

Collateral Data Model and Services 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 |
| 0.0.1 |  | * Initial Creation |
| 0.0.2 |  | * Switch to use X-HTTP-METHOD-OVERRIDE standard rather than subMethod non-Standard method for overriding request types. * Create a collateralMessage wrapper for every message to increase ability for infrastructure to serialize the data * Change Create method to accept a list so that it can create multiple collateral at the same time. * Change Delete method to accept a Filter so that it can delete multiple collateral at the same time. * Update Delete method to use DELETE override rather than GET |
| 3.0 | **10/29/2013** | * Versioning and format change with release CUFX 3.0 |
| 3.0 | **12/12/2013** | * 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 |
| 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** | * Updated to release 4.4, Collateral - Added CustomData to Collateral. Added CollateralType, CollteralSubType, collateralSubTypeDescriptionand accountIdentificationList * Added accountType, accountSubType and certificateNumber to SavingsCDCollateral. * Added type CollateralSubType, CollateralSubTypeList, CollateralType, CollateralTypeList * CollateralFilter - Added collateralTypeList, collateralSubTypeList, accountIdList, accountIdentificationList |
| 4.5 | **04/09/2021** | * Updated to release 4.5, Collateral - Added address to collateralBase |
| 5.0 | **06/07/2021** | * Updated to release 5.0 |

Overview of Specification

The Collateral specification describes the services used as part of a new membership application process, for creating, updating, and modifying collateral records.

Any know Errors in the document

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

Table of Contents

[Document Conventions 3](#_Toc73692757)

[CUFX API and Documentation Support 3](#_Toc73692758)

[Release 4.0 Global Update Notes 3](#_Toc73692759)

[Release 4.4 Global Update Notes 4](#_Toc73692760)

[Release 5.0 Global Update Notes 4](#_Toc73692761)

[Definitions related to the specification 4](#_Toc73692762)

[High level use cases 5](#_Toc73692763)

[Use Case 1: Adding a collateral item and associating it to a loan 5](#_Toc73692764)

[Data Elements 6](#_Toc73692765)

[Collateral Inheritence Hierarchy 6](#_Toc73692766)

[CollateralBase 6](#_Toc73692767)

[TitledCollateralBase 6](#_Toc73692768)

[PossessoryCollateralBase 7](#_Toc73692769)

[Collateral Services 7](#_Toc73692770)

[Service Definitions 7](#_Toc73692771)

[Service Message: ReadCollateral 7](#_Toc73692772)

[Service Message: CreateCollateral 9](#_Toc73692773)

[Service Message: UpdateCollateral 10](#_Toc73692774)

[Service Message: DeleteCollateral 12](#_Toc73692775)

[Bibliography 13](#_Toc73692776)

# 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

System of record

Authoritative data source for information such as the collateral for which this specification will be creating.

Collateral

Collateral is defined as something used to secure a loan. Examples of collateral are a car, boat, or Certificate of Deposit (CD).

# High level use cases

## Use Case 1: Adding a collateral item and associating it to a loan

1

CreateCollateral API

Contains all the details about the collateral

CUFX New Member Application

2

CreateCollateralResponse

Would return the CollateralId that was just created

3

CreateLoan API

Contains collateralList element with the CollateralId

CUFX Core Data Provider

4

Loan Creation Response (CreateLoanAPI) contains confirmation or error

3

Disburse loan funds (DisburseFunds API) loan disbursement information such as printing a check, transfers, cash

