

# Dynamic Connections API User Guide

Version 2.0

March 15, 2024

---



## Contents

---

<b>Introduction .....</b>	<b>5</b>
Request State Diagram .....	6
Connection State Diagram .....	6
API Security .....	6
<b>Dynamic Connections API Syntax.....</b>	<b>7</b>
Query UNIs by Account Number .....	7
Overview .....	7
URL .....	7
Input Parameters in HTTP GET URL .....	7
Response .....	7
Query Partner Interconnects .....	9
Overview .....	9
URL .....	9
Input Parameters in HTTP GET URL .....	9
Response .....	10
Create Connection .....	12
Overview .....	12
URL .....	12
Input Parameters in HTTP POST Body .....	12
Response .....	16
Delete Connection .....	19
Overview .....	19
URL .....	19
Input Parameters in HTTP DELETE Body .....	19
Response .....	19
Query Connections by Account Number .....	21
Overview .....	21
URL .....	21
Input Parameters in HTTP GET URL .....	21
Response .....	22
Query Requests by Account ID .....	26
Overview .....	26
URL .....	26
Input Parameters in HTTP GET URL .....	26
Response .....	27
Add CE-VLAN to EVC .....	31
Overview .....	31
URL .....	31
Input Parameters in HTTP POST Body .....	31
Response .....	31
Remove CE-VLAN from EVC .....	33
Overview .....	33
URL .....	33
Input Parameters in HTTP DELETE Body .....	33
Response .....	33
Query CE-VLANs .....	35

Overview .....	35
URL .....	35
Input Parameters in HTTP GET URL .....	35
Response .....	35
Create Connection to AWS .....	37
Overview .....	37
URL .....	37
Input Parameters in HTTP POST Body .....	37
Response .....	39
Delete Connection to AWS.....	40
Overview .....	40
URL .....	41
Input Parameters in HTTP DELETE Body .....	41
Response .....	41
Query Connections to AWS by Account .....	42
Overview .....	42
URL .....	43
Input Parameters in HTTP GET URL .....	43
Response .....	44
Query EVCs by Account.....	47
Overview .....	47
URL .....	47
Input Parameters in HTTP GET URL .....	47
Response .....	47
Query End Points by Account .....	52
Overview .....	52
URL .....	52
Input Parameters in HTTP GET URL .....	52
Response .....	52
Query AWS Requests by Account .....	54
Overview .....	54
URL .....	54
Input Parameters in HTTP GET URL .....	54
Response .....	55
Query Locations by Location ID .....	58
Overview .....	58
URL .....	58
Input Parameters in HTTP GET URL .....	58
Response .....	58
<b>Digest-Based Security .....</b>	<b>60</b>
On-Boarding for Digest Based API Access .....	60
Requirements For a Digest-Based Request .....	60
Generating A Digest .....	61
Helpful Tips for Digest Generation .....	61
Error Message: "Digest Time provided not within 15 minutes".....	61
Error Message: "Digest provided does not match".....	61
Digest Sample Code .....	62
How to Generate a Digest.....	62

Troubleshooting .....	64
Why Did My Call Fail? .....	64
Interpreting Results: Mediation Id .....	64
<b>OAuth2-Based Security .....</b>	<b>65</b>
On-Boarding for OAuth2 OpenAPI Access .....	65
OAuth Production and Test Environments .....	65
OAuth Request URI: Test Environment .....	65
OAuth Request URI: Production Environment .....	65
Accessing OAuth-Based API Services .....	65
Access Token Server - TEST .....	65
Access Token Server - PRODUCTION .....	65
Making an OAuth-Based API Request .....	66
Step 1 – Access Token Request .....	66
OAuth Sample Code .....	66
Step 2 - The API Request .....	67
<b>Interpreting API Results: HTTP Status Codes .....</b>	<b>68</b>
Standard HTTP Status Codes .....	68

## Introduction

The Dynamic Connection service is a layer 2 Ethernet Virtual Connection (EVC) network service developed and offered by Level 3 Communications. The service can be scheduled on-demand in real-time from our customer portal or invoked real-time from systems by calling our APIs.

The Dynamic Connection application is a C++ multi-threaded, high-performance, high-reliable, real-time application that serves dynamic connection requests from either customer portal or direct API calls. The request type can be one of the following types: (a) connection creation; (b) connection deletion; (c) connection status check; (d) request status query. During connection creation processing, the application checks request parameters, validates business rules, assigns proper resources, builds connection paths, auto-configures network elements on the paths, and notifies requestors as well as Operation Supporting Systems (OSS) and Business Supporting Systems (BSS) about the processing results. During connection deletion processing, the application validates business rules, releases network resources accordingly. During connection status check processing, the application queries its database and returns proper status to the requestors.

The first use case is to create an EVC from one multiplexed UNI endpoint 1 (which has physical Ethernet connection to a customer's network) to another multiplexed UNI endpoint 2 (which could be in a different geolocation). This use case is referred to as UNI – UNI scenario, as described in Figure 1.



Figure 1 UNI to UNI Network Topology

The second use case is to create an EVC from a multiplexed UNI endpoint 1 to an eLynk interface endpoint 2. An eLynk interface can have customer VLAN remapped. This use case is referred to as UNI – ELYNK scenario, as described in Figure 2.

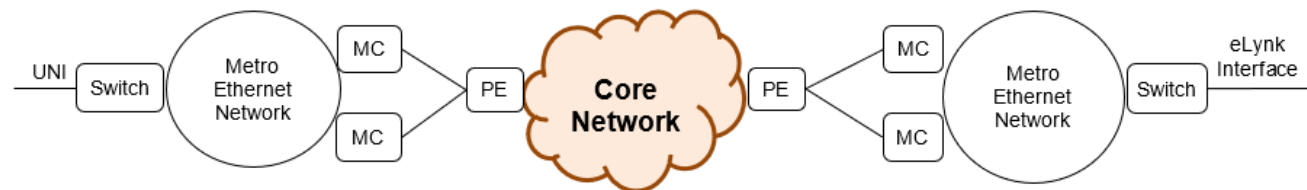


Figure 2 UNI to eLynk Network Topology

The third use case is to create an EVC from a multiplexed UNI endpoint 1 to an eLynk interface endpoint 2 and then to Amazon Web Service (AWS) cloud. This use case is referred to as UNI – ELYNK – AWS cloud scenario, as described in Figure 3.

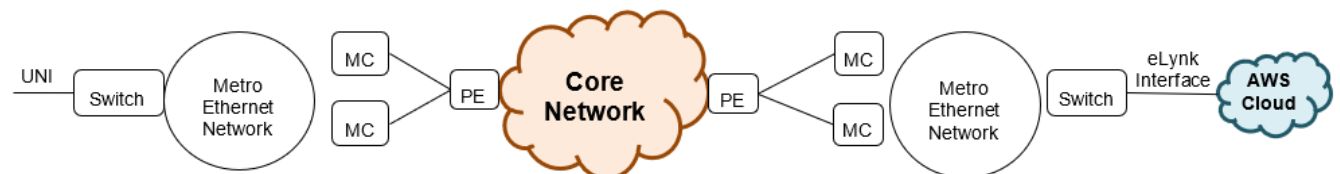


Figure 3 UNI to eLynk to AWS Cloud Network Topology

The fourth use case is to create an EVC from an AWS region 1 to an AWS region 2. This use case is referred to as AWS – AWS cloud scenario, as described in Figure 4.

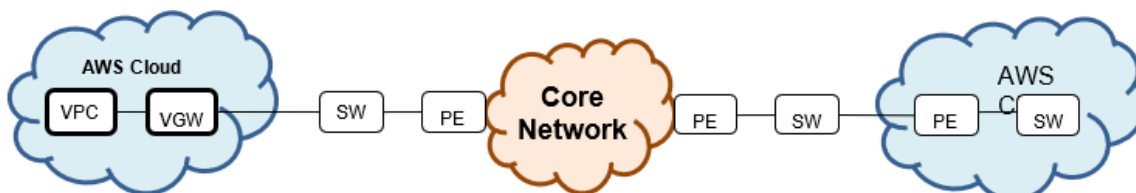


Figure 4. AWS to AWS Cloud Network Topology

## Request State Diagram

Many requests in this context are asynchronous requests. The following state diagram shows the results of such request in various processing stages. Each state is represented by a status code which can be queried using proper API.

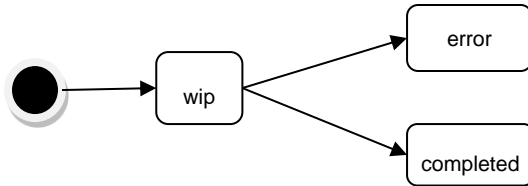


Figure 5 Request State Diagram

## Connection State Diagram

A connection state is composed from “create connection” request state and “delete connection” request state. It is shown in the following diagram.

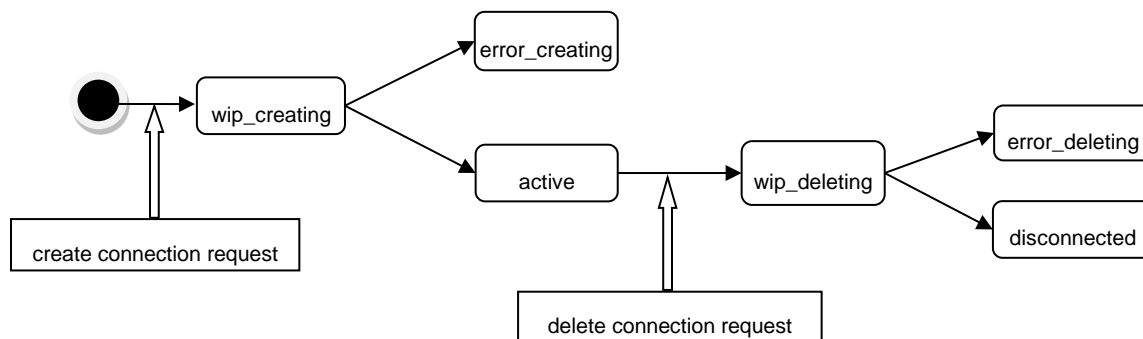


Figure 6 Connection State Diagram

## API Security

There are two authorization methods to access Level 3 API services:

- The OAuth 2.0 Framework
- Level 3 propriety [Digest Authorization](#) method.

Due to the fact that the official [OAuth 2.0 specification](#) is widely recognized as an industry standard, it should be noted that it has also been chosen as the go-forward method of authorization for all Level 3 OpenAPI services. Although the OAuth2 access method is **not** yet available in production, the currently supported Digest method will be deprecated in the near future so please be aware of this change for planning purposes. It is anticipated that the new OAuth2 access method will be available in mid to late Q4.

Please refer to the following sections for details on the two API security methods:

- [OAuth Based API Security](#)
- [Digest Based API Security](#)

## Dynamic Connections API Syntax

### Query UNIs by Account Number

#### Overview

A user can use this API request via the HTTP GET method to query for available multiplexed UNIs and eLynk interfaces for a given billing account number. Each multiplexed UNI or eLynk interface is identified by a unique ID and can be used as an endpoint in a dynamic connection. A call using this API is synchronous.

#### URL

Example:

<https://api.level3.com/Network/v2/DynamicConnection/unis?billingAccountNumber={billingAccountNumber}>

#### Input Parameters in HTTP GET URL

The input parameters are encoded in URL and sent in HTTP GET.

Name	Type	Requirement	Value (example)	Description
billingAccountNumber	string	mandatory	123456	The unique ID identifying a customer account in Level 3 inventory system.

Example:

`.../unis?billingAccountNumber=123456`

#### Response

A HTTP response is returned after the request is accepted and processed by the dynamic controller.

