RESO Web API Client Testing Rules v1.0.3

RESO Web API Client Testing Rules v1.0.3

Date of Last Update: December 12, 2018

1.0 Introduction

2.0 RESO Web API Client Compliance Testing Rules

3.0 RESO Web API Client Certification Rules

4.0 RESO Web API Client Endorsement Rules

5.0 RESO Web API Report Card and Specifications

1.0 Introduction

This document contains the RESO Web API Client Testing Rules. These must be satisfied for clients to become certified against the RESO Web API v1.0.3 standard. This ensures clients can communicate with RESO Certified Web API Servers.

This document should be read by any organization who wants:

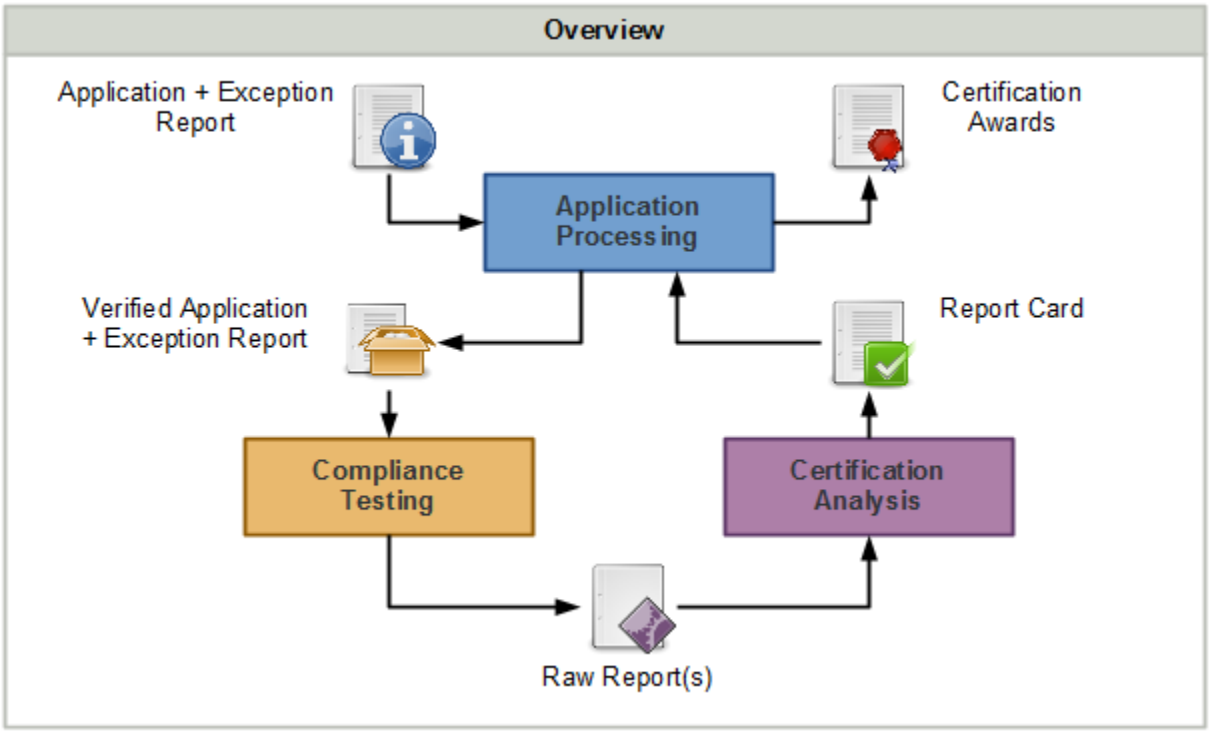
- To create a RESO Web API compliant clients.
- To gain an understanding of the certification process.

The multi-step certification process begins with an application submitted through http://reso.org/certification. Client certification are separate from the RESO Web API Server Certifications. Organizations that have both a client and a server implementation must complete two certification processes, one for each type.

1.1 Glossary

A glossary for common terms for RESO Web Client API processes is here: RESO Web API Certification Glossary.