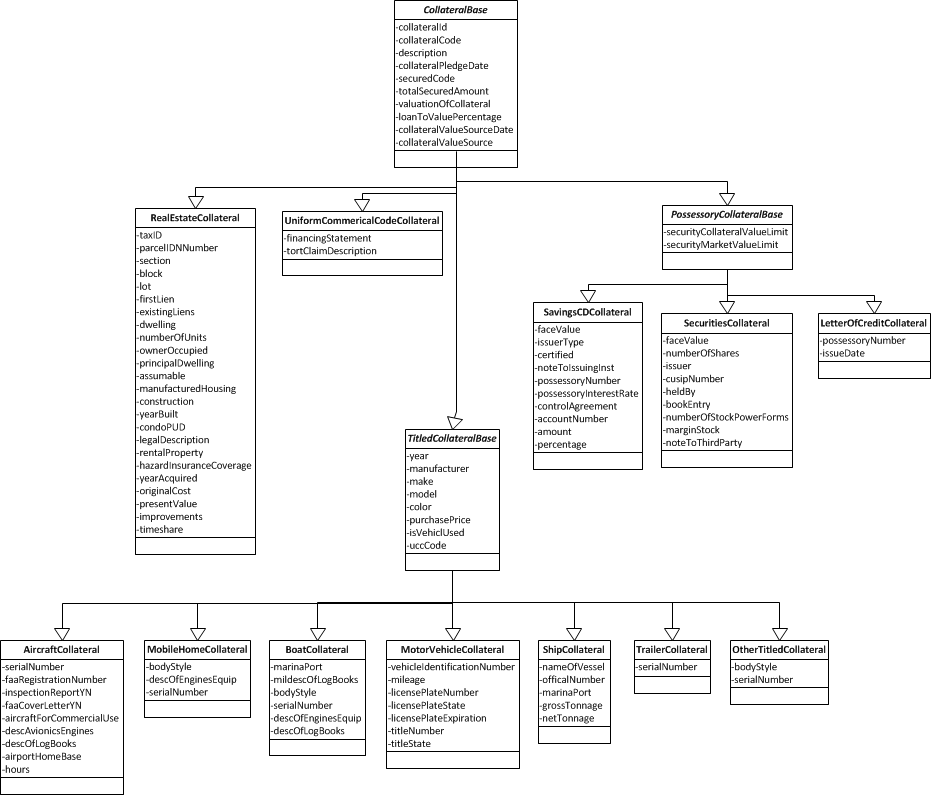
4

Disburse loan funds response (DisburseFunds API) contains confirmation or error

# Data Elements

## Collateral Inheritence Hierarchy

There are many different types of collateral that have many different attributes. To support the various types of collateral, the following inheritance hierarchy was created. Any object that ends with “Base”, is an abstract base type, and cannot be instantiated. For example, you cannot create a PossessoryCollateralBase object, but you can create a SecuritiesCollateral object which inherits from PossessoryCollateralBase.



### CollateralBase

This is an abstract base type which contains fields that are common to all collateral items, such as the collateralId, collateral description, etc.

### TitledCollateralBase

This is an abstract base type which contains fields that are common to all titled goods collateral items, where ownership is evidenced by a title. Examples are an automobile, airplane, boat, etc.

### PossessoryCollateralBase

This is an abstract base type for collateral for possessory collateral types. Examples are: Deposit Accounts/CD; Securities; Notes; Receipts/Bills; Letters of Credit; Other.

# Collateral Services

## Service Definitions

The Collateral data model and services support Create, Read, Delete, and Update operations against a Collateral item.

### Service Message: ReadCollateral

When provided a collateralFilter, the ReadCollateral service message will return a list of collateral items that match the filter. For example, the collateralFilter when containing an accountId would return all collateral associated to that account.

|  |  |
| --- | --- |
| INPUTS | cufx:collateralMessage (which includes)   * [cufx:MessageContext](file:///\\files2\users\CMarjaniemi\Projects\CUFX\MessageContext.html) * [cufx:Collateral](file:///\\files2\users\CMarjaniemi\Projects\CUFX\Generated%20HTML%20Docs\Loan.html)Filter * [cufx:Collateral](file:///\\files2\users\CMarjaniemi\Projects\CUFX\Generated%20HTML%20Docs\Loan.html)List |
| Outputs | cufx:collateralMessage (which includes)   * [cufx:MessageContext](file:///\\files2\users\CMarjaniemi\Projects\CUFX\MessageContext.html) * [cufx:Collateral](file:///\\files2\users\CMarjaniemi\Projects\CUFX\Generated%20HTML%20Docs\Loan.html)List |
| Return Values | cufx:collateralMessage (which includes)   * cufx:MessageContext   + statusList |
| Side Effects | No data is being manipulated, only a lookup is being performed. |
| Dependencies | None. |
| CUFX REST LINK | https://api.dataprovider.com/collateralmessage |

REST-JSON Example

The following example illustrates how to return all collateral items on account 12345.