#### Successful Response

A response with status code of “200” is returned after a request is accepted and successfully processed. The following data encoded in JSON format is returned in the response body.

Name	Type	Value (example)	Description
receivedParams	JSON object	{"billingAccountNumber": "123456"}	The echo of the request parameters in JSON.
unis	JSON list object	[ {...}, {...} ]	The list of available multiplexed UNIs and eLynk Interfaces for the given account.

Each JSON object in the returned list has the following attributes:

Name	Type	Value (example)	Description
uniServiceId	string	59/KGFN/108765/TWCS	The unique identifier of a UNI in Level 3 inventory system.
serviceAddress	string	2001 6TH, SEATTLE, WA 98121, US	The physical address of the UNI device.
locationId	string	STTLWAWB	The unique identifier of a location.
latitude	string	47.614379	The latitude of the location.
longitude	string	-122.338599	The longitude of the location.
uniPortSpeed	string	1000 Mbps	The port speed of the UNI including unit.
bandwidthAvailable	integer	100000000	The available bandwidth on the UNI in bits per second (bps).
bandwidthUsed	integer	900000000	The bandwidth used on the UNI in bits per second (bps).
ceVlansInUse	JSON array	["111","222"]	The list of ce-vlans used on the UNI across all active EVCs.
ceVlansAvailable	JSON array	["2-110","112-221","223-4094"]	The list of available ce-vlans on the UNI from Level 3 perspective.

Example:

```
{
  "receivedParams":
  {
    "billingAccountNumber": "123456"
  },
  "unis":
  [
    {
      "uniServiceId": "59/KGFN/108765/TWCS",
      "serviceAddress": "2001 6TH, SEATTLE, WA 98121, US",
      "locationId": "STTLWAWB",
      "latitude": "47.614379",
      "longitude": "-122.338599",
      "uniPortSpeed": "1000 Mbps",
      "bandwidthAvailable": 100000000,
      "bandwidthUsed": 900000000,
      "ceVlansInUse": ["111","222"],
      "ceVlansAvailable": ["2-110","112-221","223-4094"]
    }
  ]
}
```



## Failure Response

A response with some failure HTTP status code is returned after the request is rejected. The following data encoded in JSON format is returned in the response body to carry more details.

Name	Type	Value (example)	Description
statusCode	integer	400	The HTTP status code following HTTP standard.
code	integer	400003	A Level 3 generated code combining HTTP code with sub-category code.
message	string	Missing data	Failure info at summarized level.
detail	string	The billingAccountNumber is missing.	Info in detail about the failure.

Example:

```
{
  "exception":
  {
    "statusCode": "400",
    "code": "400003",
    "message": "Missing data",
    "detail": "The billingAccountNumber is missing."
  }
}
```

## Query Partner Interconnects

### Overview

A user can use this API request via the HTTP GET method to query for available partner interconnects. Each partner interconnect point (aka, eLynk interface) is identified by a unique ID and can be used as an endpoint in a dynamic connection. A call using this API is synchronous.

### URL

Example:

<https://api.level3.com/Network/v2/DynamicConnection/partnerInterconnects>

### Input Parameters in HTTP GET URL

The input parameters are encoded in URL and sent in HTTP GET.

Name	Type	Requirement	Value (example)	Description
------	------	-------------	-----------------	-------------

Example:

```
.../partnerInterconnects
```

## Response

A HTTP response is returned after the request is accepted and processed by the dynamic controller.

### Successful Response

A response with status code of “200” is returned after a request is accepted and successfully processed. The following data encoded in JSON format is returned in the response body.

Name	Type	Value (example)	Description
receivedParams	JSON object	{...}	The echo of the request parameters in JSON.
partnerInterconnects	JSON list object	[ {...}, {...} ]	The list of available interconnects to partners.

Each JSON object in the returned list contains the attributes described in the table below.

The “partnerInterconnectProfile” has the following attributes:

Name	Type	Value (example)	Description
provider	string	aws	The provider name of the partner interconnect.
region	string	us-west-1	The region of the partner interconnect.
id	string	dxcon-fh7bhf10	The ID of the partner interconnect managed by the partner.
name	string	San Jose eLynk PSP Service Alias 63/KGFN/105601/TWCS	The name of the partner interconnect.
location	string	EqSV5	The location of the partner interconnect.
device	string	EqSV5-3ee0qp20oxwt3	The device of the partner interconnect.
state	string	available	The state of the partner interconnect.

Example:

```
{
  "receivedParams":
    {
      "...": ""
    },
  "partnerInterconnects":
    [
      {
        "uniServiceId": "63/KGFN/105601/TWCS",
        "serviceAddress": "11 GREAT OAKS, SAN JOSE, CA 95119, US",
        "locationId": "SNJUCACL",
        "longitude": "-121.780935",
        "latitude": "37.242066",
        "uniPortSpeed": "10000 Mbps",
        "bandwidthAvailable": 5798000000,
        "bandwidthUsed": 4202000000,
        "ceVlansInUse": ["111", "222"],
      }
    ]
}
```

```

    "ceVlansAvailable": ["2-110","112-221","223-4094"]
  "partnerInterconnectProfile":
  {
    "provider":"aws",
    "region":"us-west-1",
    "id":"dxcon-fh7bhf10",
    "name":"San Jose eLynk PSP Service Alias 63/KGFN/105601/TWCS",
    "location":"EqSV5",
    "device":"EqSV5-3ee0qo20oxwt3",
    "state":"available"
  }
}
]
}

```

## Failure Response

A response with some failure HTTP status code is returned after the request is rejected. The following data encoded in JSON format is returned in the response body to carry more details.

Name	Type	Value (example)	Description
statusCode	integer	400	The HTTP status code following HTTP standard.
code	integer	400003	A Level 3 generated code combining HTTP code with sub-category code.
message	string	Missing data	Failure info at summarized level.
detail	string	The billingAccountNumber is missing.	Info in detail about the failure.

Example:

```

{
  "exception":
  {
    "statusCode": "400",
    "code": "400003",
    "message": "Missing data",
    "detail": "The billingAccountNumber is missing."
  }
}

```

## Create Connection

### Overview

A layer 2 Ethernet Virtual Connection (EVC) between two network endpoints can be dynamically created from customer portal or application by calling this API request via the HTTP POST method. A network endpoint is a logical concept, defined by MEF (Metro Ethernet Forum) as EVC endpoint per UNI. In this context, it can also include info at Level 3 side of a layer 2 connection to AWS cloud. A call using this API is asynchronous. The caller needs to check the processing status and act accordingly using requestId or evcServiceId returned its response when successful.

### URL

Example:

<https://api.level3.com/Network/v2/DynamicConnection/evcs>

### Input Parameters in HTTP POST Body

The input parameters are encoded in JSON format and sent in HTTP POST request body.

Name	Type	Requirement	Value (example)	Description
userEmail	string	mandatory	john.doe@level3.com	The email address of a user. This address identifies the user. The address is also used for email notifications when the connection is created successfully or deleted successfully. A failure notification will be sent when processing encounters error.
billingAccountNumber	string	mandatory	123456	The customer billing account number, which is used for billing, reporting, and certain validations.
bandwidth	string	mandatory	10000000	The requested bandwidth value in bits per second.
cos	string	mandatory	Basic	The requested "Class of Service" on the connection.
evcName	string	optional	Level3 test 1	An optional name to describe the connection by API user. The name is opaque to Level 3 system.
endPoint1	JSON object	mandatory	{...}	A JSON object that contains parameters for endpoint 1. The parameters contained in the object are described in table below.
endPoint2	JSON object	mandatory	{...}	A JSON object that contains parameters for endpoint 2. The parameters contained in the object are described in table below.

The parameters within endPoint1 or endPoint2 JSON object:

Name	Type	Requirement	Value (example)	Description
uniServiceId	string	mandatory / optional	47/KEFN/103458/TWCS	The unique identifier of UNI or Elynk interface at one endpoint, which is assigned and managed by Level 3 systems. This parameter is ignored when "awsConnectionProfile" is present. It is mandatory when "awsConnectionProfile" is not present.
ceVlan	string	mandatory / optional	11	A list of customer VLAN IDs delimited by space, which identifies customer Ethernet traffic on the connection. This parameter is ignored when "awsConnectionProfile" is present. It is mandatory when "awsConnectionProfile" is not present.
awsConnectionProfile	JSON object	optional	{...}	An optional JSON object that contains connection profile to AWS cloud. This parameter is required only when connecting to AWS cloud from its associated endpoint.

The parameters within awsConnectionProfile JSON object:

Name	Type	Requirement	Value (example)	Description
awsRegionId	string	mandatory	us-east-1	The unique identifier of an AWS region. Valid values: <a href="http://docs.aws.amazon.com/general/latest/gr/ande.html">http://docs.aws.amazon.com/general/latest/gr/ande.html</a>
uniServiceId	string	optional	74/KGFN/110160/TWCS	The unique identifier of an UNI (interconnect to AWS) in Level 3 inventory. This UNI will be used for connection if it is present. If this parameter is absent, an available UNI to the above AWS region will be auto-selected.
ceVlan	string	optional	11	A VLAN ID, which identifies Ethernet traffic to AWS cloud in a region. This parameter is optional. It will be used if it is present. Otherwise, Level 3 system will choose an available value when it's not present or its value is empty.
customerAwsAccount	string	mandatory	123412341234	The customer's account registered in AWS cloud.
customerAsn	string	mandatory	1234	The customer's ASN for IP routing.
autoGenerateBgpPeerAddress	string	mandatory	true or false	A flag to indicate whether peer IP addresses will be auto-generated.
customerBgpPeerAddress	string	optional	10.0.0.1/30	The customer's destination IPv4 CIDR AWS should use to send traffic to, which is required only when "autoGenerateBgpPeerAddress" flag is set to "false". It will be auto-generated when "autoGenerateBgpPeerAddress" flag is set to "true".
awsBgpPeerAddress	string	optional	10.0.0.2/30	The AWS's destination IPv4 CIDR the customer should use to send traffic to, which is required only when "autoGenerateBgpPeerAddress" flag is set to "false". It will be auto-generated when

Name	Type	Requirement	Value (example)	Description
				“autoGenerateBgpPeerAddress” flag is set to “true”.
autoGenerateBgpAuthenticationKey	string	mandatory	true or false	A flag to indicate whether a BGP authentication key will be auto-generated.
bgpAuthenticationKey	string	optional	secret	The password (authentication key) that will be used to authenticate BGP session, which is required only when “autoGenerateBgpAuthenticationKey” flag is set to “false”. It will be auto-generated when “autoGenerateBgpAuthenticationKey” flag is set to “true”.
autoAcceptConnection	string	mandatory	true or false	The flag to indicate whether auto-accepting a connection on customer’s behalf is allowed.
virtualGatewayId	string	optional	vgw-test123	The identifier of the virtual private gateway that will be attached to the virtual interface being created. It is only required when the “autoAcceptConnection” flag is set to “true”. It is ignored when the “autoAcceptConnection” flag is set to “false”.
arnSecurityToken	string	optional	arn:aws:iam::1234:role/xyz	The ARN security role which will be used to auto-accept ownership of the virtual interface being created. It is only required when the “autoAcceptConnection” flag is set to “true”. It is ignored when the “autoAcceptConnection” flag is set to “false”.

