, Update, Delete Data 13](#_Toc68098608)

[Deposit Funding Resource Based Create, Read, Update, Delete Data 13](#_Toc68098609)

[REST-JSON Create Deposit Funding Example 13](#_Toc68098610)

[REST-JSON READ Deposit Funding Example 14](#_Toc68098611)

[General Error handling For All Services 14](#_Toc68098612)

[Bibliography 15](#_Toc68098613)

# 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

Deposit Account

A liability for a financial institution to its clients.

# Data Elements

* 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

# Use Case

CUFX Core Data Provider

1

Deposit Creation Request

2

Deposit Creation Data Response Contains Deposit Id or error

CUFX Compliant New Member Account Application

3

PartyRelationship Creation Request Mapped to Share ID

4

PartyRelationship Creation Data Response contains confirmation or error

5

Overdraft Creation

Share ID and priority

6

Overdraft Creation Data Response contains confirmation or error

10

Funding Data Response (Funding API) contains confirmation or error

9

Funding Instructions (Funding API) share holds and source identification

8

ATM/Debit Card Creation Data Response contains confirmation or error

7

ATM/Debit Card Creation (Card API)

# Services

|  |  |
| --- | --- |
| Definition | Deposit account |
| Overview of Capabilities | Service to create deposit accounts and set dependent data. |
| Dependencies | Security Services, messageContext |
| CUFX REST LINK | <https://api.dataprovider.com/deposit>message |
| CUFX WSDL LINK |  |

1. Create a new share, draft or certificate account
   1. Inputs: party number & account object
2. Specify overdraft priority
3. Specify a relationship between an account and a party
4. Order a Debit/ATM card (see card services documentation)
5. Fund a new account

## Authentication

See Security Services and messageContext for more detail.

## 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) * cufx:DepositList (for create, update, delete) |
| Outputs | cufx: depositMessage (which includes)   * [cufx:MessageContext](file:///\\files2\users\CMarjaniemi\Projects\CUFX\MessageContext.html) * cufx:DepositList |
| Return Values | cufx: 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 |
| 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 RESERVE DEPOSIT ACCOUNT ID

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: >=4.5.0

**POST** <https://api.dataprovider.com/deposit>message

{

“depositMessage”:{

"messageContext" : <See MessageContext.xsd>

,

"depositList" : [{

"idtype":"Reserved",

"type":"Certificate"

}]

}

}

RESPONSE

Headers:

Status Code: 200 Ok

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

Content-Language: en-us

Payload:

{“depositMessage”:{

"messageContext" : <See MessageContext.xsd>

,

“depositList”:[{

"accountId":"1357853634817d",

"idtype":"Reserved",

"type":"Certificate"

}]

}

}

### REST-JSON CREATE DEPOSIT ACCOUNT

This example shows call creates a new account.

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/depositmessage

{

“depositMessage”:{

“messageContext”: { <see messageContext.xsd>

},

“depositList”:[{

“description”:”CUFX CD Special”,

“type”:”Investment”,

“subType”:”1534”,

“openDate”:”2012-05-22”,

“branch”:”1234”,

“minimumBalance”:{“value”:1000.00},

“rateType”: “Fixed”,

“sourceCode”:”OnlineBanking”,

“partyIdList”:[“123456”],

“meta”: {

“startingBalance”:{“value”:2000.00}

“maturityDate”:”2012-11-22”,

“interestRate”: 1.234,

“compoundingFrequency”:”Monthly”,

},

“bumpEffectiveDate”:”2013-12-31”,

“bumpExpirationDate”:”2014-12-31”,

“bumpRate”:3.000,

“dividendPostCode”:”ToAccount”,

“dividendRate”:2.500,

“dividendType”:”12”,

“maturityPostCode”:”Renew”,

“minimumDeposit”:{“value”:2000.00}

}]

}

}

RESPONSE

Headers:

Status Code: 200 Ok

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

Content-Language: en-us

Payload:

“depositMessage”:{

"messageContext" : <See MessageContext.xsd>

,

“depositList”:[{

“accountId”:”1357853634817d”