**REQUEST:**

Headers:

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

Accept: application/json

Accept-Charset: utf-8

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

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

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

X-API-Version: >=5.0.0

**POST h**ttps://api.dataprovider.com/collateralmessage

{

collateralMessage:{

"messageContext" : <SeeMessageContext.xsd>

,

"collateralFilter" : {

"accountIdList" : [ "12345" ],

}

}

}

RESPONSE:

**Headers:**

Status Code: 200 Ok

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

Content-Language: en-us

Payload:

{"collateralMessage": {

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

"messageContext": {

"cufxVersion": "5.0.0",

"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": {

"currentPage": "1",

"totalPages": "1",

"pageSize": "1",

"positionIndex": "positionIndex1",

"returnPage": "1",

"returnPositionIndex": "returnPositionIndex1",

"status": {

"statusType": "Success"

}

}

},

"collateralList": {

"currentPage": "1",

"totalPages": "1",

"pageSize": "1",

"positionIndex": "positionIndex1",

"returnPage": "1",

"returnPositionIndex": "returnPositionIndex1",

"collateral": {

"motorVehicleCollateral": {

"collateralId": "collateralId1",

"collateralCode": "collateralCode1",

"collateralPledgedDate": "1900-01-01",

"description": "description1",

"securedCode": "securedCode1",

"totalSecuredAmount": {

"value": "1",

"currencyCode": "USD",

"exchangeRate": "1"

},

"valuationOfCollateral": {

"value": "1",

"currencyCode": "USD",

"exchangeRate": "1"

},

"ownerOfCollateral": "ownerOfCollateral1",

"loanToValuePercentage": "1",

"collateralValueSourceDate": "1900-01-01",

"collateralValueSource": "collateralValueSource1",

"year": "1",

"manufacturer": "manufacturer1",

"make": "make1",

"model": "model1",

"color": "color1",

"purchasePrice": {

"value": "1",

"currencyCode": "USD",

"exchangeRate": "1"

},

"isVehicleUsed": "true",

"uccCode": "uccCode1",

"vehicleIdentificationNumber": "vehicleIdentificationNumber1",

"mileage": "1",

"licensePlateNumber": "licensePlateNumber1",

"licensePlateState": "licensePlateState1",

"licensePlateExpiration": "1900-01-01",

"titleNumber": "titleNumber1",

"titleState": "titleState1"

}

}

}

}

}

### Service Message: CreateCollateral

The CreateCollateral service message will create the collateral record on the host system. To associate the collateral item to a loan, you would need to create or update the loan object and have it contain the collateralId returned from this service call.

|  |  |
| --- | --- |
| INPUTS | cufx:collateralMessage (which includes)   * [cufx:MessageContext](file:///\\files2\users\CMarjaniemi\Projects\CUFX\MessageContext.html) * [cufx:Collateral](file:///\\files2\users\CMarjaniemi\Projects\CUFX\Generated%20HTML%20Docs\Loan.html)List |
| Outputs | cufx:collateralMessage (which includes)   * [cufx:MessageContext](file:///\\files2\users\CMarjaniemi\Projects\CUFX\MessageContext.html) * [cufx:Collateral](file:///\\files2\users\CMarjaniemi\Projects\CUFX\Generated%20HTML%20Docs\Loan.html)List |
| Return Values | cufx:collateralMessage (which includes)   * cufx:MessageContext   + statusList |
| Side Effects | Creates a new collateral record on the core system if successful. |
| Dependencies | The account that this collateral will belong to must already exist. |
| CUFX REST LINK | https://api.dataprovider.com/collateralmessage |

REST-JSON Example

**REQUEST:**

Headers:

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

Accept: application/json

Accept-Charset: utf-8

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

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

X-API-Version: >=5.0.0

**POST h**ttps://api.dataprovider.com/collateralmessage

{

collateralMessage:{

"messageContext" : <See MessageContext.xsd>,

"collateralList" : { <See Collateral.xsd> }

}

}

RESPONSE:

**Headers:**

Status Code: 200 Ok

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

Content-Language: en-us

Payload:

{"collateralMessage": {

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

"messageContext": {

"cufxVersion": "5.0.0",

"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": {

"currentPage": "1",

"totalPages": "1",

"pageSize": "1",

"positionIndex": "positionIndex1",

"returnPage": "1",

"returnPositionIndex": "returnPositionIndex1",

"status": {

"statusType": "Success"

}

}

},

"collateralList": {

"collateral": {

"motorVehicleCollateral": {

"collateralId": "collateralId1",

"collateralCode": "collateralCode1",

}

}

}

}

}

### Service Message: UpdateCollateral

The UpdateCollateral service message is used to update the collateral record on the host system. A valid collateralFilter must be passed in that indicate the collateral or collaterals to be updated.