Example for endpoint 1 (UNI) to endpoint 2 (UNI):

```

{
  "userEmail": "john.doe@level3.com",
  "billingAccountNumber": "123456",
  "bandwidth": "10000000",
  "cos": "Basic",
  "endPoint1":
  {
    "uniServiceId": "47/KEFN/103458/TWCS",
    "ceVlan": "11"
  },
  "endPoint2":
  {
    "uniServiceId": "54/KFFN/102109/TWCS",
    "ceVlan": "22"
  }
}

```

Example for endpoint 1 (UNI) to endpoint 2 (UNI connecting to AWS cloud):

```
{
  "userEmail": "john.doe@level3.com",
  "billingAccountNumber": "123456",
  "bandwidth": "10000000",
  "cos": "Basic",
  "endPoint1":
  {
    "uniServiceId": "47/KEFN/103458/TWCS",
    "ceVlan": "11"
  },
  "endPoint2":
  {
    "awsConnectionProfile":
    {
      "awsRegionId": "us-east-1",
      "uniServiceId": "74/KGFN/110160/TWCS",
      "customerAwsAccount": "123412341234",
      "customerAsn": "1234",
      "autoGenerateBgpPeerAddress": "false",
      "customerBgpPeerAddress": "10.0.0.1/30",
      "awsBgpPeerAddress": "10.0.0.2/30",
      "autoGenerateBgpAuthenticationKey": "false",
      "bgpAuthenticationKey": "secret",
      "autoAcceptConnection": "true",
      "virtualGatewayId": "vgw-test567",
      "arnSecurityToken": "arn:aws:iam::1234:role/xyz"
    }
  }
}
```

## Response

A HTTP response is returned after the request is accepted while it is being processed asynchronously by the dynamic controller. The processing status of the request can be checked using API in 2.5.

### Successful Response

A response with status code of “202” is returned after the request is accepted and queued to be processed. The following data encoded in JSON is returned in the response body.

Name	Type	Value (example)	Description
requestId	string	a41d00cd-be8f-49ac-a4c2-9082cca28981	A unique ID (GUID) generated by Level 3 dynamic controller to identify the request. This ID can be used to query the request status using API in 0.
evcServiceId	string	VLXX/D00115/LVLC	A unique ID generated by Level 3 dynamic controller to identify the connection. This ID can/should be used to query the connection status and to delete the connection using other APIs.
action	string	createConnection	The action of the request. Valid values: “createConnection”.
status	string	wip	The status of the request. The status will be one of the enumerated strings defined in 0, e.g., “wip” (work in progress), “complete”, or “error”.
receivedParams	JSON object	{...}	The echo of the request parameters in JSON.

Example for endpoint 1 (UNI) to endpoint 2 (UNI):

```
{
  "requestId": "a41d00cd-be8f-49ac-a4c2-9082cca28981",
  "evcServiceId": "VLXX/D00115/LVLC",
  "action": "createConnection",
  "status": "wip",
  "receivedParams":
  {
    "requestId": "jgw_test_10",
    "userEmail": "john.doe@level3.com",
    "billingAccountNumber": "123456",
    "bandwidth": "10000000",
    "cos": "Basic",
    "endPoint1":
    {
      "uniServiceId": "47/KEFN/103458/TWCS",
      "ceVlan": "11"
    },
    "endPoint2":
    {
      "uniServiceId": "54/KFFN/102109/TWCS",
      "ceVlan": "22"
    }
  }
}
```



Example for endpoint 1 (UNI) to endpoint 2 (UNI connecting to AWS cloud):

```
{
  "requestId": "a41d00cd-be8f-49ac-a4c2-9082cca28981",
  "svcServiceId": "VLXX/D00115/LVLC",
  "action": "createConnection",
  "status": "wip",
  "receivedParams":
  {
    "requestId": "jgw_test_10",
    "userEmail": "john.doe@level3.com",
    "billingAccountNumber": "123456",
    "bandwidth": "10000000",
    "cos": "Basic",
    "endPoint1":
    {
      "uniServiceId": "47/KEFN/103458/TWCS",
      "ceVlan": "11"
    },
    "endPoint2":
    {
      "awsConnectionProfile":
      {
        "awsRegionId": "us-east-1",
        "uniServiceId": "74/KGFN/110160/TWCS",
        "customerAwsAccount": "123412341234",
        "customerAsn": "1234",
        "autoGenerateBgpPeerAddress": "false",
        "customerBgpPeerAddress": "10.0.0.1/30",
        "awsBgpPeerAddress": "10.0.0.2/30",
        "autoGenerateBgpAuthenticationKey": "false",
        "bgpAuthenticationKey": "secret",
        "autoAcceptConnection": "true",
        "virtualGatewayId": "vgw-test567",
        "arnSecurityToken": "arn:aws:iam::1234:role/xyz"
      }
    }
  }
}
```

## Failure Response

A response with some failure HTTP status code is returned after the request is rejected. The following data encoded in JSON is returned in the response body to carry more details.

Name	Type	Value (example)	Description
statusCode	integer	503	The HTTP status code following HTTP standard.
code	integer	503001	A Level 3 generated code combining HTTP code with sub-category code.
message	string	Service Unavailable	Failure info at summarized level.
detail	string	Missing parameter billingAccountNumber	Info in detail about the failure.

Example:

```
{
  "exception":
  {
    "statusCode": "503",
    "code": "503001",
    "message": "Service Unavailable",
    "detail": "Missing parameter billingAccountNumber"
  }
}
```

## Delete Connection

### Overview

A layer 2 Ethernet Virtual Connection (EVC) between two EVC end points can be dynamically deleted from customer portal or application by calling this API request via the HTTP DELETE method. A call using this API is asynchronous. The caller needs to check the processing status and act accordingly using the “requestId” returned in the response.

### URL

Example:

<https://api.level3.com/Network/v2/DynamicConnection/evcs/{evcServiceId}>

### Input Parameters in HTTP DELETE Body`

The input parameters are encoded in JSON format and sent in HTTP DELETE request URL.

Name	Type	Requirement	Value (example)	Description
evcServiceId	string	mandatory	VLXX/D00112/LVLC	The unique ID returned by Level 3 platform for identifying the EVC after a successful “createConnection” call.

Example:

```
../evcs/VLXX%2FD00112%2FLVLC
```

### Response

A HTTP response is returned by dynamic controller after the request is accepted while it is being processed asynchronously. The status of the request can be checked using API in 2.5 or 2.6.

#### Successful Response

A response with status code of “202” is returned after the request is accepted and queued to be processed. The following data encoded in JSON format is returned in the response body.

Name	Type	Value (example)	Description
requestId	string	a41d00cd-be8f-49ac-a4c2-9082cca28981	A unique ID (GUID) generated by Level 3 dynamic controller to identify the request. This ID can be used to query the request status.
evcServiceId	string	VLXX/D00112/LVLC	The unique ID identifying the connection in the Level 3 inventory.
action	string	deleteConnection	The action of the request. Valid values: “deleteConnection”.
status	string	wip	The status of the request. The status will be one of the following enumerated strings: “wip” (work in progress), “complete”, or “error”.
receivedParams	JSON object	{...}	The echo of the request parameters.

Example:

```
{
  "requestId": "a41d00cd-be8f-49ac-a4c2-9082cca28981",
  "svcServiceId": "VLXX/D00112/LVLC",
  "action": "deleteConnection",
  "status": "wip",
  "receivedParams":
  {
    "svcServiceId": "VLXX/D00112/LVLC"
  }
}
```

## Failure Response

A response with some failure HTTP status code is returned after the request is rejected. The following data encoded in JSON is returned in the response body to carry more details.

Name	Type	Value (example)	Description
statusCode	integer	503	The HTTP status code following HTTP standard.
code	integer	503001	A Level 3 generated code combining HTTP code with sub-category code.
message	string	Service Unavailable	Failure info at summarized level.
detail	string	Missing parameter svcServiceId	Info in detail about the failure.

Example:

```
{
  "exception":
  {
    "statusCode": "503",
    "code": "503001",
    "message": "Service Unavailable",
    "detail": "Missing parameter svcServiceId"
  }
}
```

## Query Connections by Account Number

### Overview

The connection status of each EVC within a time period for a given customer account can be queried using this API request via the HTTP GET method. A call using this API is synchronous.

### URL

Examples:

<https://api.level3.com/Network/v2/DynamicConnection/evcs?billingAccountNumber={billingAccountNumber}&source=dynamic>

<https://api.level3.com/Network/v2/DynamicConnection/evcs/{evcServiceId}?billingAccountNumber={billingAccountNumber}&source=dynamic>

### Input Parameters in HTTP GET URL

The inputs parameters are encoded in URL and sent using HTTP GET.

Name	Type	Requirement	Value (example)	Description
billingAccountNumber	string	mandatory	123456	The unique ID of a given customer account.
source	string	mandatory	dynamic	The source of EVCs: provisioned or dynamically created. Valid value: dynamic
evcServiceId	string	optional	VLXX/D00001/LV LC	The unique ID identifying an EVC in Level 3 inventory.
startTime	string	optional	2016-08-01T00:00:00Z	The start time of query window specified in ISO8601 format (YYYY-MM-DDTHH24:MM:SSZ). If it's omitted, the start time will be defaulted to "2016-01-01T00:00:00Z".
endTime	string	optional	2016-09-01T00:00:00Z	The end time of query window specified in ISO8601 format (YYYY-MM-DDTHH24:MM:SSZ). If it's omitted, the end time will be defaulted to current time.
connectionStatus	string	optional	active	The connectionStatus to filter out certain records. The empty value or missing "connectionStatus" filter will return all records.

Example 1:

```
.../evcs?billingAccountNumber=123456&source=dynamic
```

Example 2;

```
.../evcs?billingAccountNumber=123456&source=dynamic&startTime=2016-08-01T00%3A00%3A00Z&endTime=2016-09-01T00%3A00%3A00Z
```

## Response

An HTTP response is returned after the request is accepted and processed by the dynamic controller.

### Successful Response

A response with status code of “200” is returned after the request is accepted and processed successfully. The following data encoded in JSON format is returned in the response body.

Name	Type	Value (example)	Description
receivedParams	JSON object	{"billingAccountNumber": "123456"}	The echo of the request parameters in JSON.
evcs	JSON list	[ {...}, {...} ]	The list of available EVCs for the given account.

Each JSON object in the returned list has the following attributes:

Name	Type	Value (example)	Description
evcServiceId	string	VLXX/D00112/LVLC	The unique ID returned by Level 3 platform to identify the EVC after a successful “createConnection” call.
evcName	string	Level3 test 1	The name to describe the connection by API user. The name is opaque to Level 3 system.
connectionStatus	string	active	The status of a connection. Reference to 0 for valid status value in each state.
requestor	string	jgwang	The identifier of a request initiator.
requestId	string	test11	The request ID of “createConnection” or “deleteConnection” calls.
bandwidth	integer	100000000	The requested bandwidth value in bits per second received in “createConnection” call.
cos	string	Basic	The requested "Class of Service" on the connection received in “createConnection” call.
billingAccountNumber	string	123456	The customer billing account number, which is used for billing, reporting, and certain validations received in “createConnection” call.
detail	string		It may carry additional info or be empty.
startDateTime	string	2016-08-15T12:40:42Z	The timestamp when the connection was ready for service (“started”). The timestamp is encoded in ISO8601 format. It may be empty if the connection was created before the query window.
endDateTime	string	2016-08-15T12:40:48Z	The timestamp when the connection was deleted successfully (“end” its service life). The timestamp is encoded in ISO8601 format. It should be empty when the connection is in “active” status.
endPoint1	JSON object	{...}	A JSON object represents an endpoint of the EVC. The parameters contained in the object are described in table below.
endPoint2	JSON object	{...}	A JSON object represents an endpoint of the EVC. The parameters contained in the object are described in table below.
cloudConnections	JSON	[...]	A JSON array of connections to cloud networks. Each item in

	array		the array represents a L2 connection between an endpoint and a cloud network. The only supported cloud provider is "aws" (as of 2017-08-28). The array is empty when there are no cloud connections.
--	-------	--	--

The parameters within each endPoint JSON object:

Name	Type	Value (example)	Description
uniServiceId	string	47/KEFN/103458/TWCS	Another unique identifier of UNI or Elynk interface at the endpoint, which is assigned and managed by Level 3 systems.
ceVlans	JSON array	["11","22"]	A list of CE-VLAN IDs which identify customer Ethernet traffic on the connection.