1.2 RESO Certification Flow (Summary)



RESO Group	Action	Output
Application Processing (Pre-Certification)	Accept and Verify Applicant's ***'Certification Application'*** via reso.org/certification	Prepare for Compliance Testing. Pass application and any information provided by applicant to Compliance Department.
Compliance Testing	Test applicant's implementation against well-defined Compliance Rules as set forth by the Transport and Compliance Workgroups.	Testing results formatted in ***'Raw Report'*** package. Pass ***'Raw Report'*** to Certification Department.
Certification Analysis	Analyze ***'Raw Report'*** to determine if applicant qualifies for a certificate. Create a ***' Report Card'*** with findings.	Pass analysis results and ***'Report Card'*** back to Application Processing.

Application Processing (Post-Certification)	Act on Certification Department recommendation ("Certify" or "Request Changes")	Notify applicant of Certificate Pass/Fail. Send notification and ***'Report Card'*** back to Applicant.

2.0 RESO Web API Client Compliance Testing Rules

This section contains multiple rule sets RESO Staff will use during compliance testing. The specific sets of rules that must be passed to receive a Certificate are discussed in Section 3.

> **NOTE 1:** This version of the RESO Web API Testing Rules will only contain those applying to RESO Web API Clients. The RESO Web API Server Testing Rules 1.0.3 are recorded in a separate document and may be merged for future RESO Web API standard versions.

The RESO Web API Client Testing Rules are complied as one of two types: **Individual Requests** or **Scenario Executions**.

The **Individual Request** is a single technical activity. These reflect a fundamental API query, such as OpenID authorization request, OpenID access token request, API GET query, $metadata query and similar. The **Individual Request** testing rules will evaluate the RESO client's conformance and functionality, where each request is executed individually.

The **Scenario Executions** will test the client against practical, real-world usage examples by coupling several **Individual Requests** that would represent a finite routine. For example:

1. Send OpenID authorization request with login credentials;
2. Receive authentication token, if authorization is successful;
3. Use the authorization token to submit API queries;
4. Query /Properties?$top=10.
5. Receive response with successful HTTP Headers and results content.

After the whole scenario steps are accurately executed and completed according the rule set, the client gets a pass for the specific scenario.

The **Individual Request** and **Scenario Executions** are grouped by similar functionality in the sections below. The description of what groups are required for certification and endorsements type is provided is Section 3.

2.1 RESO Web API Security Support (Security)

2.2 Client Communication Requests

2.3 Client Requests and Scenarios

2.1 RESO Web API Security Support (Security)

The RESO Web API Security v1.0.3 Standard Section 1 defines multiple methods from the OAuth 2.0 and OpenID Connect Standards.

> *A compliant RESO Web API Server v1.0.3* ***MUST*** *support* ***token based authentication*** *with an HTTP `Authorization` header of "Bearer <`token`>" where the format of <`token`> is defined by the compliant RESO Web API Server v1.0.3.*
>
> *A compliant RESO Web API Server v1.0.3* ***MUST*** *support Section 2.1 of RFC 6750 OAuth2 Bearer Token Usage as the authentication method for server-to-server based communication.*
>
> *A compliant RESO Web API Server v1.0.3* ***MUST*** *support Authorization Code Flow of the OpenID Connect Protocol Suite as the authentication method for user based communication.*
>
> ---
>
> *A compliant RESO Web API Server v1.0.3* ***MAY*** *use any of the additional OpenID Connect Protocol Suite of standard specifications.*
>
> *For data warehousing and replicating data between servers, a compliant RESO Web API Server v1.0.3* ***MAY*** *use the OAuth2 Client Credentials Grant specified in Section 4.4 of RFC 6749 The OAuth 2.0 Authorization Framework.*

RESO Web API Client Testing is limited to the "Authorization Code Flow of the OpenID Connect Protocol Suite" and "RFC 6750 OAuth2 Bearer Token Usage" security standards. The "RFC 6749 The OAuth 2.0 Authorization Framework" **WILL NOT** be tested nor supported in certification testing.

The sections below contain the rules that RESO will use in the security testing. The specific set of rules that need must be passed for a "Certification" are discussed in Section 3.

2.1.1 OpenID Connect Standard

2.1.2 OAuth 2.0 Standard Bearer Token

2.1.1 OpenID Connect Standard

The RESO Web API Security Document outlines the security protocols required for RESO Web API Servers. These security protocols focus on OpenID Connect technologies.

Applications that apply for RESO Web API Client Certificates that require OpenID Connect Standard **MUST** satisfy **ALL** of the following requirements.

