

ERROR MAPPING 3.3

# Document Status

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

This version: **Assembla**.com. Files Tag = CUFX\_3.3\_RFC\_Active

Previous Version: **Assembla**.com. Files Tag = CUFX\_3.2\_RFC\_Archive

# Change Log

|  |  |  |
| --- | --- | --- |
| Version |  | Changes |
| 0.1 |  | * Initial Creation |
| 0.2 |  | * Added Validation Error Codes |
| 0.3 |  | * Added/modified generic error codes with parameterized detail text to support specific data integrity/validation issues and actionable error messages. |
| 0.4 |  | * Convert to XSD format and reference in this document. Update example errors. |
| 0.0.5 |  | * Change plural items to List form |
| 0.0.6 |  | * Transmission of errors |
| 0.0.7 |  | * Updated Overview of Specification |
| 0.0.8 |  | * Added Error Types |
| 0.0.9 |  | * Make language support up to end points * Add substitution ability for portions of the error message |
| 3.0 | **10/29/2013** | * Versioning and format change with release CUFX 3.0 |
| 3.0 | **12/18/2013** | * Minor formatting and text updates |
| 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 |

# Overview of Specification

The CUFX Error Mapping specification describes the standardize error codes and format for the CUFX specifications. It contains all the specific error messages and their customizable constructs to provide additional detail for the end user and client application.

# Any know Errors in the document

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

Table of Contents

[Document Status 1](#_Toc474999144)

[Change Log 1](#_Toc474999145)

[Overview of Specification 2](#_Toc474999146)

[Any know Errors in the document 2](#_Toc474999147)

[Document Conventions 2](#_Toc474999148)

[Concepts 3](#_Toc474999149)

[Data Elements 3](#_Toc474999150)

[Error Response Example - REST-JSON 3](#_Toc474999151)

[Error codes 4](#_Toc474999152)

[Error types 4](#_Toc474999153)

[Bibliography 4](#_Toc474999154)

# 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

# Concepts

The CUFX Data Provider sends error messages when the client fails to provide an expected request or the Data Provider has an internal failure. Each error response consists of a non-200 HTTP response roughly corresponding to the accepted list of HTTP status codes and a payload containing a detailed error code and message. Fields not returned should be assumed to be null. The header record of the request contains the language that the error message should be displayed in. It is up to the endpoints to determine if multi-language will be supported.

The CUFX service can return default errors like HTTP error code 500 or 404. When a CUFX Data Provider returns those error codes, it’s the responsibility of the service consumer to look deep into the error object for more details.

The CUFX service needs to configure to return the error object when a HTTP error code is returned.

Example:

If you build using .net platform this behavior can be overridden via web.config file as depicted below. This will not trash the custom error object that’s part of the response when a HTTP error code is returned by the CUFX service.

<configuration>

<system.webServer>

       <!-- ensures that on remote server we get our custom json responses for HTTP code 500 and NOT generic IIS response-->

              <httpErrors existingResponse="PassThrough"></httpErrors>

...

# Data Elements

The data elements returned by an error object is contained in Error.xsd.

### Error Response Example - REST-JSON

{

“errorList”: [

{

“code”: 499,

“type”: 400,

“subCode”: 548541521,

“message”: “General Error. Review subCode for more Information”

},

{

“code”: 400,

“type”: 300,

“subCode”: 0,

“message”: “Bad Request. Data in the request was invalid.”

},

{

“code”: 412,

“type”: 300,

“subCode”: 0,

“message”: “Previous request required. A contact creation request must be made before this one, and was not”

},

{   “code”: 413,

   “subCode”: 0,

   "type":"ValidationException",

    “message”: “Value out of acceptable range. The value ‘%1’ is not supported

for the field ‘%2’ by this service. Valid values must be between

‘%3’ and ‘%4’.

    “substitutionList:[

            {“id”:”%1”,

             “value”:”999”,

            },

            {“id”:”%2”,

             “value”:”CreditScore”,

            },

            { “id”:”%3”,

             “value”:”250”

            },

           { "id":"%4",

             "value":"899"

           }

    ]

}

]

}

### Error codes

See the Error.xsd for all valid error codes.

### Error types

See the Error.xsd for all valid error types.

# Bibliography

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