Example with no cloud connection:

```
{
  "evcs":
  [
    {
      "evcServiceId": "VLXX/D00112/LVLC",
      "evcName": "Level3 test 1",
      "connectionStatus": "active",
      "requestor": "jgwang",
      "requestId": "test11",
      "bandwidth": "100000000",
      "cos": "Basic",
      "detail": "",
      "billingAccountNumber": "123456",
      "startDateTime": "2016-08-15T12:40:42Z",
      "endDateTime": "", "endPoint1":
      {
        "uniServiceId": "47/KEFN/103458/TWCS",
        "ceVlans": ["22"]
      },
      "endPoint2":
      {
        "uniServiceId": "54/KFFN/102109/TWCS",
        "ceVlans": ["33"]
      },
      "couldConnections": []
    },
  ],
  "receivedParams":
  {
    "billingAccountNumber": "123456",
    "source": "dynamic"
  }
}
```

Example with one cloud connection:

```
{
  "evcs":
  [
    {
      "evcServiceId": "VLXX/D00112/LVLC",
      "evcName": "Level3 test 2",
      "connectionStatus": "active",
      "requestor": "jgwang",
      "requestId": "test11",
      "bandwidth": "100000000",
      "cos": "Basic",
      "detail": "",
      "billingAccountNumber": "123456",
      "startDateTime": "2016-08-15T12:40:42Z",
      "endDateTime": "",
      "endPoint1":
      {
        "uniServiceId": "47/KEFN/103458/TWCS",
        "ceVlans": ["22"]
      },
      "endPoint2":
      {
        "uniServiceId": "54/KFFN/102109/TWCS",
        "ceVlans": ["33"]
      },
      "cloudConnections":
      [
        {
          "cloudProvider": "aws",
          "connectionStatus": "active",
          "awsConnectionProfile":
          {
            "cloudNetworkId": "152420170405181901",
            "status": "complete",
            "type": "createConnection",
            "requestor": "john.doe@level3.com",
            "requestId": "ef2c8a5f-7d97-46cc-b38b-d9e247e30993",
            "awsRegionId": "us-east-1",
            "customerAccount": "123456",
            "customerAwsAccount": "12341234",
            "customerAsn": "1234",
            "awsAsn": "5678",
            "uniServiceId": "47/KEFN/103458/TWCS",
            "ceVlan": "11",
          }
        }
      ]
    }
  ]
}
```



```

"customerBgpPeerAddress": "10.0.0.1/30",
"awsBgpPeerAddress": "10.0.0.2/30",
    "bgpAuthenticationKey": "secret",
    "connectionId": "dxcon-abcd1234",
    "virtualInterfaceId": "dxvif-wxyz5678",
    "virtualInterfaceName":
"Internal_NNI_to_Amazon_Web_Services_152420170405181901",
    "virtualGatewayId": "test1234",
    "arnSecurityToken": "arn:aws:iam::1234:role/xyz",
    "startDateTime": "2016-08-15T12:40:42Z"
    }
}
]
},
"receivedParams":
{
    "billingAccountNumber": "123456",
    "source": "dynamic"
}
}

```

## Failure Response

A response with some failure HTTP status code is returned after the request is rejected. The following data encoded in JSON format is returned in the response body to carry more details.

Name	Type	Value (example)	Description
statusCode	integer	400	The HTTP status code following HTTP standard.
code	integer	400003	A Level 3 generated code combining HTTP code with sub-category code.
message	string	Missing data	Failure info at summarized level.
detail	string	Missing parameter billingAccountNumber	Info in detail about the failure.

Example:

```

{
  "exception":
  {
    "statusCode": "400",
    "code": "400003",
    "message": "Missing data",
    "detail": "Missing parameter billingAccountNumber"
  }
}

```

## Query Requests by Account ID

### Overview

The processing status of any asynchronous request, e.g., “create connection” or “delete connection”, made from a given customer account within a time window can be queried using this API request via the HTTP GET method. A call using this API is synchronous.

### URL

Examples:

<https://api.level3.com/Network/v2/DynamicConnection/requests?billingAccountNumber={billingAccountNumber}>

<https://api.level3.com/Network/v2/DynamicConnection/requests/{requestId}?billingAccountNumber={billingAccountNumber}>

<https://api.level3.com/Network/v2/DynamicConnection/requests?billingAccountNumber={billingAccountNumber}&evcServiceId={evcServiceId}>

### Input Parameters in HTTP GET URL

The input parameters are encoded in URL and sent using HTTP GET.

Name	Type	Requirement	Value (example)	Description
billingAccountNumber	string	mandatory	123456	The unique ID of a given customer account.
requestId	string	optional	878bd9be-87fc-4980-bc24-ec182730d2eb	The globally unique request ID returned by Level 3 platform to identify each request.
evcServiceId	string	optional	VLXX/D00001/LV LC	The unique ID identifying an EVC in Level 3 inventory.
startTime	string	optional	2016-08-01T00:00:00Z	The start time of query window specified in ISO8601 format in UTC (YYYY-MM-DDTHH24:MM:SSZ). If it's omitted, the start time will be defaulted to “2016-01-01T00:00:00Z”.
endTime	string	optional	2016-09-01T00:00:00Z	The end time of query window specified in ISO8601 format in UTC (YYYY-MM-DDTHH24:MM:SSZ). If it's omitted, the end time will be defaulted to current time.

Example 1:

```
.../requests?billingAccountNumber=123456
```

Example 2:

```
.../requests?billingAccountNumber=123456&startTime=2016-08-01T00%3A00%3A00Z&endTime=2016-09-01T00%3A00%3A00Z
```

## Response

A HTTP response is returned after the request is accepted and processed by the dynamic controller.

### Successful Response

A response with status code of "200" is returned after the request is accepted and successfully processed. The following data encoded in JSON format is returned in the response body.

Name	Type	Value (example)	Description
evcServiceId	string	VLXX/D00001/LVLC	The unique ID identifying an EVC, managed by Level 3 platform.
evcServiceName	string	Level3 test 1	The name to describe the connection by API user. The name is opaque to Level 3 system.
type	string	createConnection	The type of a request. Valid values: createConnection, deleteConnection, addCeVlan, deleteCeVlan
status	string	complete	The status of a request. The status will be one of the following enumerated strings: "wip" (work in progress), "complete", or "error".
requestor	string	john.doe@level3.com	The identifier of a request initiator. It should be a valid email address.
requestId	string	878bd9be-87fc-4980-bc24-ec182730d2eb	The globally unique request ID returned by Level 3 platform to identify each request.
bandwidth	integer	100000000	The requested bandwidth value in bits per second received in "createConnection" call.
cos	string	Basic	The requested "Class of Service" on the connection received in "createConnection" call.
detail	string	Something is wrong.	Details on failures when status is "error". It may carry additional info or be empty for other statuses.
billingAccountNumber	string	123456	The customer billing account number, which is used for billing, reporting, and certain validations.
createDateTime	string	2016-08-15T12:40:42Z	The timestamp when the request is received. The timestamp is encoded in ISO8601 format in UTC
modifyDateTime	string	2016-08-15T12:40:48Z	The timestamp when last status is updated. The timestamp is encoded in ISO8601 format in UTC
endPoint1	JSON object	{...}	A JSON object represents an endpoint of the EVC. The parameters contained in the object are described in table below.
endPoint2	JSON object	{...}	A JSON object represents an endpoint of the EVC. The parameters contained in the object are described in table below.
cloudNetwork1	Optional JSON object	{...}	An optional JSON object represents an EVC segment between endPoint1 at Level 3 network and a cloud provider. The parameters contained in the object are described in table below.
cloudNetwork2	Optional JSON object	{...}	An optional JSON object represents an EVC segment between endPoint2 at Level 3 network and a cloud provider. The parameters contained in the object are described in table below.

The parameters within each “endpoint” JSON object:

Name	Type	Value (example)	Description
uniServiceId	string	47/KEFN/103458/TWCS	The unique identifier of UNI at the end point, which is assigned and managed by Level 3 systems.
ceVlans	JSON array	["11"]	A list of customer VLAN IDs which identify customer Ethernet frames on the connection.

The parameters within an optional “cloudNetwork” JSON object:

Name	Type	Value (example)	Description
cloudProvider	string	aws	The unique name identifying a cloud service provider. Valid values (at 2017-09-01): aws
cloudConnectionProfile	JSON object	{...}	The only cloud connection profile supported (as of 2017-09-01) is “awsConnectionProfile”.

Example 1 with no EVC segment to AWS:

```
{
  "requests":
  [
    {
      "evcServiceId": "VLXX/D00001/LVLC",
      "evcName": "Level3 test 1",
      "type": "createConnection",
      "status": "complete",
      "requestor": "john.doe@level3.com",
      "requestId": "878bd9be-87fc-4980-bc24-ec182730d2eb",
      "bandwidth": "100000000",
      "cos": "Basic",
      "detail": "",
      "billingAccountNumber": "1234567",
      "createDateTime": "2016-08-15T12:40:42Z",
      "modifyDateTime": "2016-08-15T12:40:48Z",
      "endPoint1":
      {
        "ceVlans": ["11"],
        "uniServiceId": "47/KEFN/103458/TWCS"
      },
      "endPoint2":
      {
        "ceVlans": ["11"],
        "uniServiceId": "54/KFFN/102109/TWCS"
      }
    }
  ],
  "receivedParams":
  {
    "billingAccountNumber": "123456",
    "evcServiceId": " VLXX/D00001/LVLC"
  }
}
```

Example 2 with EVC segment to AWS:

```
{
  "requests":
  [
    {
      "evcServiceId": "VLXX/D00002/LVLC",
      "evcName": "Level3 test 2",
      "type": "createConnection",
      "status": "complete",
      "requestor": "john.doe@level3.com",
      "requestId": "ff8bd9be-87fc-4980-bc24-ec182730d2eb",
      "bandwidth": "100000000",
      "cos": "Basic",
      "detail": "",
      "billingAccountNumber": "123456",
      "createDateTime": "2016-08-15T12:40:42Z",
      "modifyDateTime": "2016-08-15T12:40:48Z",
      "endPoint1":
      {
        "ceVlans": ["11"],
        "uniServiceId": "03/KFFN/103172/TWCS"
      },
      "endPoint2":
      {
        "ceVlans": [""],
        "uniServiceId": "74/KGFN/110160/TWCS"
      },
      "cloudNetwork2":
      {
        "cloudProvider": "aws",
        "cloudConnectionProfile":
        {
          "cloudNetworkId": "VLXX/D00002/LVLC",
          "status": "complete",
          "detail": ""
        }
      }
    },
    {
      "type": "createConnection",
      "requestor": "john.doe@level3.com",
      "requestId": "ff8bd9be-87fc-4980-bc24-ec182730d2eb",
      "awsRegionId": "us-east-1",
      "customerAccount": "123456",
      "customerAwsAccount": "12341234",
      "customerAsn": "1234",
      "awsAsn": "5678",
      "uniServiceId": "74/KGFN/110160/TWCS",
      "ceVlan": "3000",
    }
  ]
}
```

```

"customerBgpPeerAddress": "10.0.0.1/30",
"awsBgpPeerAddress": "10.0.0.2/30",
    "bgpAuthenticationKey": "secret",
    "connectionId": "dxcon-abcd1234",
    "virtualInterfaceId": "dxvif-wxyz5678",
    "virtualInterfaceName":
"Internal_NNI_to_Amazon_Web_Services_VLXX/D00002/LVLC",
    "virtualGatewayId": "test1234",
    "arnSecurityToken": "arn:aws:iam::1234:role/xyz",
    "createDateTime": "2016-08-15T12:40:42Z",
    "modifyDateTime": "2016-08-15T12:40:42Z"
    }
}
],
"receivedParams":
{
    "billingAccountNumber": "123456",
    "evcServiceId": " VLXX/D00002/LVLC"
}
}

```

## Failure Response

A response with some failure HTTP status code is returned after the request is rejected. The following data encoded in JSON format is returned in the response body to carry more details.