RESO Rule ID	Rule Summary	Rule Details	Technical Details
REQ-WC103-OPENID-01	Send OpenID authorize request to retrieve auth code - part 1/2 (HTTP GET)	Checks whether the client sends a properly formed authorization request. Depending on the authorization type a certain set of headers and / or body content must be included in the request.	Request Type: HTTP GET Header: Depending on the authorization method, the client must provide client_id, scope, redirect_uri and response_type = "code". Scope: Authorization request to obtain auth code
REQ-WC103-OPENID-02	Send OpenID request for access / bearer token	Checks whether the client, having the auth code obtained, constructs a proper access / bearer token request and succesfully receives it for further use in the requests.	Request Type: HTTP POST Header: Must contain a "Authorization" header containing of "Basic: " and base64 encoded string of "client_id:client_secret"; post parameters - grant_type = "authorization_code", client_id, redirect_uri and code = auth code, received during authroize request set Scope: Retrieval of the access / bearer token
REQ-WC103-OPENID-03	Send OpenID refresh token request	Submits a request to retrieve refresh token and receives it.	

> **NOTE 1:** Additional testing procedures **MAY** be derived from issues that come when using the OpenID Foundation's Relying Party certification track (https://openid.net/certification/rp_testing/) and software (https://rp.certification.openid.net:8080/).

2.1.2 OAuth 2.0 Standard Bearer Token

The RESO Web API Security Document outlines the security protocols required for RESO Web API Servers. These security protocols focus on OAuth 2.0 technologies.

Applications that apply for RESO Web API Client Certificates that require OAuth 2.0 Bearer Token Usage Standard **MUST** satisfy **ALL** of the following requirements.

RESO Rule ID	Rule Details
REQ-WS103-OABT-01	A compliant RESO Web API Client **MUST** support token based authentication with an HTTP Authorization header of "Bearer <token>" where the format of <token> is defined by the compliant RESO Web API Server v1.0.3.
REQ-WS103-OABT-02	A compliant RESO Web API Client **MUST** support Section 2.1 of RFC 6750 OAuth2 Bearer Token Usage as the authentication method for server-to-server based communication.

> **NOTE 1:** Additional testing procedures **MAY** be derived from issues that come when using the RESO Web API Client Testing Tool.

2.2 Client Communication Requests

The following testing rules focus on a client's ability to provide proper HTTP Header information to communicate with a RESO Web API Server or other miscellaneous communication requirements.

RESO Rule ID	Rule Summary	Rule Details	Technical Details
REQ-WC103-COMM-01	Requests contain client version header	Every clients sent HTTP reques to the API service must contain a "Version" header with the value of the client's version.	Request Type: any HTTP request Header: "Version" is set, provides client name and verison Scope: Provided in all client's HTTP requests to API service

REQ-WC103-COMM-02	Requests after authentication contain proper access token header	Every client sent HTTP request after successful authentication must contain a proper access / bearer token entry, set in the request header.	Request Type: any HTTP request requiring access token Header: "Authorization" header is set to "Bearer <access_token>" value Scope: Provided in all client's HTTP requests to API service, where access token is required
REQ-WC103-COMM-03	Client populates proper "Accept" header	Every client sent request to API service endpoints, which returns data in selected format (e.g. JSON, XML and others). The Accept header should contain the expected response MIME type, which is supported by the API service.	Request Type: any HTTP request returning data Header: "Accept" header is set to expected response MIME type, e.g. "application/json", supported by API service and endpoints Scope: Any request to API service's endpoints which return data in response
REQ-WC103-COMM-04	Client acts as a valid HTTP 1.1 request client	The RESO Web API v1.0.3 client must create and send valid HTTP requests that adhere to sections 4 and 5 of https://www.ietf.org/rfc/rfc7231.txt ***NOTE:*** *Additional testing procedures* ***MAY*** *be derived from RFC7230, RFC7231, RFC7232, RFC7233, RFC7234, and RFC7235 to support this rule.*	Request Type: any HTTP request Scope: All requests

2.3 Client Requests and Scenarios

Please refer to section 2.0 RESO Web API Client Compliance Testing Rules for a description of the difference between the **Individual Requests** and **Scenario Executions** listed below.

2.3.1 Client Requests

The following testing rules focus on a client's ability to request information from a RESO Web API Server other miscellaneous client request requirements.

