

CARRIER SERVICEABILITY API INTERFACE

Table of contents

Introduction	3
Online help, testing and documentation	4
Security	8
Single address API	9
Sample response single API	10
Bulk address API	11
Request for access token (secure calls)	16
Creating a SoapUI test client for waCarrierEli and Security Eli	18

Introduction

The Spectrum Enterprise Carrier serviceability web API provides two methods for obtaining serviceability information: A single address request and a batch address request.

Production:

<https://eli-ws-carrier.charter.com/CarrierSoap.svc>

QA:

<https://eli-ws-carrier-uat.charter.com/CarrierSoap.svc>

Online help, testing and documentation

Clients can access Microsoft's web API help pages, pull in a WADL file, and do more with a Swagger UI endpoint. Via Swagger UI, clients can actively test out the API and see both the request and response in real time.

Production:

<https://eli-ws-carrier.charter.com>

<https://eli-ws-carrier.charter.com/help>

<https://eli-ws-carrier.charter.com/help/wadl>

<https://eli-ws-carrier.charter.com/swagger/ui/index>

QA:

<https://eli-ws-carrier-uat.charter.com>

<https://eli-ws-carrier-uat.charter.com/help>

<https://eli-ws-carrier-uat.charter.com/help/wadl>

<https://eli-ws-carrier-uat.charter.com/swagger/ui/index>

To test the bulk endpoint via Swagger UI, navigate to:

<https://eli-ws-carrier-uat.charter.com/swagger/ui/index#/Carrier>

Click on "expand operations" and there will be a screen similar to the screenshot below.

The screenshot displays the Swagger UI for the **waCarrierELI** API. At the top, there is a green header with the Swagger logo, the API URL `https://eli-ws-carrier-uat.charter.com:443/swagger/docs/v1`, an `api_key` input field, and an **Explore** button. Below the header, the API title **waCarrierELI** is shown. Underneath, the **Carrier** group is expanded, showing a list of operations. Each operation is represented by a colored bar (blue for GET, green for POST) with the HTTP method, the endpoint path, and the operation name. The operations listed are:

Method	Endpoint	Operation Name
GET	<code>/api/CarrierServiceability/GetCarrierServiceability</code>	Carrier Serviceability for Single Address
POST	<code>/api/CarrierServiceability/GetCarrierServiceabilityBulk</code>	Carrier Serviceability Bulk
GET	<code>/api/CarrierServiceability/GetCarrierUSServiceability</code>	Carrier Serviceability for Single Address
POST	<code>/api/CarrierServiceability/GetCarrierUSServiceabilityBulk</code>	Carrier Serviceability Bulk
GET	<code>/api/CarrierServiceability/GetCarrierIntServiceability</code>	Carrier Serviceability for Single Address
POST	<code>/api/CarrierServiceability/GetCarrierIntServiceabilityBulk</code>	Carrier Serviceability Bulk

At the bottom left, there is a status bar showing `[BASE URL: , API VERSION: v1]`. At the bottom right, there is a **VALID** button and a Swagger logo.

Online help, testing and documentation

Expanding either option will expose parameters and additional information about the operation. Calls to **waCarrierELI** need to be authorized, which is why there is a **red exclamation mark (!)** on the page. Clicking on the **red exclamation mark (!)** on the right side of the screen will start the authorization process. The **red exclamation mark (!)** will appear after expanding either operation endpoint, but the bulk is shown below:

GET /api/CarrierServiceability/GetCarrierUSServiceability Carrier Serviceability for Single Address

Implementation Notes
Returns serviceability data about a single address

Response Class (Status 200) !
OK

Model **Example Value**

```
{
  "ClientId": "string",
  "ClientTrackingId": "string",
  "ServiceReturnStatus": {
    "ResponseCode": 0,
    "ResponseMessage": "string",
    "CurrentPeriodRequestsUsed": 0,
    "CurrentPeriodRequestsAvailable": 0
  },
  "CarrierServiceabilityResult": {
```

Response Content Type **application/json**

Parameters

Parameter	Value	Description	Parameter Type	Data Type
trackingId	<input type="text" value="(required)"/>	An ID supplied by the client to track or name this request	query	string
address	<input type="text" value="(required)"/>	Main Address with optional suite information	query	string
city	<input type="text" value="(required)"/>	City	query	string
state	<input type="text" value="(required)"/>	State abbreviation	query	string
zip	<input type="text" value="(required)"/>	Five digit ZIPCode	query	string
clientAddressKey	<input type="text"/>	ID of the address	query	string

After clicking on the **red exclamation mark (!)**, an overlay popup will require the user to select a scope (only one option is currently available). Then click **authorize**.

Online help, testing and documentation

Available authorizations

Select OAuth2.0 Scopes

OAuth2 Client Credentials Grant Flow

Scopes are used to grant an application different levels of access to data on behalf of the end user. Each API may declare one or more scopes. [Learn how to use](#)

API requires the following scopes. Select which ones you want to grant to Swagger UI.

Authorization URL:
flow: application

TWC_MIDAS_Api_CarrierServiceability_Read
Read access to protected resources

Next, fill in the required **carrierRequestAddresses** parameter. If the **example value** to the right is clicked, it will fill in the parameter with the proper format.