“description”:”CUFX CD Special”,

“type”:”Investment”,

“subType”:”1534”,

“openDate”:”2012-05-22”,

“branch”:”1234”,

“minimumBalance”:{“value”:1000.00},

“rateType”: “Fixed”,

“sourceCode”:”OnlineBanking”,

“partyIdList”:[“123456”],

“meta”: {

“startingBalance”:{“value”:2000.00}

“maturityDate”:”2012-11-22”,

“interestRate”: 1.234,

“compoundingFrequency”:”Monthly”,

},

“bumpEffectiveDate”:”2013-12-31”,

“bumpExpirationDate”:”2014-12-31”,

“bumpRate”:3.000,

“dividendPostCode”:”ToAccount”,

“dividendRate”:2.500,

“dividendType”:”12”,

“maturityPostCode”:”Renew”,

“minimumDeposit”:{“value”:2000.00}

]

}

}

}

### REST-JSON READ DEPOSIT ACCOUNT

This example shows reading a deposit account.

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/depositmessage

{

“depositMessage”:{

“messageContext”: { <see messageContext.xsd>

},

“depositFilter”:{

“accountIdList”:[”1357853634817d”]

}

}

}

RESPONSE

Headers:

Status Code: 200 Ok

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

Content-Language: en-us

Payload:

{

“depositMessage”:{

“messageContext”: { <see messageContext.xsd>

},

“depositList”:[{

“accountId”:”1357853634817d”

“description”:”CUFX CD Special”,

“type”:”Investment”,

“subType”:”1534”,

“openDate”:”2012-05-22”,

“branch”:”1234”,

“minimumBalance”:{“value”:1000.00},

“rateType”: “Fixed”,

“sourceCode”:”OnlineBanking”,

“partyIdList”:[“123456”],

“meta”: {

“startingBalance”:{“value”:2000.00}

“maturityDate”:”2012-11-22”,

“interestRate”: 1.234,

“compoundingFrequency”:”Monthly”,

},

“bumpEffectiveDate”:”2013-12-31”,

“bumpExpirationDate”:”2014-12-31”,

“bumpRate”:3.000,

“dividendPostCode”:”ToAccount”,

“dividendRate”:2.500,

“dividendType”:”12”,

“maturityPostCode”:”Renew”,

“minimumDeposit”:{“value”:2000.00}

]

}

}

}

## Overdraft priority Resource Based Create, Read, Update, Delete Data

|  |  |
| --- | --- |
| INPUTS | cufx:OverdraftPriorityMessage (which includes)   * [cufx:MessageContext](file:///\\files2\users\CMarjaniemi\Projects\CUFX\MessageContext.html) * cufx:OverdraftPriorityFilter (for read, update) * cufx:OverdraftPriority (for create, update, delete) |
| Outputs | cufx:OverdraftPriorityMessage (which includes)   * [cufx:MessageContext](file:///\\files2\users\CMarjaniemi\Projects\CUFX\MessageContext.html) * cufx:OverdraftPriority |
| Return Values | cufx:OverdraftPriorityMessage (which includes)   * cufx:MessageContext   + statusList |
| Side Effects | Creation, update or deletion of overdraft priority records for a deposit account, 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 OverdraftPriorityFilter.xsd  Attributes: OverdraftPriority : SeeOverdraftPriorityList**.**xsd |
| Testing Procedures for Certification | Send details for a new overdraft priority object and verify that it has been created. |

### REST-JSON Create Overdraft priority

Description: Set overdraft priority for an account(s)

Overview: Establishes the order of accounts used to cover overdrafts on a given existing account.