RESO Rule ID	Rule Summary	Rule Details	Technical Details
REQ-WC103-CREQ-01	Send raw HTTP GET request to endpoint	Submits a raw HTTP GET request to any API endpoint, requesting data records. API must return a 200 OK response with record body.	Request Type: HTTP GET Header: Proper "Accept" value (e.g. */*) and "Authorization" = "Bearer: <access_token>" Scope: Request to any API service's endpoints to retrieve data, e.g. $metadata
REQ-WC103-CREQ-02	Send request to API endpoint with special chars	Submits a request with special chars, which need to be properly escaped (url-encoded) to API endpoints to retrieve record data. API service delivers back a 200 OK response with returned record data.	Request Type: HTTP GET / POST Header: any necessary to reach the API endpoint and finalize a request, parameters - HTTP GET URL parameters or POST parameters must be properly escaped Scope: Any single request to an API endpoint requesting for record return with parameters, containing special chars or any character that needs to be url-encoded / escaped.
REQ-WC103-CREQ-03	Send raw GET request to $metadata endpoint and store the output for later use	Using the access token obtained the client requests $metadata in XML format, parses its information for later use.	Technical requirements as per Single Action in Core Level, related to raw GET request.

2.3.2 Client Scenarios

The following testing scenarios focus on the "Client Requests" Individual Requests described previously.

RESO Rule ID	Rule Summary	Rule Details	Technical Details

REQ-WC103-CSCE-01	Send raw HTTP GET request to endpoint	Submits a raw HTTP GET request to any API endpoint, requesting data records. API must return a 200 OK response with record body.	Request Type: HTTP GET Header: Proper "Accept" value (e.g. */*) and "Authorization" = "Bearer: <access_token>" Scope: Request to any API service's endpoints to retrieve data, e.g. $metadata
REQ-WC103-CSCE-02	Send request to API endpoint with special chars	Submits a request with special chars, which need to be properly escaped (url-encoded) to API endpoints to retrieve record data. API service delivers back a 200 OK response with returned record data.	Request Type: HTTP GET / POST Header: any necessary to reach the API endpoint and finalize a request, parameters - HTTP GET URL parameters or POST parameters must be properly escaped Scope: Any single request to an API endpoint requesting for record return with parameters, containing special chars or any character that needs to be url-encoded / escaped.
REQ-WC103-CSCE-03	Send raw GET request to $metadata endpoint and store the output for later use	Using the access token obtained the client requests $metadata in XML format, parses its information for later use.	Technical requirements as per Single Action in Core Level, related to raw GET request.
REQ-WC103-CSCE-04	Send raw GET request to Properties endpoint	The client demonstrates possibility to query Properties endpoint succesfully.	Technical requirements as per Single Action in Core Level, related to raw GET request.
REQ-WC103-CSCE-05	Send raw GET response to get Properties top 10 output	The client demonstrates the possibility to add parameters in the Properties endpoint query. Requests $top=10 results.	Technical requirements as per Single Action in Core Level, related to raw GET request.
REQ-WC103-CSCE-06	Get Properties single record response in XML or JSON	The client demonstrates the possibility to request either of the different output formats (XML or JSON) for the same API endpoint query.	Technical requirements as per Single Action in Core Level, related to raw GET request. Header "Accept" must be set appropriately.

2.3.3 Client Query Requests

The following testing rules focus on a client's ability to structure valid queries to extract data from a RESO Web API Server other miscellaneous client query request requirements. These query request requirements are limited to the minimal functionality required by RESO Web API Servers at the time these rules were published.

RESO Rule ID	Rule Summary	Rule Details	Technical Details
REQ-WC103-QREQ-01	Using $metadata information send a raw GET request on an endpoint with a specific set of columns in $select	The client, using the $metadata output, picks several columns for a certain endpoint and queries it using the $select parameter and the specific column list selected. The API endpoint returns a 200 OK and response body.	Technical requirements as per Single Action in Core Level, related to raw GET request.
REQ-WC103-QREQ-02	API request: Search by UniqueID	The client demonstrates the possibility to query a given API endpoint using the UniqueID defined in the $metadata.	Technical requirements as per Single Action in Core Level, related to raw GET request.
REQ-WC103-QREQ-03	API request with $orderby usage	The client demonstrates the possibility to query a given API endpoint and ordering the result set in several ways.	Technical requirements as per Single Action in Core Level, related to raw GET request. The request should include multiple order columns and multiple types - ASC and DESC.
REQ-WC103-QREQ-04	API request with $top usage	The client demonstrates the possibility to query a given API endpoint with any request that would use the $top parameter.	Technical requirements as per Single Action in Core Level, related to raw GET request.
REQ-WC103-QREQ-05	API request with $skip usage	The client demonstrates the possibility to query a given API endpoint with any request that would use the $skip parameter.	Technical requirements as per Single Action in Core Level, related to raw GET request.