GET /api/CarrierServiceability/GetCarrierUSServiceability Carrier Serviceability for Single Address

Implementation Notes
Returns serviceability data about a single address

Response Class (Status 200)
OK

Model **Example Value**

```
{
  "clientId": "string",
  "clientTrackingId": "string",
  "serviceReturnsStatus": {
    "responseCode": 0,
    "responseMessage": "string",
    "currentPeriodRequestsUsed": 0,
    "currentPeriodRequestsAvailable": 0
  },
  "carrierServiceabilityResult": {
```

Response Content Type **application/json**

Parameter	Value	Description	Parameter Type	Data Type
trackingId	(required)	An ID supplied by the client to track or name this request	query	string
address	(required)	Main Address with optional suite information	query	string
city	(required)	City	query	string
state	(required)	State abbreviation	query	string
zip	(required)	Five digit ZIPCode	query	string
clientAddressKey		ID of the address	query	string

[Hide Response](#)

Curl

```
curl -X GET --header 'Accept: application/json' --header 'Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIngicmE6InEick
```

Online help, testing and documentation

Finally, click on “try it out” and all of the details of the call and the response will appear below.

Curl

```
curl -X GET --header 'Accept: application/json' --header 'Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIngidCI6ImEzck
```

Request URL

```
https://ell-ws-carrier-uat.charter.com/api/CarrierServiceability/GetCarrierUSServiceability?trackingId=Test&address=7800%20crescen
```

Response Body

```
{
  "ClientId": "swaggerUI",
  "ClientTrackingId": "Test",
  "ServiceReturnStatus": {
    "ResponseCode": 2001,
    "ResponseMessage": "One match found",
    "CurrentPeriodRequestsUsed": 6,
    "CurrentPeriodRequestsAvailable": 94
  },
  "CarrierServiceabilityResult": {
    "AddressStatus": {
      "StatusCode": 1,
      "StatusMessage": "Success"
    },
    "ClientAddressKey": "Not Supplied",
    "RequestNumber": 1,
    "ValidatedAddress": {
      "MapPoint": {
        "Longitude": -80.914842,
        "Latitude": 35.145062
      }
    }
  }
}
```

Response Code

200

Response Headers

```
{
  "cache-control": "no-cache",
  "content-length": "1178",
  "content-type": "application/json; charset=utf-8",
  "date": "Wed, 27 Jan 2021 18:15:36 GMT",
  "expires": "1",
  "pragma": "no-cache",
  "server": "Microsoft-IIS/10.0",
  "strict-transport-security": "max-age=31536000; includeSubDomains",
  "x-aspnet-version": "4.0.30319",
  "x-content-type-options": "nosniff",
  "x-frame-options": "SAMEORIGIN",
  "x-powered-by": "ASP.NET",
  "x-xss-protection": "1; mode=block"
}
```

Security

waCarrierELI uses OAuth 2.0 (Client Credentials Grant) for authorizing access to the API. At the end of this document, we have included an example code for obtaining a security token and calling the single address API method. It is suggested that consumers become familiar with the OAuth2 specifications.

(<http://oauth.net/2/>)

Access token URL:

Production:

<https://eli-security.charter.com/core/connect/token>

<https://eli-security.charter.com/core/.well-known/openid-configuration>

QA:

<https://eli-security-uat.charter.com/core/connect/token>

<https://eli-security-uat.charter.com/core/.well-known/openid-configuration>

Single address API

Request

For a single address request, the URL will contain the following query string parameters:

1. **TrackingId:** This is a string provided by the requestor and will be returned in the response to provide a mechanism for customer request/response synchronization. Limited to 50 characters.
2. **Address:** This is the street number and name of the requested address. Limited to 50 characters.
3. **City:** This is the city address. Limited to 100 characters.
4. **State:** This is the state address. Limited to 2 characters.
5. **Zip:** This is the zip code of the location. Limited to 10 characters but only the first 5 will be used.
6. **ClientAddressKey:** This is an optional unique address identifier provided by the client.

An example URL for the single address request API method is provided below:

<https://eli-ws-carrier-uat.charter.com/api/CarrierServiceabilityGetCarrierUSServiceability?trackingId=Test&address=7800%20Crescent%20Executive%20drive&city=charlotte&state=NC&zip=28217>

Note: All parameters are required in the query string, but the parameters may contain empty string values like the zip code in the above example. The address parameters will be validated per the Spectrum Enterprise geocoding requirements. If the address is invalid, it will have a primary reason for the address' invalidity returned. This reason should be used to clean up the address and resubmit.

Response

The response content will be returned as a string formatted in JavaScript object notation (JSON). The **ClientId** (derived from the "authorization header token") and **trackingId** provided in the request will be present in the response as well as the validated address. The JSON response will also contain 0 or 1 address results that the Spectrum Enterprise geocoder has matched for the submitted address. If no geocode results are present, the geocoder either did not find any matches or the address submitted was invalid. In the latter case, one or many validation reason codes/messages will be returned in the response.

Note: For the case of multiple address candidates returned by the geocoder, no result addresses will be returned, and a status message will be provided indicating multiple/ambiguous candidates returned. Each submitted address must geocode to one address.

The geocoded address result is returned in the **ValidatedAddress** object. Also, for qualifying clients, geocoder location details (e.g., latitude/longitude/CensusBlock, etc.) will be provided as well in the **MapPoint** member of the **ValidatedAddress** object. If "client" is not qualified for geocoder detail, the **MapPoint** object will be null.