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/overdraftprioritymessage

{

“overdraftPriorityMessage”: {

“messageContext”: { <see messageContext.xsd>

},

“overdraftPriorityList”:[

{“accountId”:”13534134555”,

[

{"priority": 1, "accountid":"123534"},

{“priority”: 2, "accountid":"1235555"}

]

}

]

}

}

RESPONSE

Headers:

Status Code: 200 Ok

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

Content-Language: en-us

“overdraftPriorityMessage”: {

“messageContext”: { <see messageContext.xsd>

},

“overdraftPriorityList”:[

{“accountId”:”13534134555”,

[

{“overdraftPriorityId”: “1”, "priority": 1, "accountid":"123534"},

{“overdraftPriorityId”: “2”, “priority”: 2, "accountid":"1235555"}

]

}

]

}

}

### REST-JSON READ Overdraft priority

Description: Read overdraft priority for an account(s)

Overview: Read the order of accounts used to cover overdrafts on a given existing account.

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/overdraftprioritymessage

{

“overdraftPriorityMessage”: {

“messageContext”: { <see messageContext.xsd> },

“overdraftPriorityListFilter”:{

“accountIdList”:[”13534134555”]

}

}

}

Headers:

Status Code: 200 Ok

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

Content-Language: en-us

“overdraftPriorityMessage”: {

“messageContext”: { <see messageContext.xsd>

},

“overdraftPriorityList”:[

{“accountId”:”13534134555”,

[

{“overdraftPriorityId”: “1”, "priority": 1, "accountId":"123534"},

{“overdraftPriorityId”: “2”, “priority”: 2, "accountId":"1235555"}

]

}

]

}

}

## ATM/Debit Card Resource Based Create, Read, Update, Delete Data

See the latest CUFX Card services document for how to create cards.

## Deposit Funding Resource Based Create, Read, Update, Delete Data

|  |  |
| --- | --- |
| INPUTS | cufx: depositFundingMessage (which includes)   * [cufx:MessageContext](file:///\\files2\users\CMarjaniemi\Projects\CUFX\MessageContext.html) * cufx:DepositFilter (for read, update, delete) * cufx: depositFunding (for create, update) |
| Outputs | cufx: depositFunding (for create, read, update) |
| Return Values | cufx: Error |
| Side Effects | Creation, update or deletion of deposit funding records for a deposit account, 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: DepositFundingList** **: See**  **DepositFunding.xsd** |
| Testing Procedures for Certification | Send details for a new deposit funding and verify that it has been created. |
| CUFX RESET LINK | https://api.dataprovider.com/depositfundingmessage |

### REST-JSON Create Deposit Funding Example

This call creates a new deposit funding record.

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/depositfundingmessage

{

“depositFundingMessage”:{

“messageContext”: { <see messageContext.xsd> },

“depositFundingList”:[

{“targetAccountId”:“1234”,

“fundingSourceId”:“4321”,

“sourceOfFunds”:“ACH”,

“fundingAmount”:{“value”: “10.00”,“currencyCode”: “USD”},

“holdAmount”:{“value”: “10.00”,“currencyCode”: “USD”},

“holdExpirationDate”:”2013-12-31”

}

]

}

}

RESPONSE:

Headers:

Status Code: 200 Ok

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

Content-Language: en-us

Payload:

{

“depositFundingList”:[

{“targetAccountId”:“1234”,

“fundingSourceId”:“4321”,

“sourceOfFunds”:“ACH”,

“fundingAmount”:{“value”: “10.00”,“currencyCode”: “USD”},

“holdAmount”:{“value”: “10.00”,“currencyCode”: “USD”},

“holdExpirationDate”:”2013-12-31”

}

]

}

### REST-JSON READ Deposit Funding Example

This call reads new deposit funding record.

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/depositfundingmessage

{

“depositFundingMessage”:{

“messageContext”: { <see messageContext.xsd> },

“depositFilter”: {

“accountIdList”:[“1234”]

}

}

}

RESPONSE:

Headers:

Status Code: 200 Ok

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

Content-Language: en-us

Payload:

{

“depositFundingList”:[

{“targetAccountId”:“1234”,

“fundingSourceId”:“4321”,

“sourceOfFunds”:“ACH”,

“fundingAmount”:{“value”: “10.00”,“currencyCode”: “USD”},

“holdAmount”:{“value”: “10.00”,“currencyCode”: “USD”},

“holdExpirationDate”:”2013-12-31”

}

]

}

# 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.