Name	Type	Value (example)	Description
httpStatusCode	integer	400	The HTTP status code following HTTP standard.
code	integer	400003	A Level 3-generated code combining HTTP code with sub-category code.
message	string	Missing data	Failure info at summarized level.
detail	string	Missing parameter billingAccountNumber	Info in detail about the failure.

Example:

```

{
  "exception":
  {
    "httpStatusCode": "400",
    "code": "400003",
    "message": "Missing data",
    "detail": "Missing parameter billingAccountNumber"
  }
}

```

## Add CE-VLAN to EVC

### Overview

A new CE-VLAN can be dynamically added to an existing layer 2 Ethernet Virtual Connection (EVC) between two network endpoints from customer portal or application by calling this API request via the HTTP POST method. A network endpoint is a logical concept that can be either a multiplexed User Network Interface (UNI) or a multiplexed eLink interface. It can also include a layer 2 connection to AWS cloud. A call using this API is asynchronous. The caller needs to check the processing status and act accordingly.

### URL

Example:

<https://api.level3.com/Network/v2/DynamicConnection/ceVlans>

### Input Parameters in HTTP POST Body

The input parameters are encoded in JSON format and sent in HTTP POST request body.

Name	Type	Requirement	Value (example)	Description
userEmail	string	mandatory	john.doe@level3.com	The email address of a user. This address identifies the user. The address is also used for email notifications when the processing is completed successfully. A failure notification will be sent when processing encounters error.
billingAccountNumber	string	mandatory	123456	The customer billing account number, which is used for billing, reporting, and certain validations.
ceVlan	string	mandatory	11	A customer VLAN ID, which identifies customer Ethernet traffic on the EVC.
evcServiceId	string	mandatory	VLXX/D00001/LVLC	The EVC service ID to which a CE-VLAN is to be added.

Example:

```
{
  "userEmail": "john.doe@level3.com",
  "billingAccountNumber": "123456",
  "ceVlan": "11",
  "evcServiceId": "VLXX/D00001/LVLC",
}
```

### Response

A HTTP response is returned after the request is accepted while it is being processed asynchronously by the dynamic controller. The status of the request can be checked API in 2.5 or 0.

#### Successful Response

A response with status code of "202" is returned after the request is accepted and queued to be processed. The following data encoded in JSON is returned in the response body.

Name	Type	Value (example)	Description
requestId	string	a41d00cd-be8f-49ac-a4c2-9082cca28981	A unique ID (GUID) generated by Level 3 dynamic controller to identify the request. This ID can be used to query the request status using API in 2.5 or 0.
evcServiceId	string	VLXX/D00001/LVLC	A unique ID that identifies the EVC.
action	string	addCeVlan	The action of the request. Valid values: "addCeVlan".

status	string	wip	The status of the request. The status will be one of the following enumerated strings: "wip" (work in progress), "complete", or "error".
receivedParams	JSON object	{...}	The echo of the request parameters in JSON.

Example:

```
{
  "requestId": "a41d00cd-be8f-49ac-a4c2-9082cca28981",
  "evcServiceId": "VLXX/D00001/LVLC",
  "action": "addCeVlan",
  "status": "wip",
  "receivedParams":
  {
    "userEmail": "john.doe@level3.com",
    "billingAccountNumber": "123456",
    "ceVlan": "11",
    "evcServiceId": "VLXX/D00001/LVLC"
  }
}
```

## Failure Response

A response with some failure HTTP status code is returned after the request is rejected. The following data encoded in JSON is returned in the response body to carry more details.

Name	Type	Value (example)	Description
statusCode	integer	503	The HTTP status code following HTTP standard.
code	integer	503001	A Level 3 generated code combining HTTP code with sub-category code.
message	string	Service Unavailable	Failure info at summarized level.
detail	string	Missing parameter billingAccountNumber	Info in detail about the failure.

Example:

```
{
  "exception":
  {
    "statusCode": "503",
    "code": "503001",
    "message": "Service Unavailable",
    "detail": "Missing parameter billingAccountNumber"
  }
}
```



## Remove CE-VLAN from EVC

### Overview

A CE-VLAN can be dynamically removed from an existing layer 2 Ethernet Virtual Connection (EVC) between two network endpoints from customer portal or application by calling this API request via the HTTP DELETE method. A call using this API is asynchronous. The caller needs to check the processing status and act accordingly using API in 2.5 or 0.

### URL

Example:

<https://api.level3.com/Network/v2/DynamicConnection/ceVlans>

### Input Parameters in HTTP DELETE Body

The input parameters are encoded in JSON format and sent in HTTP DELETE request body.

Name	Type	Requirement	Value (example)	Description
userEmail	string	mandatory	John.doe@level3.com	The email address of a user. This address identifies the user. The address is also used for email notifications when the processing is completed successfully. A failure notification will be sent when processing encounters error.
billingAccount Number	string	mandatory	123456	The customer billing account number, which is used for billing, reporting, and certain validations.
evcServiceId	string	mandatory	VLXX/D00001/LVLC	The unique ID for identifying the EVC from which a CE-VLAN is to be removed.
ceVlan	string	mandatory	11	A customer VLAN ID, which identifies customer Ethernet traffic on the EVC.

Example:

```
{
  "userEmail": "john.doe@level3.com",
  "billingAccountNumber": "123456",
  "evcServiceId": "VLXX/D00001/LVLC",
  "ceVlan": "11"
}
```

### Response

A HTTP response is returned after the request is accepted while it is being processed asynchronously by dynamic controller. The status of the request can be checked using request status query API in 2.5 or 0.

#### Successful Response

A response with status code of "202" is returned after the request is accepted and queued to be processed. The following data encoded in JSON format is returned in the response body.

Name	Type	Value (example)	Description
requestId	string	a41d00cd-be8f-49ac-a4c2-9082cca28981	A unique ID (GUID) generated by Level 3 dynamic controller to identify the request. This ID can be used to query the request status using API in 2.5 or 0.
evcServiceId	string	VLXX/D00001/LVLC	A unique ID that identifies the EVC.

action	string	removeCeVlan	The action of the request. Valid values: "removeCeVlan".
status	string	wip	The status of the request. The status will be one of the following enumerated strings: "wip" (work in progress), "complete", or "error".
receivedParams	JSON object	{...}	The echo of the request parameters.

Example:

```
{
  "requestId": "a41d00cd-be8f-49ac-a4c2-9082cca28981",
  "evcServiceId": "VLXX/D00001/LVLC",
  "action": "removeCeVlan",
  "status": "wip",
  "receivedParams":
  {
    "userEmail": "john.doe@level3.com",
    "billingAccountNumber": "123456",
    "evcServiceId": "VLXX/D00001/LVLC",
    "ceVlan": "11",
  }
}
```

## Failure Response

A response with some failure HTTP status code is returned after the request is rejected. The following data encoded in JSON is returned in the response body to carry more details.

Name	Type	Value (example)	Description
statusCode	integer	503	The HTTP status code following HTTP standard.
code	integer	503001	A Level 3 generated code combining HTTP code with sub-category code.
message	string	Service Unavailable	Failure info at summarized level.
detail	string	Missing parameter billingAccountNumber	Info in detail about the failure.

Example:

```
{
  "exception":
  {
    "statusCode": "503",
    "code": "503001",
    "message": "Service Unavailable",
    "detail": "Missing parameter billingAccountNumber"
  }
}
```

## Query CE-VLANs

### Overview

A user can use this API request via the HTTP GET method to query for CE-VLAN info. A call using this API is synchronous.

### URL

Example:

<https://api.level3.com/Network/v2/DynamicConnection/ceVlans?billingAccountNumber={billingAccountId}&source={sourceId}>

### Input Parameters in HTTP GET URL

The input parameters are described below.

Name	Type	Requirement	Value (example)	Description
billingAccountNumber	string	mandatory	123456	The unique ID identifying a customer account in Level 3 inventory system.
source	string	mandatory	aws	The unique name identifying the data source for CE-VLANs. Valid values: aws, or level3
uniServiceId	string	optional	47/KEFN/103458/TWCS	The unique identifier of UNI, which is assigned and managed by Level 3 systems. This parameter is mandatory when domain is "aws". It identifies the UNI that has connections to AWS cloud. It is ignored when domain is not "aws".
evcServiceId	string	optional	VLXX/123456/LVLC	The unique ID identifying an EVC in Level 3 inventory system. This parameter is mandatory when domain is "level3". It identifies the EVC that is associated with the CE-VLAN info returned. It is ignored when domain is not "level3".

Example 1:

```
.../ceVlans?billingAccountNumber=123456&source=aws&uniServiceId=47/KEFN/103458/TWCS
```

Example 2:

```
.../ceVlans?billingAccountNumber=123456&source=level3&evcServiceId=VLXX/123456/LVLC
```

### Response

A HTTP response is returned after the request is accepted and processed by the dynamic controller.

#### Successful Response

A response with status code of "200" is returned after a request is accepted and successfully processed. The following data encoded in JSON format is returned in the response body.

Name	Type	Value (example)	Description
receivedParams	JSON object	{...}	The echo of request parameters in JSON.
ceVlans	JSON object	{...}	The details of CE-VLANs.

The ceVlans JSON object has the following attributes:

Name	Type	Value (example)	Description
ceVlansInUse	JSON	[...]	A list of CE-VLANs in use (not available). Each item in the list

	array		represents a CE-VLAN.
ceVlansAvailable	JSON array	[...]	A list of CE-VLANs available from Level 3 perspective. Each item represents a range of CE-VLANs available. The format is “begin-end”, e.g., “1-10” meaning CE-VLANs of “1” to “10” (inclusive) are all available. A customer may have further constraints on the CE-VLANs that can be used from customer’s perspective. Therefore, a customer has to choose which CE-VLAN to use.

Example:

```
{
  "receivedParams":
    {
      "billingAccountNumber": "123456",
      "source": "aws",
      "uniServiceId": "47/KEFN/103458/TWCS"
    },
  "ceVlans":
    {
      "ceVlansInUse": ["11", "22"],
      "ceVlansAvailable": ["1-10", "12-21", "23-4095"]
    }
}
```

### Failure Response

A response with some failure HTTP status code is returned after the request is rejected. The following data encoded in JSON format is returned in the response body to carry more details.

Name	Type	Value (example)	Description
statusCode	integer	400	The HTTP status code following HTTP standard.
code	integer	400003	A Level 3-generated code combining HTTP code with sub-category code.
message	string	Missing data	Failure info at summarized level.
detail	string	The source is missing.	Info in detail about the failure.

Example:

```
{
  "exception":
    {
      "statusCode": "400",
      "code": "400003",
      "message": "Missing data",
      "detail": "The source is missing."
    }
}
```

## Create Connection to AWS

### Overview

A new L3 connection to an AWS virtual gateway in an AWS region can be dynamically created. The L2 connection associated with this L3 connection is identified by a CE-VLAN. A customer portal or application will call this API request via the HTTP POST method. A call using this API is asynchronous. The caller needs to check the processing status and act accordingly using API in 2.5 or 2.6.

### URL

Example:

<https://api.level3.com/Network/v2/DynamicConnection/cloudNetworks>

### Input Parameters in HTTP POST Body

The input parameters are encoded in JSON format and sent in HTTP POST request body.