For each address submitted to the service, an algorithm is applied to attempt to match the address to a Spectrum Enterprise location entity in order to obtain the serviceability status, CLLI, LATA, etc. for that building. This fiber/serviceability data is provided in the "serviceability" object and is included in the response. If a building match is not found, the "serviceability" object is null. The combined geocoded address and fiber result are represented in the **CarrierServiceabilityResult** object.

Sample response single API

<https://eli-ws-carrier-uat.charter.com/api/CarrierServiceability/GetCarrierUSServiceability?trackingId=Test&address=7800%20crescent%20Executive%20drive&city=charlotte&state=NC&zip=28217>

```
{
  "ClientId": "swaggerUI",
  "ClientTrackingId": "Test",
  "ServiceReturnStatus": {
    "ResponseCode": 2001,
    "ResponseMessage": "One match found",
    "CurrentPeriodRequestsUsed": 5,
    "CurrentPeriodRequestsAvailable": 95
  },
  "CarrierServiceabilityResult": {
    "AddressStatus": {
      "StatusCode": 1,
      "StatusMessage": "Success"
    },
    "ClientAddressKey": "Not Supplied",
    "RequestNumber": 1,
    "ValidatedAddress": {
      "MapPoint": {
        "Longitude": -80.914842,
        "Latitude": 35.145062
      },
      "Address1": "7800 CRESCENT EXECUTIVE DR",
      "Address2": "",
      "City": "CHARLOTTE",
      "State": "NC",
      "ZipCode": "28217"
    },
    "RequestAddress": {
      "ClientAddressKey": null,
      "Address": "7800 crescent Executive drive",
      "City": "charlotte",
      "State": "NC",
      "Zipcode": "28217"
    },
    "Serviceability": {
      "BuildingKey": 13848123,
      "Lata": "422",
      "ClliCode": "CHRMNCJW",
      "WholesaleFiberStatus": "Near-Net",
      "InstallationInterval": "75",
      "PricingTier": "Category 1",
      "LegacyFootprint": "T"
    }
  }
}
```

Bulk address API

Request

Bulk address requests will be done via a post. Each address request in a bulk request has the same requirements as a single address request. The bulk address request is similar to submitting a bunch of single address requests as an array in a post.

An example URL for the bulk address request API method is provided below:

https://eli-ws-carrier-uat.charter.com/swagger/ui/index#!/Carrier/Carrier_GetCarrierUSServiceabilityBulk

Example input of 4 addresses for version 4:

```
{
  "ClientTrackingId": "Test",
  "InputAddresses": [{
    "ClientAddressKey": "1",
    "Address": "7800 crescent Executive",
    "City": "Charlotte",
    "State": "NC",
    "Zipcode": "28217"
  }, {
    "ClientAddressKey": "2",
    "Address": "2217 Silver Crescent Drive",
    "City": "Charlotte",
    "State": "NC",
    "Zipcode": "28273"
  }, {
    "ClientAddressKey": "3",
    "Address": "60 Columbus Circle",
    "City": "Newyork",
    "State": "NY",
    "Zipcode": "10012"
  }, {
    "ClientAddressKey": "4",
    "Address": "400 Atlantic Street 10th Floor",
    "City": "stamford",
    "State": "CT",
    "Zipcode": "06901"
  }
]}
}
```

The **ClientAddressKey** has to be unique within each bulk request, but not from bulk request to bulk request.

Response

The bulk response will just wrap the single address call response in an array called **CarrierServiceabilityResults**. "Tracking information," "your client id," "request period information" and "over status information" can all be found at the end of the response.

Address errors will be noted in the overall **ServiceReturnStatus** element, and details of each specific address failure can be found in the **AddressStatus** section of each **CarrierServiceabilityResult**.

Bulk address API

The following is a sample response for the 4 address sample request for version 4:

```
{
  "CarrierServiceabilityResults": [{
    "AddressStatus": {
      "StatusCode": 1,
      "StatusMessage": "Success"
    },
    "ClientAddressKey": "1",
    "RequestNumber": 1,
    "ValidatedAddress": {
      "MapPoint": {
        "Longitude": -80.914842,
        "Latitude": 35.145062
      },
      "Address1": "7800 CRESCENT EXECUTIVE DR",
      "Address2": "",
      "City": "CHARLOTTE",
      "State": "NC",
      "ZipCode": "28217"
    },
    "RequestAddress": {
      "ClientAddressKey": null,
      "Address": "7800 crescent Executive",
      "City": "Charlotte",
      "State": "NC",
      "Zipcode": "28217"
    },
    "Serviceability": {
      "BuildingKey": 13848123,
      "Lata": "422",
      "ClliCode": "CHRMNCJW",
      "WholesaleFiberStatus": "Near-Net",
      "InstallationInterval": "75",
      "PricingTier": "Category 1",
      "LegacyFootprint": "T"
    }
  }], {
    "AddressStatus": {
      "StatusCode": 1,
      "StatusMessage": "Success"
    },
    "ClientAddressKey": "2",
    "RequestNumber": 2,
    "ValidatedAddress": {
```