REQ-WC103-QREQ-06	API request with $filter and negate with not operator – general requirement.	The client demonstrates the possibility to query a given API endpoint with the negation operator 'not'. Uses any column and operator and / or function from previously tested ones.	Technical requirements as per Single Action in Core Level, related to raw GET request. The request should contain multiple negations on different endpoint columns.
REQ-WC103-QREQ-06.EQ	API request with $filter and negate with not operator – 'eq' (equal) operator	See **REQ-WC103-QREQ-06** for Rule Details.	See **REQ-WC103-QREQ-06** for Technical Details.
REQ-WC103-QREQ-06.NE	API request with $filter and negate with not operator – 'ne' (not equal)	See **REQ-WC103-QREQ-06** for Rule Details.	See **REQ-WC103-QREQ-06** for Technical Details.
REQ-WC103-QREQ-06.GT	API request with $filter and negate with not operator – 'gt' (greater than)	See **REQ-WC103-QREQ-06** for Rule Details.	See **REQ-WC103-QREQ-06** for Technical Details.
REQ-WC103-QREQ-06.GE	API request with $filter and negate with not operator – 'ge' (greater or equal)	See **REQ-WC103-QREQ-06** for Rule Details.	See **REQ-WC103-QREQ-06** for Technical Details.
REQ-WC103-QREQ-06.LT	API request with $filter and negate with not operator – 'lt' (less than)	See **REQ-WC103-QREQ-06** for Rule Details.	See **REQ-WC103-QREQ-06** for Technical Details.
REQ-WC103-QREQ-06.LE	API request with $filter and negate with not operator – 'le' (less or equal)	See **REQ-WC103-QREQ-06** for Rule Details.	See **REQ-WC103-QREQ-06** for Technical Details.
REQ-WC103-QREQ-06.HAS	API request with $filter and negate with not operator – 'has'	See **REQ-WC103-QREQ-06** for Rule Details.	See **REQ-WC103-QREQ-06** for Technical Details.
REQ-WC103-QREQ-06.AND	API request with $filter and negate with not operator – 'and'	See **REQ-WC103-QREQ-06** for Rule Details.	See **REQ-WC103-QREQ-06** for Technical Details.
REQ-WC103-QREQ-06.OR	API request with $filter and negate with not operator – 'or'	See **REQ-WC103-QREQ-06** for Rule Details.	See **REQ-WC103-QREQ-06** for Technical Details.
REQ-WC103-QREQ-06.NOT	API request with $filter – 'not'	See **REQ-WC103-QREQ-06** for Rule Details.	See **REQ-WC103-QREQ-06** for Technical Details.
REQ-WC103-QREQ-06.DATE	API request with $filter and negate with not operator – 'Date: Date'	See **REQ-WC103-QREQ-06** for Rule Details.	See **REQ-WC103-QREQ-06** for Technical Details.
REQ-WC103-QREQ-06.NOW	API request with $filter and negate with not operator – 'Date: Now'	See **REQ-WC103-QREQ-06** for Rule Details.	See **REQ-WC103-QREQ-06** for Technical Details.

2.3.4 Client Query Scenarios

The following testing scenarios focus on the "Client Query Requests" Individual Requests described previously.

RESO Rule ID	Rule Summary	Rule Details	Technical Details
REQ-WC103-QSCE-01	Advanced Scenario with $metadata and $select, where multiple endpoints with a different combination of columns are queried.	The client retrieves the $metadata output, extracts the endpoint information and queries multiple endpoints with different column sets of the endpoints using $select parameter.	Technical requirements as per Single Action in Core Level, related to raw GET request. The parameters and values must be properly escaped (url-encoded).
REQ-WC103-QSCE-02	Scenario with $filter on numeric columns with multiple operators gt, lt, eq, le, ge	Following the $metadata and $select activity, the client must construct a $filter parameter, using the selected columns and according their types construct and use the comparison (gt, lt, eq, le, ge) operators and values.	Technical requirements as per Single Action in Core Level, related to raw GET request. The parameters and values must be properly escaped (url-encoded). Multiple columns must be queried and different operators and values used. The API endpoint must return a 200 OK response and potentially a body with records.