Name	Type	Requirement	Value (example)	Description
userEmail	string	mandatory	john.doe@level3.com	The email address of a user. This address identifies the user. The address is also used for email notifications when the processing is completed successfully. A failure notification will be sent when processing encounters error.
billingAccountNumber	string	mandatory	123456	The customer billing account number, which is used for billing, reporting, and certain validations.
cloudProvider	string	mandatory	aws	The unique name identifying a cloud service provider. Valid values are currently: 1) <u>aws</u> (as of 2017-10-09)
awsConnectionProfile	JSON object	mandatory	{...}	A JSON object that contains connection profile to AWS cloud.

The parameters within awsConnectionProfile JSON object:

Name	Type	Requirement	Value (example)	Description
uniServiceId	string	Mandatory	47/KEFN/103458/TWCS	The unique identifier of UNI at an endpoint of a EVC, which is assigned and managed by Level 3.
customerAwsAccount	string	mandatory	123412341234	The customer's account registered in AWS cloud.
customerAsn	string	mandatory	1234	The customer's ASN for IP routing.
evcServiceId	string	mandatory	VLXX/D00001/LVLC	The unique identifier of an EVC, to which this AWS connection will be mapped to. <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p><b>Note:</b> This parameter may be changed to optional later with a new API request to attach an EVC to a cloud connection.</p> </div>
ceVlan	string	optional	11	A VLAN ID, which identifies Ethernet traffic to AWS cloud in a region. This parameter is optional. It will be used if it is present.

Name	Type	Requirement	Value (example)	Description
				Otherwise, Level 3 system will choose an available value when it's not present or its value is empty.
autoGenerateBgpPeerAddress	string	mandatory	true or false	A flag to indicate whether peer IP addresses will be auto-generated.
customerBgpPeerAddress	string	optional	10.0.0.1/30	The customer's destination IPv4 CIDR AWS should use to send traffic to, which is required only when "autoGenerateBgpPeerAddress" flag is set to "false". It will be auto-generated when "autoGenerateBgpPeerAddress" flag is set to "true".
awsBgpPeerAddress	string	optional	10.0.0.2/30	The AWS's destination IPv4 CIDR the customer should use to send traffic to, which is required only when "autoGenerateBgpPeerAddress" flag is set to "false". It will be auto-generated when "autoGenerateBgpPeerAddress" flag is set to "true".
autoGenerateBgpAuthenticationKey	string	mandatory	true or false	A flag to indicate whether a BGP authentication key will be auto-generated.
bgpAuthenticationKey	string	optional	secret	The password (authentication key) that will be used to authenticate BGP session, which is required only when "autoGenerateBgpAuthenticationKey" flag is set to "false". It will be auto-generated when "autoGenerateBgpAuthenticationKey" flag is set to "true".
autoAcceptConnection	string	mandatory	true or false	The flag to indicate whether auto-accepting a connection on customer's behalf is allowed.
virtualGatewayId	string	optional	vgw-test123	The identifier of the virtual private gateway that will be attached to the virtual interface being created. It is only required when the "autoAcceptConnection" flag is set to "true". It is ignored when the "autoAcceptConnection" flag is set to "false".
arnSecurityToken	string	optional	arn:aws:iam::1234:role/xyz	The ARN security role which will be used to auto-accept ownership of the virtual interface being created. It is only required when the "autoAcceptConnection" flag is set to "true". It is ignored when the "autoAcceptConnection" flag is set to "false".

Example:

```

{
  "userEmail": "john.doe@level3.com",
  "billingAccountNumber": "123456",
  "cloudProvider": "aws",
  "awsConnectionProfile":
  {
    "uniServiceId": "47/KEFN/103458/TWCS",
  }
}
    
```

```

"customerAwsAccount": "123412341234",
"customerAsn": "1234",
"evcServiceId": "VLXX/D00001/LVLC",
"ceVlan": "11",
"autoGenerateBgpPeerAddress": "false",
"customerBgpPeerAddress": "10.0.0.1/30",
"awsBgpPeerAddress": "10.0.0.2/30",
"autoGenerateBgpAuthenticationKey": "false",
"bgpAuthenticationKey": "secret",
"autoAcceptConnection": "true",
"virtualGatewayId": "vgw-test567",
"arnSecurityToken": "arn:aws:iam::1234:role/xyz"
}
}

```

## Response

A HTTP response is returned after the request is accepted while it is being processed asynchronously by the dynamic controller. The processing status of the request can be checked using API in 2.5 or 2.6.

### Successful Response

A response with status code of “202” is returned after the request is accepted and queued to be processed. The following data encoded in JSON is returned in the response body.

Name	Type	Value (example)	Description
requestId	string	a41d00cd-be8f-49ac-a4c2-9082cca28981	A unique ID (GUID) generated by Level 3 dynamic controller to identify the request. This ID can be used to query the request status using API in 2.5 or 0.
cloudNetworkId	string	152420170405181901	A unique ID that identifies the EVC between Level 3 eLynk interface and AWS virtual gateway.
action	string	createConnectionAws	The action of the request. Valid values: “createConnectionAws”.
status	string	wip	The status of the request. The status will be one of the following enumerated strings: “wip” (work in progress), “complete”, or “error”.
receivedParams	JSON object	{...}	The echo of the request parameters in JSON.

Example:

```

{
  "requestId": "a41d00cd-be8f-49ac-a4c2-9082cca28981",
  "cloudNetworkId": "152420170405181901",
  "action": "createConnectionAws",
  "status": "wip",
  "receivedParams":
  {
    "userEmail": "john.doe@level3.com",
    "billingAccountNumber": "123456",
    "cloudProvider": "aws",
    "awsConnectionProfile":

```

```
{
  "uniServiceId": "47/KEFN/103458/TWCS",
  "awsRegionId": "us-east-1",
  "customerAwsAccount": "123412341234",
  "customerAsn": "1234",
  "evcServiceId": "VLXX/D00001/LVLC",
  "ceVlan": "11",
  "autoGenerateBgpPeerAddress": "false",
  "customerBgpPeerAddress": "10.0.0.1/30",
  "awsBgpPeerAddress": "10.0.0.2/30",
  "autoGenerateBgpAuthenticationKey": "false",
  "bgpAuthenticationKey": "secret",
  "autoAcceptConnection": "true",
  "virtualGatewayId": "vgw-test567",
  "arnSecurityToken": "arn:aws:iam::1234:role/xyz"
}
```

## Failure Response

A response with some failure HTTP status code is returned after the request is rejected. The following data encoded in JSON is returned in the response body to carry more details.

Name	Type	Value (example)	Description
httpStatusCode	integer	503	The HTTP status code following HTTP standard.
code	integer	503001	A Level 3 generated code combining HTTP code with sub-category code.
message	string	Service Unavailable	Failure info at summarized level.
detail	string	Missing parameter billingAccountNumber	Info in detail about the failure.

Example:

```
{
  "exception":
  {
    "httpStatusCode": "503",
    "code": "503001",
    "message": "Service Unavailable",
    "detail": "Missing parameter billingAccountNumber"
  }
}
```

## Delete Connection to AWS

### Overview

A layer 2 Ethernet Virtual Connection from a eLink interface to an AWS virtual gateway can be dynamically deleted from customer portal or application by calling this API request via the HTTP DELETE method. A call using this API is asynchronous. The caller needs to check the processing status and act accordingly using the key "requestId" returned in the response.



## URL

Example:

<https://api.level3.com/Network/v2/DynamicConnection/cloudNetworks>

## Input Parameters in HTTP DELETE Body

The input parameters are encoded in JSON format and sent in HTTP DELETE request body.

Name	Type	Requirement	Value (example)	Description
userEmail	string	mandatory	john.doe@level3.com	The email address of a user. This address identifies the user. The address is also used for email notifications when the processing is completed successfully. A failure notification will be sent when processing encounters error.
billingAccountNumber	string	mandatory	123456	The customer billing account number, which is used for billing, reporting, and certain validations.
cloudProvider	string	mandatory	aws	The unique name identifying a cloud service provider. Valid values are currently: 1) aws (as of 2017-10-09).
cloudNetworkId	string	mandatory	152420170405181901	The unique ID returned by Level 3 platform for identifying the connection after a successful "create connection to AWS" call.

Example:

```
{
  "userEmail": "john.doe@level3.com",
  "billingAccountNumber": "123456",
  "cloudProvider": "aws",
  "cloudNetworkId": "152420170405181901"
}
```

## Response

A HTTP response is returned after the request is accepted while it is being processed asynchronously by dynamic controller. The processing status of the request can be checked using API in 2.5 or 2.6.

### Successful Response

A response with status code of "202" is returned after the request is accepted and queued to be processed. The following data encoded in JSON format is returned in the response body.

Name	Type	Value (example)	Description
requestId	string	a41d00cd-be8f-49ac-a4c2-9082cca28981	A unique ID (GUID) generated by Level 3 dynamic controller to identify the request. This ID can be used to query the request status using API in 2.5 or 2.6.
cloudNetworkId	string	152420170405181901	The unique ID identifying the cloud connection in Level 3 inventory.
action	string	deleteConnectionAws	The action of the request. Valid values: "deleteConnectionAws".
status	string	wip	The status of the request. The status will be one of the following enumerated strings: "wip" (work in progress), "complete", or "error".

Name	Type	Value (example)	Description
receivedParams	JSON object	{...}	The echo of the request parameters.

Example:

```
{
  "requestId": "a41d00cd-be8f-49ac-a4c2-9082cca28981",
  "cloudNetworkId": "152420170405181901",
  "action": "deleteConnectionAws",
  "status": "wip",
  "receivedParams":
  {
    "userEmail": "john.doe@level3.com",
    "billingAccountNumber": "123456",
    "cloudProvider": "aws",
    "cloudNetworkId": "152420170405181901"
  }
}
```

## Failure Response

A response with some failure HTTP status code is returned after the request is rejected. The following data encoded in JSON is returned in the response body to carry more details.

Name	Type	Value (example)	Description
statusCode	integer	503	The HTTP status code following HTTP standard.
code	integer	503001	A Level 3 generated code combining HTTP code with sub-category code.
message	string	Service Unavailable	Failure info at summarized level.
detail	string	Missing parameter cloudNetworkId	Info in detail about the failure.

Example:

```
{
  "exception":
  {
    "statusCode": "503",
    "code": "503001",
    "message": "Service Unavailable",
    "detail": "Missing parameter cloudNetworkId"
  }
}
```

## Query Connections to AWS by Account

### Overview

The status of connection instances to AWS for a given customer billing account can be queried using this API request via the HTTP GET method. A call using this API is synchronous.

## URL

Example:

<https://api.level3.com/Network/v2/DynamicConnection/cloudNetworks?cloudProvider=aws&billingAccountNumber={billingAccountNumber}>

## Input Parameters in HTTP GET URL

The input parameters are encoded in URL and sent using HTTP GET.

Name	Type	Requirement	Value (example)	Description
cloudProvider	string	mandatory	aws	The unique name identifying a cloud service provider. Valid values (at 2017-08-31): aws
billingAccountNumber	string	mandatory	123456	The customer billing account number, which is used for billing, reporting, and certain validations.
uniServiceId	string	optional	47/KEFN/103458/TWCS	The unique identifier of UNI or Elynk interface at one endpoint, which is assigned and managed by Level 3 systems. This parameter is used as a filter: (a) return only records matching the "uniServiceId" when it is present; (b) return all records of any "uniServiceId" when it is absent.
ceVlan	string	optional	11	A CE-VLAN, which identifies Ethernet traffic to AWS cloud in a region. This parameter is optional and used as a filter: (a) return only records matching the "ceVlan" when it is present; (b) return all records of any "ceVlan" when it is absent.
cloudNetworkId	string	optional	152420170405181901	The unique ID returned by Level 3 platform to identify the cloud connection instance after a successful "create connection to AWS" call. This parameter is used as a filter: (a) return only records matching the "cloudNetworkId" when it is present; (b) return all records of any "cloudNetworkId" when it is absent.
connectionStatus	string	optional	active	This parameter is used as a filter. Valid values are: "active", "disconnected", "wip_creating", "wip_deleting", "error_creating", or "error_deleting". The "active" filter only returns connections in "active" status. The "disconnected" filter only returns connections in "disconnected" status, etc. It returns records in all status when the parameter is absent or empty.