Bulk address API

Code continued:

```

        "MapPoint": {
            "Longitude": -80.931639,
            "Latitude": 35.134046
        },
        "Address1": "2217 SILVER CRESCENT DR",
        "Address2": "",
        "City": "CHARLOTTE",
        "State": "NC",
        "ZipCode": "28273"
    },
    "RequestAddress": {
        "ClientAddressKey": null,
        "Address": "2217 Silver Crescent Drive",
        "City": "Charlotte",
        "State": "NC",
        "Zipcode": "28273"
    },
    "Serviceability": {
        "BuildingKey": 13523675,
        "Lata": "",
        "ClliCode": "",
        "WholesaleFiberStatus": "Request a Quote",
        "InstallationInterval": "",
        "PricingTier": "Request a Quote",
        "LegacyFootprint": "T"
    }
}, {
    "AddressStatus": {
        "StatusCode": 1,
        "StatusMessage": "Success"
    },
    "ClientAddressKey": "3",
    "RequestNumber": 3,
    "ValidatedAddress": {
        "MapPoint": {
            "Longitude": -73.983307,
            "Latitude": 40.768845
        },
        "Address1": "60 COLUMBUS CIR",
        "Address2": "",
        "City": "NEW YORK",
        "State": "NY",
        "ZipCode": "10023"
    },

```

Bulk address API

Code continued:

```

        "Address1": "60 COLUMBUS CIR",
        "Address2": "",
        "City": "NEW YORK",
        "State": "NY",
        "ZipCode": "10023"
    },
    "RequestAddress": {
        "ClientAddressKey": null,
        "Address": "60 Columbus Circle",
        "City": "Newyork",
        "State": "NY",
        "Zipcode": "10012"
    },
    "Serviceability": {
        "BuildingKey": 2829836,
        "Lata": "132",
        "CllCode": "NYMXNYUH",
        "WholesaleFiberStatus": "Non-Serviceable",
        "InstallationInterval": "\\\"",
        "PricingTier": "Non-Serviceable",
        "LegacyFootprint": "T"
    }
}, {
    "AddressStatus": {
        "StatusCode": 1,
        "StatusMessage": "Success"
    },
    "ClientAddressKey": "4",
    "RequestNumber": 4,
    "ValidatedAddress": {
        "MapPoint": {
            "Longitude": -73.538909,
            "Latitude": 41.049855
        },
        "Address1": "400 ATLANTIC ST FL 10TH",
        "Address2": "",
        "City": "STAMFORD",
        "State": "CT",
        "ZipCode": "06901"
    },
    "RequestAddress": {

```

Bulk address API

Code continued:

```
        "ClientAddressKey": null,  
        "Address": "400 Atlantic Street 10th Floor",  
        "City": "stamford",  
        "State": "CT",  
        "Zipcode": "06901"  
    },  
    "Serviceability": {  
        "BuildingKey": 126091,  
        "Lata": "920",  
        "CliCode": "SMFRCTAU",  
        "WholesaleFiberStatus": "Near-Net",  
        "InstallationInterval": "75",  
        "PricingTier": "Category 1",  
        "LegacyFootprint": "C"  
    }  
}],  
"ClientId": "swaggerUI",  
"ClientTrackingId": "SOAP UI",  
"ServiceReturnStatus": {  
    "ResponseCode": 2002,  
    "ResponseMessage": "More than one match found",  
    "CurrentPeriodRequestsUsed": 10,  
    "CurrentPeriodRequestsAvailable": 90  
}  
}
```

Request for access token (secure calls)

Spectrum Enterprise Carrier serviceability API adheres to the OAuth 2.0 specification for secure web service access. Note that only the Client Credentials Grant is available for this API at this time. Client applicants that wish to access secure web resources are required to obtain an “access token” prior to making the request to the protected resource (web API) and then submitting that token in the request via the request “authorization header.” Client applicants need to be registered with the Token Issuing Authority (Spectrum Enterprise in this case) and will be provided with an ID and secret key. The client applicants will use this ID and key to request “access tokens” via the OAuth 2.0 Client Credentials Grant from the Token Issuing Authority. Example code for requesting “access tokens” from the Token Issuing Authority is provided below.

Once the “access token” is obtained, the client applicant can make requests to the protected resource with the token until the token expires (after one hour). In this case, the client applicant will then need to request another “access token.” It is the responsibility of the client applicant to make the request for the original “access token” and to handle the use case of token expiration and re-request. For example, .NET code for making requests to the protected resource (Carrier serviceability API) is provided below.

Example for invoking the Carrier serviceability single address HTTP GET request: Test URL (NOT production ready code) C#/.NET

```
using IdentityModel.Client; //nuget> install-package IdentityModel
using Newtonsoft.Json; //nuget> install-package Newtonsoft.Json
using System;
using System.Net.Http;
namespace waCarrierSampleCode {
    class Program {
        static void Main(string[] args){
            //Get access token
            var token = RequestToken();
            Console.WriteLine("Access Token:\n");
            Console.WriteLine(token.AccessToken + "\n\n");

            //Call service
            var response = CallService(token.AccessToken);
        }
    }
}
```


Request for access token (secure calls)

```

        Console.WriteLine("Service Response:");

        //To format the Json string for display
        dynamic parsedJson = JsonConvert.DeserializeObject(response);
        Console.WriteLine(JsonConvert.SerializeObject(parsedJson, Formatting.Indented));

        //Wait for user input so the console window doesn't close
        Console.WriteLine("\n\nPress Enter to continue");
        Console.ReadLine();
    }

    static TokenResponse RequestToken(){
        var client = new TokenClient(
            "https://eli-security-uat.charter.com/core/connect/token ",
            "YOUR_CLIENT_ID", //clientid
            "YOUR_SECRET" //secret
        );

        return client.RequestClientCredentialsAsync("TWC_MIDAS_Api_CarrierServiceability_Read").Result;
    }

    static string CallService(string token){
        if (token != null){
            using (var httpClient = new HttpClient()){
                httpClient.SetBearerToken(token);
                return httpClient.GetStringAsync("https://eli-ws-carrier-
uat.charter.com/api/CarrierServiceability/GetCarrierServiceability?trackingId=test&address=605 W MAIN ST&city=NEW
LEBANON&state=OH&zip=45345").Result;
            }
        }

        return "No valid token";
    }
}
}
}