REQ-WC103-QSCE-03	Scenario with $orderby and confirmation of the result set parsing	Using the filtered result set from the previous activity the client must request several queries with different (multiple) column and order type (asc. desc) combinations. After the results are returned using the queries, the client should parse the first record in the result set and construct a query, which would in particular return only that single record, so confirming the advanced query and response parsing capabilities (except the use of $top=1).	Technical requirements as per Single Action in Core Level, related to raw GET request. The parameters and values must be properly escaped (url-encoded). Multiple columns must be queried and different operators and values used. The API endpoint must return a 200 OK response and potentially a body with records.
REQ-WC103-QSCE-04	Scenario where $top is used and confirmation of the result set parsing	Similarly as the $orderby activity, the client should construct a record select query on a given API endpoint and providing the $top parameter (> 1). Afterwards pick the last record in the result set and construct a query on the endpoint which would return only that specific record.	Technical requirements as per Single Action in Core Level, related to raw GET request.
REQ-WC103-QSCE-05	Scenario where $top and $skip is used and pagination	The client demonstrates the possibility to do record result "pagination" by using the $top and $skip parameters. The client **MAY** use the $inlinecount query to count the total number of results. The acceptance criteria is when the client traverses all the result "pages" according a fixed $top parameter.	Technical requirements as per Single Action in Core Level, related to raw GET request. The $top parameter must be a fixed number throughout all the scenario activity.

3.0 RESO Web API Client Certification Rules

This page lists the rules that RESO will use in awarding RESO Web API Client Certificates. The specific set of rules that must be passed for "Compliance" are discussed in Section 2.

Certification is awarded when all requirements detailed in this testing rules document have been satisfied.

> **NOTE 0-1:** There is only one RESO Web API Client Certificate. **ALL** clients applying for certification is expected to satisfy **ALL** requirements. Separate "endorsements" are awarded to clients that perform additional functionality.

REQ-WC103-CERT-01: Satisfy **ALL** requirements from Client Communication Requests Section.

REQ-WC103-CERT-02: Satisfy **ALL** requirements from Client Requests and Scenarios Section.

REQ-WC103-CERT-03: Satisfy **ALL** requirements from **ONE** of the RESO Web API Client Security Endorsements, based on client purposes. Client **MAY** earn more than one.

> **NOTE 3-1:** The Security Endorsements available for RESO Web API Client Certification include the "OAuth 2.0 Bearer Token" and "OpenID Connect" endorsement. Compliance Rules are available in Section 2.1 and Endorsement Rules are available in Section 4.0.

REQ-WC103-CERT-04: Clients used for interactive, on-demand communications **MUST** receive the "OpenID Connect" endorsement.

> **NOTE 4-1:** Interactive, on-demand communication refers to any Client-to-Server API submissions initiated by a Client's End User.

REQ-WC103-CERT-05: Clients used for server-to-server communications **MUST** receive the "OAuth 2.0 Bearer Token" endorsement.

> **NOTE 4-1:** Clients used by End Users (Agents, Brokers, Staff, Public, etc.) **WILL NOT** be certified when the "OAuth 2.0 Bearer Token" endorsement is used to satisfy **REQ-WC103-CERT-03**. (End User Clients **MUST** receive the "OpenID Connect" endorsement.

4.0 RESO Web API Client Endorsement Rules

RESO Web API Client Endorsements are awarded to clients that perform additional functionality. These **MAY** be required for certification, as defined in the certification testing rules in Section 3.0.

4.1 OpenID Connect Endorsement

The RESO Web API Client "OpenID Connect" endorsement is used for on-demand communications where credentials are entered when logging into systems.

REQ-WC103-OIDCE-01: Satisfy **ALL** requirements from RESO Web API Security Support (Security) "OpenID Connect Standard" Section.

4.2 OAuth 2.0 Bearer Token Endorsement

The RESO Web API Client "OAuth 2.0 Bearer Token" endorsement is used for server-to-server communications.

REQ-WC103-OABTE-01: Satisfy **ALL** requirements from RESO Web API Security Support (Security) "OAuth 2.0 Standard Bearer Token" Section.

5.0 RESO Web API Report Card and Specifications

The RESO Web API Report Card is used to report to the applicant the certification findings. This will include a list of the testing results from testing.

The exact format will be determined by the RESO Compliance Staff with input from the RESO Web API Workgroup.