Example 1:

```
.../cloudNetworks?cloudProvider=aws&billingAccountNumber=123456
```

Example 2 with “unild” filter:

```
.../cloudNetworks?cloudProvider=aws&billingAccountNumber=123456&unild=47/KEFN/103458/TWCS
```

## Response

A HTTP response is returned after the request is accepted and processed by the dynamic controller.

### Successful Response

A response with status code of “200” is returned after the request is accepted and successfully processed. The following data encoded in JSON format is returned in the response body.

Name	Type	Value (example)	Description
cloudProvider	string	aws	The descriptive unique name identifying a cloud provider. Valid values: “aws” (as of 2017-08-31).
connectionStatus	string	active	The status of a connection.
awsConnectionProfile	JSON object	{...}	A JSON object that contains connection profile to AWS cloud.

The parameters within awsConnectionProfile JSON object:

Name	Type	Value (example)	Description
cloudNetworkId	string	152420170405181901	The unique ID returned by Level 3 platform to identify the EVC connection instance to AWS after a successful “create connection to AWS” call.
status	string	complete	The status of the request. The status will be one of the following enumerated strings: “wip” (work in progress), “complete”, or “error”.
type	string	createConnection	Last request type been processed. Valid values: “createConnection”, “deleteConnection”.
requestor	string	john.doe@level3.com	The identifier of a request initiator.
requestId	string	ef2c8a5f-7d97-46cc-b38b-d9e247e30993	The request ID of “create connection to AWS” or “delete connection to AWS” calls.
awsRegionId	string	us-east-1	The unique identifier of an AWS region. Valid values: <a href="http://docs.aws.amazon.com/general/latest/gr/rande.html">http://docs.aws.amazon.com/general/latest/gr/rande.html</a> e.g., us-east-1, us-west-1, us-west-2
customerAccount	string	123456	The customer’s billing account with Level 3.
customerAwsAccount	string	12341234	The customer’s account registered in AWS cloud.
customerAsn	string	1234	The customer’s ASN for IP routing.
awsAsn	string	5678	The AWS’s ASN for IP routing.
uniServiceId	string	47/KEFN/103458/TWCS	The unique identifier of UNI or Elynk interface at one endpoint, which is assigned and managed by Level 3 systems.
ceVlan	string	11	The CE-VLAN of the cloud connection instance.
customerBgpPeerAddress	string	10.0.0.1/30	The customer’s IPv4 CIDR AWS should use to establish

Name	Type	Value (example)	Description
ss			BGP sessions.
awsBgpPeerAddress	string	10.0.0.2/30	The AWS's IPv4 CIDR a customer should use to establish BGP sessions.
bgpAuthenticationKey	string	secret	The password (authentication key) that will be used to authenticate BGP session.
connectionId	string	dxcon-abcd1234	The AWS connection identifier of the AWS connection instance.
virtualInterfaceId	string	dxvif-wxyz5678	The virtual interface identifier of the AWS connection instance.
virtualInterfaceName	string	Internal_NNI_to_Amazon_Web_Services_VLXX/D00001/LVLC	The virtual interface name of the AWS connection instance.
virtualGatewayId	string	vgw-test1234	The identifier of the private virtual gateway of the AWS connection instance.
arnSecurityToken	string	arn:aws:iam::1234:role/xyz	The ARN security role which is used to auto-accept ownership of the virtual interface.
startDateTime	string	2016-08-15T12:40:42Z	The timestamp when a connection (in "active" state) was ready for service ("started"). The timestamp is encoded in ISO8601 format. It may be empty if the connection was created before the query window.
endDateTime	string	2016-08-30T12:40:42Z	The timestamp when a connection (in "disconnected" state) was deleted successfully ("end" its service life). The timestamp is encoded in ISO8601 format. It should be empty when the connection is in "active" status.

Example:

```
{
  "cloudConnections":
  [
    {
      "cloudProvider": "aws",
      "connectionStatus": "active",
      "awsConnectionProfile":
      {
        "cloudNetworkId": "152420170405181901",
        "status": "complete",
        "requestor": "john.doe@level3.com",
        "requestId": "ef2c8a5f-7d97-46cc-b38b-d9e247e30993",
        "awsRegionId": "us-east-1",
        "customerAccount": "123456",
        "customerAwsAccount": "12341234",
        "customerAsn": "1234",
        "awsAsn": "5678",
        "uniServiceId": "47/KEFN/103458/TWCS",
      }
    }
  ]
}
```

```

        "ceVlan": "11",
"customerBgpPeerAddress": "10.0.0.1/30",
"awsBgpPeerAddress": "10.0.0.2/30",
        "bgpAuthenticationKey": "secret",
        "connectionId": "dxcon-abcd1234",
        "virtualInterfaceId": "dxvif-wxyz5678",
        "virtualInterfaceName":
"Internal_NNI_to_Amazon_Web_Services_152420170405181901",
        "virtualGatewayId": "test1234",
        "arnSecurityToken": "arn:aws:iam::1234:role/xyz",
        "startDateTime": "2016-08-15T12:40:42Z"
    }
},
"receivedParams":
{
    "cloudProvider": "aws",
    "billingAccountNumber": "123456"
}
}

```

## Failure Response

A response with some failure HTTP status code is returned after the request is rejected. The following data encoded in JSON format is returned in the response body to carry more details.

Name	Type	Value (example)	Description
statusCode	integer	400	The HTTP status code following HTTP standard.
code	integer	400003	A Level 3-generated code combining HTTP code with sub-category code.
message	string	Missing data	Failure info at summarized level.
detail	string	Missing parameter billingAccountNumber	Info in detail about the failure.

Example:

```

{
  "exception":
  {
    "statusCode": "400",
    "code": "400003",
    "message": "Missing data",
    "detail": "Missing parameter billingAccountNumber"
  }
}

```

## Query EVCs by Account

### Overview

A user can use this API request via the HTTP GET method to query for available existing EVCs for an account. Each EVC is identified by a unique ID and contains a list of endpoints. A call using this API is synchronous.

### URL

Examples:

<https://api.level3.com/Network/v2/DynamicConnection/evcs?billingAccountNumber={billingAccountNumber}&source=provisioned>

<https://api.level3.com/Network/v2/DynamicConnection/evcs/{evcServiceId}?billAccountNumber={billingAccountNumber}&source=provisioned>

### Input Parameters in HTTP GET URL

The input parameters are described below.

Name	Type	Requirement	Value (example)	Description
billingAccountNumber	string	mandatory	123456	The unique ID identifying a customer account in Level 3 inventory system.
source	string	mandatory	provisioned	The source of EVCs: provisioned or dynamically created. Valid value: provisioned
evcServiceId	string	optional	VLXX/123456/LVLC	The unique ID identifying an EVC in Level 3 inventory system. This optional parameter will filter out the returned list to this single EVC if it is present. All EVCs will be returned if it is absent.

Example:

```
.../evcs?billingAccountNumber=123456&source=provisioned
.../evcs/VLXX%2F123456%2FLVLC?billingAccountNumber=123456&source=provisioned
```

### Response

A HTTP response is returned after the request is accepted and processed by the dynamic controller.

#### Successful Response

A response with status code of "200" is returned after a request is accepted and successfully processed. The following data encoded in JSON format is returned in the response body.

Name	Type	Value (example)	Description
receivedParams	JSON object	{"billingAccountNumber": "123456"}	The echo of the request parameters in JSON.
evcs	JSON list	[ {...}, {...} ]	The list of available EVCs for the given account.

Each JSON object in the returned list has the following attributes:

Name	Type	Value (example)	Description
evc	JSON object	{...}	The EVC data object.
cloudNetworks	JSON list	[...]	A list of cloud network connection data objects. This object is optional. It is only present when a cloud network connection has been created for a CE-VLAN tagged traffic within the EVC.

The evc JSON object has the following attributes:

Name	Type	Value (example)	Description
evcId	string	1564031	A unique ID that identifies the EVC.
evcServiceId	string	VLXX/123456/LVLC	Another internal unique ID that identifies the EVC.
bandwidth	integer	100000000	The bandwidth of the EVC in bits per second (bps).
cos	string	Basic	The “class of service” of the EVC. Valid values: “Basic”, “Enhanced”, “Dedicated”.
ceVlanPreservation	string	yes	The flag to indicate whether CE-VLAN is preserved. Valid values: “yes”, or “no”.
ceVlansInUse	JSON array	[...]	A list of CE-VLANs in use (not available). Each item in the list represents a CE-VLAN ID.
ceVlansAvailable	JSON array	[...]	A list of CE-VLAN IDs available from Level 3 perspective. Each item represents a range of CE-VLAN IDs available. The format is “begin-end”, e.g., “1-10” meaning CE-VLAN IDs of “1” to “10” (inclusive) are all available. A customer may have further constraints on the CE-VLAN IDs that can be used from customer’s perspective. Therefore, a customer has to choose which CE-VLAN ID to use.
endPoints	JSON array	[...]	A JSON array of endPoint object that contains parameters for an EVC endpoint. The parameters contained in each endPoint object are described in table below.

The parameters within each endPoint JSON object:

Name	Type	Value (example)	Description
endPointId	string	1234567890	The unique identifier of an endpoint, which is assigned and managed by Level 3 systems.
uniId	string	56789	The unique identifier of UNI or Elynk interface at the endpoint, which is assigned and managed by Level 3 systems.
uniServiceId	string	47/KEFN/103458/TWC S	Another unique identifier of UNI or Elynk interface at the endpoint, which is assigned and managed by Level 3 systems.
ceVlans	JSON array	[“11”, “22”]	A list of customer VLAN IDs which identify customer Ethernet traffic on the connection.