```

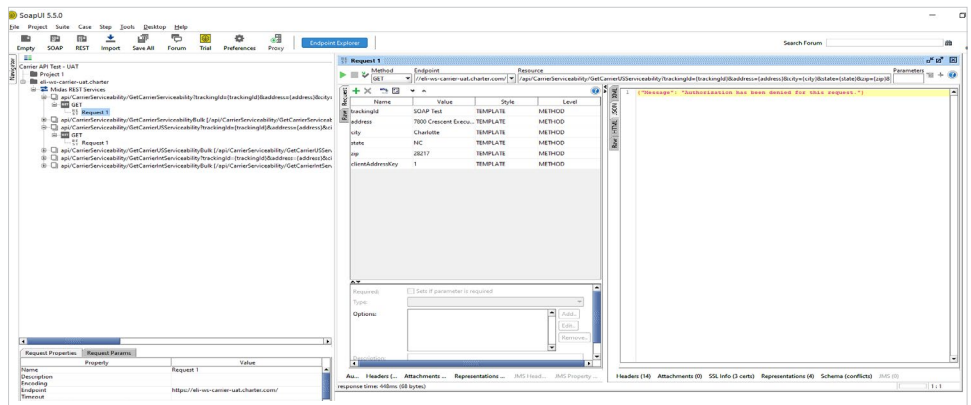
Creating a SoapUI test client for waCarrierEli and securityEli

One of the easiest ways to interact with the **waCarrierEli** API is through SoapUI. Users can easily send a test address and see what the response will be. Getting SoapUI to work with OAuth2 is not always straightforward, therefore, we've included a section on how to set up a workspace with an example project.

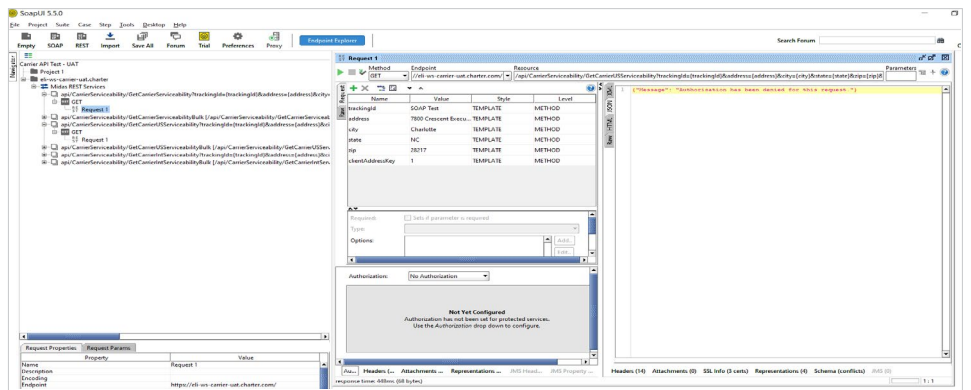
Begin by creating a new workspace; this example is named **waCarrierEliExampleWorkspace**. Once the workspace is created, click on the **rest** button at the top and then click on the **Import WADL...** button. Users can use the WADL from either production or QA, but QA is typically the safer option. If users are testing a new feature only available in QA, then this is the only WADL file that would work. The URL for the QA WADL is as follows:

<https://eli-ws-carrier-uat.charter.com/help/wadl>

Users will then see a screen like the one shown to the right:



Two example requests have already been created with all possible parameters listed in each. In the single address request at the top, add in the **trackingId**, **address**, **city**, **state**, **zip** and optionally the **clientAddressKey**. In this example, it is renamed to “single address request.” Once this request is made, it will generate an automatic notification that states “authorization has been denied for this request.” This response is to be expected since no authorization was sent. It is shown in this screenshot to the right:

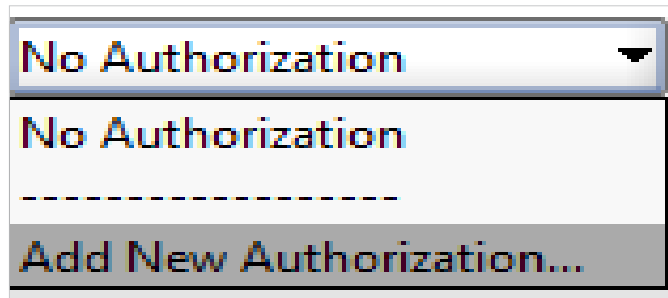


Creating a SoapUI test client for waCarrierEli and securityEli

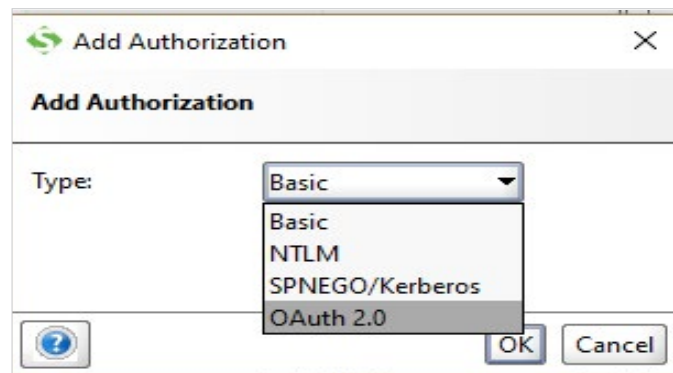
The logical next question is, “how do I send authorization data along with this request?” That is accomplished by using the “auth tab” at the bottom of the request. It will bring up a window like this:



The authorization dropdown lists “no authorization” so will have to add an authorization. First, add a new authorization:

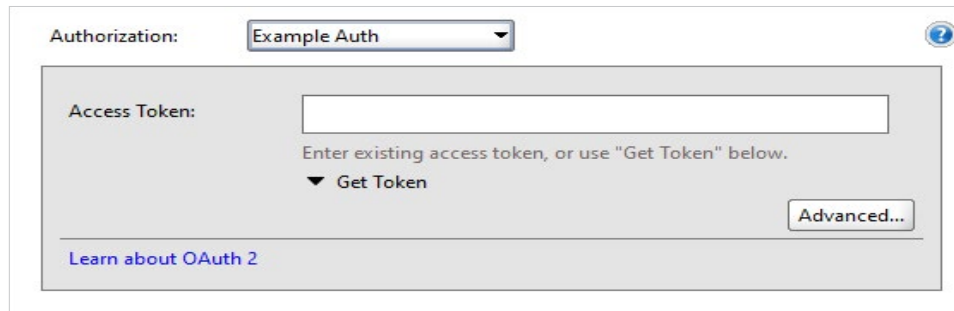


Make sure the type is OAuth 2.0:



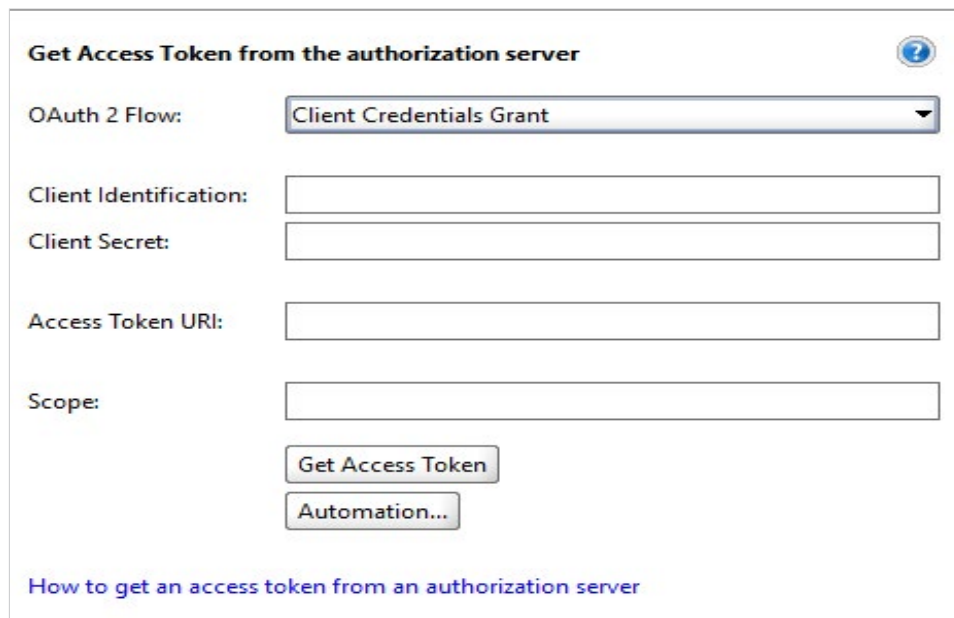
Creating a SoapUI test client for waCarrierEli and securityEli

This example is named “example auth,” and there will be a screen that looks like this:



The screenshot shows the 'Authorization' dialog box in SoapUI. At the top, there is a dropdown menu labeled 'Authorization:' with 'Example Auth' selected. Below this is a text input field for 'Access Token:' with the instruction 'Enter existing access token, or use "Get Token" below.' and a dropdown menu labeled 'Get Token'. To the right of the 'Get Token' dropdown is an 'Advanced...' button. At the bottom left, there is a link that says 'Learn about OAuth 2'.

The next step is to click the “get token” dropdown and then choose “Client Credentials Grant” as so:



The screenshot shows the 'Get Access Token from the authorization server' dialog box. At the top, there is a dropdown menu labeled 'OAuth 2 Flow:' with 'Client Credentials Grant' selected. Below this are four text input fields: 'Client Identification:', 'Client Secret:', 'Access Token URI:', and 'Scope:'. At the bottom, there are two buttons: 'Get Access Token' and 'Automation...'. At the bottom left, there is a link that says 'How to get an access token from an authorization server'.

Update the “client identification” and “client secret” to the client’s credential information and use the “access token URL” of <https://eli-security.charter.com/core/connect/token> for production and <https://eli-security-uat.charter.com/core/connect/token> for QA. Please note that if users get a token from production, the REST call to **waCarrierEli** must be to the production URL. Tokens retrieved from the production environment only work for production, and QA tokens only work for QA.