|  |  |
| --- | --- |
| INPUTS | cufx:collateralMessage (which includes)   * [cufx:MessageContext](file:///\\files2\users\CMarjaniemi\Projects\CUFX\Generated%20HTML%20Docs\MessageContext.html) * [cufx:Collateral](file:///\\files2\users\CMarjaniemi\Projects\CUFX\Generated%20HTML%20Docs\Loan.html)Filter * [cufx:Collateral](file:///\\files2\users\CMarjaniemi\Projects\CUFX\Generated%20HTML%20Docs\Loan.html) |
| Outputs | cufx:collateralMessage (which includes)   * [cufx:MessageContext](file:///\\files2\users\CMarjaniemi\Projects\CUFX\Generated%20HTML%20Docs\MessageContext.html) * [cufx:Collateral](file:///\\files2\users\CMarjaniemi\Projects\CUFX\Generated%20HTML%20Docs\Loan.html)List |
| Return Values | cufx:collateralMessage (which includes)   * cufx:MessageContext   + statusList |
| Side Effects | Updates the existing collateral record on the core system if successful. |
| Dependencies | The account that this collateral item will belong to must already exist. |
| CUFX REST LINK | https://api.dataprovider.com/collateralmessage |

REST-JSON Example

**REQUEST:**

Headers:

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

Accept: application/json

Accept-Charset: utf-8

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

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

X-API-Version: >=5.0.0

**PUT h**ttps://api.dataprovider.com/collateralmessage

{

collateralMessage:{

"messageContext" : <See MessageContext.xsd>,

"collateralFilter" : { "collateralIdList" : [ "1234" ] }

"collateralList" : { <See Collateral.xsd> }

}

}

**RESPONSE:**

**Headers:**

Status Code: 200 Ok

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

Content-Language: en-us

Payload:

{"collateralMessage": {

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

"messageContext": {

"cufxVersion": "5.0.0",

"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": {

"currentPage": "1",

"totalPages": "1",

"pageSize": "1",

"positionIndex": "positionIndex1",

"returnPage": "1",

"returnPositionIndex": "returnPositionIndex1",

"status": {

"statusType": "Success"

}

}

},

"collateralList": {

"collateral": {

"motorVehicleCollateral": {

"collateralId": "collateralId1",

"collateralCode": "collateralCode1",

}

}

}

}

}

### Service Message: DeleteCollateral

When provided a collateralId, the DeleteCollateral service message will delete the collateral record with the given collateralId.

|  |  |
| --- | --- |
| INPUTS | Cufx:collateralMessage (which includes)   * [cufx:MessageContext](file:///\\files2\users\CMarjaniemi\Projects\CUFX\MessageContext.html) * [cufx:Collateral](file:///\\files2\users\CMarjaniemi\Projects\CUFX\Generated%20HTML%20Docs\Loan.html)List |
| Outputs | Cufx:collateralMessage (which includes)   * [cufx:MessageContext](file:///\\files2\users\CMarjaniemi\Projects\CUFX\MessageContext.html) |
| Return Values | Cufx:collateralMessage (which includes)   * [cufx:MessageContext](file:///\\files2\users\CMarjaniemi\Projects\CUFX\MessageContext.html)   + statusList |
| Side Effects | No data is being manipulated, only a lookup is being performed. |
| Dependencies | None. |
| CUFX REST LINK | https://api.dataprovider.com/collateralmessage |

REST-JSON Example

The following example illustrates how to delete all collateral items on account 12345.

**REQUEST:**

Headers:

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

Accept: application/json

Accept-Charset: utf-8

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

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

**X-HTTP-Method-Override: DELETE**

X-API-Version: >=5.0.0

**POST h**ttps://api.dataprovider.com/collateralmessage

{

collateralMessage:{

"messageContext" : <SeeMessageContext.xsd>,

"collateralFilter" : {

{ "collateralIdList" : [ "12345" ] }

}

}

}

{"collateralMessage": {

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

"messageContext": {

"cufxVersion": "5.0.0",

"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": {

"currentPage": "1",

"totalPages": "1",

"pageSize": "1",

"positionIndex": "positionIndex1",

"returnPage": "1",

"returnPositionIndex": "returnPositionIndex1",

"status": {

"statusType": "Success"

}

}

}

}

}

# Bibliography

W3C. (n.d.). *Key words for use in RFCs to Indicate Requirement Levels [RFC2119].* Retrieved Sept. 8th, 2011, from W3C.