Creating a SoapUI test client for waCarrierEli and securityEli

The scope is:

TWC_MIDAS_Api_CarrierServiceability_Read.

Users should see something like this now:

Finally, after all data has been filled in, press the **get access token** button. The “access token” should be retrieved and highlighted in green, as shown to the right.

Now when users play the original “single address request” again, they should get a valid response like the one to the right:

```

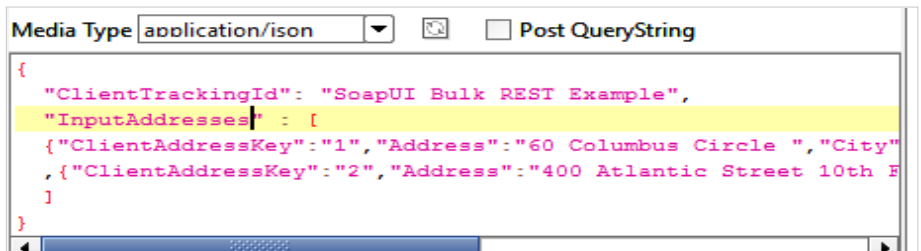
1 | {
2 |   "ClientId": "swaggerUI",
3 |   "ClientTrackingId": "SOAP Test",
4 |   "ServiceReturnStatus": {
5 |     "ResponseCode": 2001,
6 |     "ResponseMessage": "One match found",
7 |     "CurrentPeriodRequestsUsed": 5,
8 |     "CurrentPeriodRequestsAvailable": 95
9 |   },
10 |   "CarrierServiceabilityResult": {
11 |     "AddressStatus": {
12 |       "StatusCode": 1,
13 |       "StatusMessage": "Success"
14 |     },
15 |     "ClientAddressKey": "1",
16 |     "RequestNumber": 1,
17 |     "ValidatedAddress": {
18 |       "MapPoint": {
19 |         "Longitude": -80.914942,
20 |         "Latitude": 35.145062
21 |       },
22 |       "Address1": "7800 CRESCENT EXECUTIVE DR",
23 |       "Address2": "",
24 |       "City": "CHARLOTTE",
25 |       "State": "NC",
26 |       "ZipCode": "28217"
27 |     },
28 |     "RequestAddress": {
29 |       "ClientAddressKey": null,
30 |       "Address": "7800 Crescent Executive Drive",
31 |       "City": "Charlotte",
32 |       "State": "NC",
33 |       "Zipcode": "28217"
34 |     },
35 |     "Serviceability": {
36 |       "BuildingKey": 13848123,
37 |       "Lata": "422",
38 |       "CliCode": "CHRMNCJW",
39 |       "WholesaleFiberStatus": "Near-Net",
40 |       "InstallationInterval": "75",
41 |       "PricingTier": "Category 1",
42 |       "LegacyFootprint": "T"
43 |     }
44 |   }
45 | }
    
```

Creating a SoapUI test client for waCarrierEli and securityEli

The flow for the bulk request is very similar. The bulk JSON request data is added under the “media type” dropdown. Make sure to select “application/JSON” as the “media type”. Users can apply the following bulk JSON block for testing:

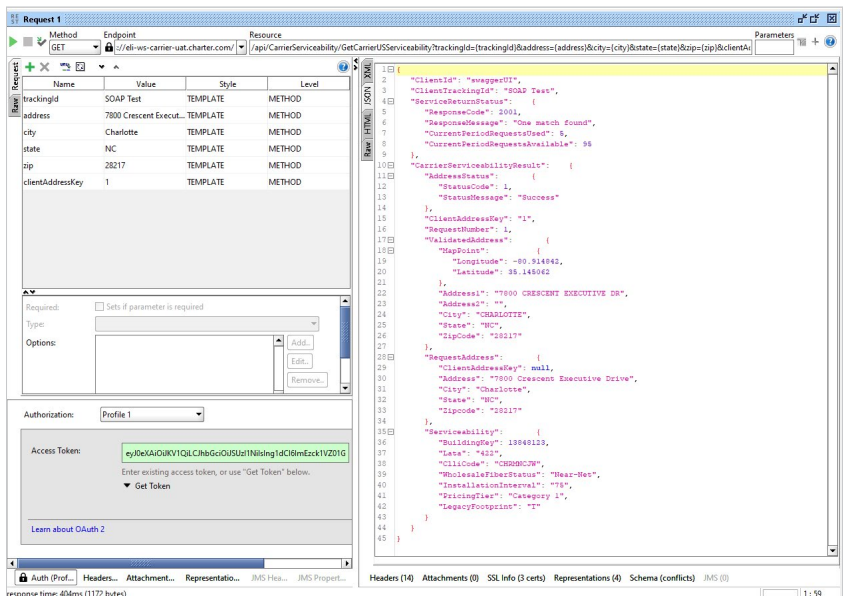
```
{
  "ClientTrackingId": "SoapUI Bulk REST Example",
  "InputAddresses": [{
    "ClientAddressKey": "1",
    "Address": "60 Columbus Circle ",
    "City": "NEW YORK",
    "State": "NY",
    "Zipcode": "10012"
  }, {
    "ClientAddressKey": "2",
    "Address": "400 Atlantic Street 10th Floor",
    "City": "STAMFORD",
    "State": "CT",
    "Zipcode": "06901"
  }
]}
```

It will look like this:



Authorization for this example is easier to set up, and since users just retrieved a token that is good for one hour, all they have to do is select the “example auth” from the authorization dropdown. Users will not have to retrieve a new token for an hour. It should still be highlighted green and look like this:

Running the request to the right will also work.



Creating a SoapUI test client for waCarrierEli and securityEli

This uses SoapUI's authorization framework to add the header parameter to requests. It looks like magic at this point. To see what is really happening, check the HTTP log like so:

The screenshot shows the SoapUI interface for a REST client. The top section displays the request configuration:

- Method:** GET
- Endpoint:** /eli-ws-carrier-uat.charter.com/
- Resource:** /api/CarrierServiceability/GetCarrierUSServiceability?trackingId={trackingId}&address={address}&city={city}&state={state}&zip={zip}&clientAddressKey={clientAddressKey}

Below the configuration is a table of request parameters:

Name	Value	Style	Level
trackingId	SOAP Test	TEMPLATE	METHOD
address	7800 Crescent Execu...	TEMPLATE	METHOD
city	Charlotte	TEMPLATE	METHOD
state	NC	TEMPLATE	METHOD
zip	28217	TEMPLATE	METHOD

The Authorization section shows a profile named "Profile 1" and an Access Token: eyJ0eXh0aWkiKV1QilCJhbGciOiJSUzI1NiIsIng1dCI6ImEzck1VZ01C...

The right-hand pane displays the JSON response:

```

1 {
2   "ClientId": "swaggerUI",
3   "ClientTrackingId": "SOAP Test",
4   "ServiceReturnStatus": {
5     "ResponseCode": 2001,
6     "ResponseMessage": "One match found",
7     "CurrentPeriodRequestsUsed": 5,
8     "CurrentPeriodRequestsAvailable": 96
9   },
10  "CarrierServiceabilityResult": {
11    "AddressStatus": {
12      "StatusCode": 1,
13      "StatusMessage": "Success"
14    },
15    "ClientAddressKey": "1",
16    "RequestNumber": 1,
17    "ValidatedAddress": {
18      "MapPoint": {
19        "Longitude": -80.914842,
20        "Latitude": 35.145062
21      },
22      "Address1": "7800 CRESCENT EXECUTIVE DR",
23      "Address2": "",
24      "City": "CHARLOTTE",
25      "State": "NC",
26      "ZipCode": "28217"
27    }
28  }
29 }
    
```

At the bottom, the HTTP log shows the following entries:

```

Tue Jan 26 16:39:45 EST 2021:DEBUG:Connection can be kept alive indefinitely
Tue Jan 26 16:39:45 EST 2021:DEBUG:Target requested authentication
Tue Jan 26 16:39:45 EST 2021:WARN:Authentication error: Unable to respond to any of these challenges: {}
Tue Jan 26 16:39:45 EST 2021:INFO:Got response for [Midas REST Services.api/CarrierServiceability/GetCarrierServiceability?trackingId={trackingId}&address={address}&city={city}&state={state}&zip={zip}&clientAddressKey={clientAddressKey}]
Tue Jan 26 16:40:19 EST 2021:DEBUG:Connection closed
Tue Jan 26 17:16:38 EST 2021:DEBUG:Attempt 1 to execute request
Tue Jan 26 17:16:38 EST 2021:DEBUG:Sending request: POST /core/connect/token HTTP/1.1
Tue Jan 26 17:16:38 EST 2021:DEBUG:Receiving response: HTTP/1.1 200 OK
Tue Jan 26 17:16:38 EST 2021:DEBUG:Connection can be kept alive indefinitely
Tue Jan 26 17:16:42 EST 2021:DEBUG:Attempt 1 to execute request
Tue Jan 26 17:16:42 EST 2021:DEBUG:Sending request: GET /api/CarrierServiceability/GetCarrierUSServiceability?trackingId=SOAP%20Test&address=7800%20Crescent%20Executive%20Drive&city=Charlotte&state=NC&zip=28217&clientAddressKey=1 HTTP/1.1
Tue Jan 26 17:16:42 EST 2021:DEBUG:Receiving response: HTTP/1.1 200 OK
Tue Jan 26 17:16:42 EST 2021:DEBUG:Connection can be kept alive indefinitely
Tue Jan 26 17:16:42 EST 2021:INFO:Got response for [Midas REST Services.api/CarrierServiceability/GetCarrierServiceability?trackingId={trackingId}&address={address}&city={city}&state={state}&zip={zip}&clientAddressKey={clientAddressKey}]
Tue Jan 26 17:17:10 EST 2021:DEBUG:Connection closed
Tue Jan 26 17:17:15 EST 2021:DEBUG:Connection closed
    
```

In it, there is an "authorization: bearer" with the token retrieved from the security service that has been added to the request to **waCarrierEli**. Without this token, the request will fail.

Please note, if users are using the example workspace, they must reset the paths to where they place the workspace and project files for SoapUI.

About Spectrum Enterprise

Spectrum Enterprise, a part of Charter Communications, Inc., is a national provider of scalable, fiber technology solutions serving America's largest businesses and communications service providers. The broad Spectrum Enterprise portfolio includes networking and managed services solutions: Internet access, Ethernet access and networks, Voice and TV solutions. Spectrum Enterprise's industry-leading team of experts works closely with clients to achieve greater business success by providing solutions designed to meet their evolving needs. More information about Spectrum Enterprise can be found at enterprise.spectrum.com.