Example 1:

```
{
  "receivedParams":
    {
      "billingAccountNumber": "123456",
      "source": "provisioned"
    },
  "evcs":
    [
      {
        "evc":
          {
            "evclid": "1564031",
            "evcServiceId": "VLXX/123456/LVLC",
            "bandwidth": 100000000,
            "cos": "Basic",
            "ceVlanPreservation": "yes",
            "ceVlansInUse": ["11", "22", "4094"],
            "ceVlansAvailable": ["1-10", "12-21", "23-4093"],
            "endPoints":
              [
                {
                  "endPointId": "9145996203713611234",
                  "unild": "1234567",
                  "uniServiceId": "47/KEFN/103458/TWCS",
                  "ceVlans": ["11", "22"]
                },
                {
                  "endPointId": "9145996203713615678",
                  "unild": "8910112",
                  "uniServiceId": "54/KFFN/102109/TWCS",
                  "ceVlans": ["11", "22"]
                }
              ]
          }
        ]
      }
    ]
}
```

Example 2 with one cloud network connection:

```
{
  "receivedParams":
    {
      "evcServiceId": "VLXX/123456/LVLC",
      "billingAccountNumber": "123456",
      "source": "provisioned"
    },
  "evcs":
    [
      {
        "evc":
          {
            "evclid": "1564031",
            "evcServiceId": "VLXX/123456/LVLC",
            "bandwidth": 100000000,
            "cos": "Basic",
            "ceVlanPreservation": "yes",
            "ceVlansInUse": ["11", "22", "4094"],
            "ceVlansAvailable": ["1-10", "12-21", "23-4093"],
            "endPoints":
              [
                {
                  "endPointId": "9145996203713611234",
                  "unild": "1234567",
                  "uniServiceId": "47/KEFN/103458/TWCS",
                  "ceVlans": ["11", "22"],
                  "awsRegionId": "us-east-1",
                },
                {
                  "endPointId": "9145996203713615678",
                  "unild": "8910112",
                  "uniServiceId": "54/KFFN/102109/TWCS",
                  "ceVlans": ["11", "22"]
                }
              ]
          },
      ],
  "cloudNetworks":
    [
      {
        "cloudProvider": "aws",
        "connectionStatus": "active",
        "awsConnectionProfile":
          {
            "cloudNetworkId": "152420170405181901",
            "status": "complete",
          },
        "type": "createConnection",
        "requestor": "john.doe@level3.com",
      }
    ]
}
```

```

        "requestId": "ef2c8a5f-7d97-46cc-b38b-d9e247e30993",
        "awsRegionId": "us-east-1",
        "customerAccount": "123456",
        "customerAwsAccount": "12341234",
        "customerAsn": "1234",
        "awsAsn": "5678",
        "unild": "47/KEFN/103458/TWCS",
        "ceVlan": "11",
        "customerBgpPeerAddress": "10.0.0.1/30",
        "awsBgpPeerAddress": "10.0.0.2/30",
        "bgpAuthenticationKey": "secret",
        "connectionId": "dxcon-abcd1234",
        "virtualInterfaceId": "dxvif-wxyz5678",
        "virtualInterfaceName":
        "Internal_NNI_to_Amazon_Web_Services_152420170405181901",
        "virtualGatewayId": "test1234",
        "arnSecurityToken": "arn:aws:iam::1234:role/xyz",
        "startDateTime": "2016-08-15T12:40:42Z"
    }
}
]
}
}

```

## Failure Response

A response with some failure HTTP status code is returned after the request is rejected. The following data encoded in JSON format is returned in the response body to carry more details.

Name	Type	Value (example)	Description
httpStatusCode	integer	400	The HTTP status code following HTTP standard.
code	integer	400003	A Level 3 generated code combining HTTP code with sub-category code.
message	string	Missing data	Failure info at summarized level.
detail	string	The billingAccountNumber is missing.	Info in detail about the failure.

## Example:

```

{
  "exception":
  {
    "httpStatusCode": "400",
    "code": "400003",
    "message": "Missing data",
    "detail": "The billingAccountNumber is missing."
  }
}

```

## Query End Points by Account

### Overview

A user can use this API request via the HTTP GET method to query for available endpoints (at UNIs) for an account. Each multiplexed UNI or eLink interface is identified by a unique ID. A call using this API is synchronous.

### URL

Example:

<https://api.level3.com/Network/v2/DynamicConnection/endPoints?billingAccountNumber={billingAccountNumber}>

### Input Parameters in HTTP GET URL

The input parameters are encoded in URL and sent in HTTP GET.

Name	Type	Requirement	Value (example)	Description
billingAccountNumber	string	mandatory	123456	The unique ID identifying a customer account in Level 3 inventory system.

Example:

`.../endPoints?billingAccountNumber=123456`

### Response

A HTTP response is returned after the request is accepted and processed by the dynamic controller.

#### Successful Response

A response with status code of "200" is returned after a request is accepted and successfully processed. The following data encoded in JSON format is returned in the response body.

Name	Type	Value (example)	Description
receivedParams	JSON object	{"billingAccountNumber": "123456"}	The echo of the request parameters in JSON.
endPoints	JSON list object	[ {...}, {...} ]	The list of available EVC end points for the given account.

Each JSON object in the returned list has the following attributes:

Name	Type	Value (example)	Description
endPointId	string	9145996203713612310	The identifier of an end point.
evclid	string	1234567	A unique ID that identifies an EVC.
evcServiceId	string	VLXX/123456/LVLC	Another internal unique ID that identifies the EVC.
unild	string	789012	The unique identifier of UNI (or Elynk interface) at the endpoint, which is assigned and managed by Level 3 systems.
uniServiceId	string	59/KGFN/108765/TWCS	Another unique identifier of UNI (or Elynk interface) at the endpoint, which is assigned and managed by Level 3 systems.
ceVlans	JSON array	["11", "22"]	A list of customer VLAN IDs, which identify customer Ethernet traffic on the connection.
awsRegionId	string	us-east-1	The unique identifier of an AWS region. Valid values <a href="http://docs.aws.amazon.com/general/latest/gr/rande.html">http://docs.aws.amazon.com/general/latest/gr/rande.html</a> e.g., us-east-1, us-east-2, us-west-1, us-west-2, eu-central-1. This value is present only when the UNI at the endpoint can be used to connect to AWS cloud.

Example:

```
{
  "receivedParams":
    {
      "billingAccountNumber": "123456"
    },
  "endPoints":
    [
      {
        "endPointId": "9145996203713612310",
        "evclid": "1234567",
        "evcServiceId": "VLXX/123456/TWCS",
        "unild": "789012",
        "uniServiceId": "59/KGFN/108765/TWCS",
        "ceVlans": ["11", "22"]
      }
    ]
}
```

## Failure Response

A response with some failure HTTP status code is returned after the request is rejected. The following data encoded in JSON format is returned in the response body to carry more details.

Name	Type	Value (example)	Description
statusCode	integer	400	The HTTP status code following HTTP standard.
code	integer	400003	A Level 3 generated code combining HTTP code with sub-category code.
message	string	Missing data	Failure info at summarized level.
detail	string	The billingAccountNumber is missing.	Info in detail about the failure.

Example:

```
{
  "exception":
    {
      "statusCode": "400",
      "code": "400003",
      "message": "Missing data",
      "detail": "The billingAccountNumber is missing."
    }
}
```

## Query AWS Requests by Account

### Overview

The “create/delete” connection requests to AWS cloud made from a given customer account within a time window can be queried using this API request via HTTP GET method. A call using this API is synchronous.

### URL

Example:

<https://api.level3.com/Network/v2/DynamicConnection/requests?cloudProvider=aws&billingAccountNumber={billingAccountNumber}>

### Input Parameters in HTTP GET URL

The input parameters are encoded in URL and sent using HTTP GET.

Name	Type	Requirement	Value (example)	Description
cloudProvider	string	mandatory	aws	The unique name identifying a cloud service provider. Valid values (as of 2017-10-09): aws
billingAccountNumber	string	mandatory	123456	The unique ID of a given customer account.
evcServiceId	string	optional	VLXX/123456/LVLC	The unique ID identifying an EVC in Level 3 inventory system. This parameter is used as a filter: (a) return only records mapping to the “evcServiceId” when it is present; (b) return all records mapping to any “evcServiceId” when it is absent.
uniServiceId	string	optional	47/KEFN/103458/TWCS	The unique identifier of UNI or Elynk interface at one endpoint, which is assigned and managed by Level 3 systems. This parameter is used as a filter: (a) return only records matching the “uniServiceId” when it is present; (b) return all records of any “uniServiceId” when it is not present.
ceVlan	string	optional	11	A CE-VLAN, which identifies Ethernet traffic to AWS cloud in a region. This parameter is optional and used as a filter: (a) return only records matching the “ceVlan” when it is present; (b) return all records of any “ceVlan” when it is not present.
cloudNetworkId	string	optional	152420170405181901	The unique ID returned by Level 3 platform to identify the cloud connection instance after a successful “create connection to AWS” call. This parameter is used as a filter: (a) return only records matching the “cloudNetworkId” when it is present; (b) return all records of any “cloudNetworkId” when it is not present.
startTime	string	optional	2016-08-01T00:00:00Z	The start time of query window specified in ISO8601 format (YYYY-MM-DDTHH24:MM:SSZ). If it’s omitted, the start time will be defaulted to “2016-01-01T00:00:00Z”.
endTime	string	optional	2016-09-01T00:00:00Z	The end time of query window specified in ISO8601 format (YYYY-MM-DDTHH24:MM:SSZ). If it’s omitted, the end time will be defaulted to current time.

Example 1:

```
.../cloudNetworks?cloudProvider=aws&billingAccountNumber=123456
```

Example 2;

```
.../cloudNetworks?cloudProvider=aws&billingAccountNumber=123456&evcServiceId=VLXX/123456/LVLC&startTime=2016-08-01T00%3A00%3A00Z&endTime=2016-09-01T00%3A00%3A00Z
```

## Response

A HTTP response is returned after the request is accepted and processed by the dynamic controller.

### Successful Response

A response with status code of “200” is returned after the request is accepted and successfully processed. The following data encoded in JSON format is returned in the response body.

Name	Type	Value (example)	Description
requests	JSON array object	[...]	An array of “awsConnectionProfile” JSON objects
awsConnectionProfile	JSON object	{...}	A JSON object that contains connection profile to AWS cloud.

The parameters within awsConnectionProfile JSON object:

Name	Type	Value (example)	Description
cloudNetworkId	string	152420170405181901	The unique ID returned by Level 3 platform to identify the EVC connection instance to AWS after a successful “create connection to AWS” call.
status	string	complete	The status of the request. The status will be one of the following enumerated strings: “wip” (work in progress), “complete”, or “error”.
type	string	createConnection	Last request type been processed. Valid values: “createConnection”, “deleteConnection”.
requestor	string	john.doe@level3.com	The identifier of a request initiator.
requestId	string	ef2c8a5f-7d97-46cc-b38b-d9e247e30993	The request ID of “create connection to AWS” or “delete connection to AWS” calls.
awsRegionId	string	us-east-1	The unique identifier of an AWS region. Valid values: <a href="http://docs.aws.amazon.com/general/latest/gr/rande.html">http://docs.aws.amazon.com/general/latest/gr/rande.html</a> e.g., us-east-1, us-west-1, us-west-2
customerAccount	string	123456	The customer’s billing account with Level 3.
customerAwsAccount	string	12341234	The customer’s account registered in AWS cloud.
customerAsn	string	1234	The customer’s ASN for IP routing.
awsAsn	string	5678	The AWS’s ASN for IP routing.
uniServiceId	string	47/KEFN/103458/TWC S	The unique identifier of UNI or Elynk interface at one endpoint, which is assigned and managed by Level 3 systems.
ceVlan	string	11	The CE-VLAN of the cloud connection instance.
customerBgpPeerAddress	string	10.0.0.1/30	The customer’s IPv4 CIDR AWS should use to establish BGP sessions.
awsBgpPeerAddress	string	10.0.0.2/30	The AWS’s IPv4 CIDR a customer should use to establish BGP sessions.

Name	Type	Value (example)	Description
bgpAuthenticationKey	string	secret	The password (authentication key) that will be used to authenticate BGP session.
connectionId	string	dxcon-abcd1234	The AWS connection identifier of the AWS connection instance.
virtualInterfaceId	string	dxvif-wxyz5678	The virtual interface identifier of the AWS connection instance.
virtualInterfaceName	string	Internal_NNI_to_Amazon_Web_Services_VLXX/D00001/LVLC	The virtual interface name of the AWS connection instance.
virtualGatewayId	string	vgw-test1234	The identifier of the private virtual gateway of the AWS connection instance.
arnSecurityToken	string	arn:aws:iam::1234:role/xyz	The ARN security role which is used to auto-accept ownership of the virtual interface.
createDateTime	string	2016-08-15T12:40:42Z	The timestamp when a request is received. The timestamp is encoded in ISO8601 format.
modifyDateTime	string	2016-08-15T12:40:42Z	The timestamp when last status is updated. The timestamp is encoded in ISO8601 format.

Example:

```
{
  "requests":
  [
    {
      "awsConnectionProfile":
      {
        "cloudNetworkId": "152420170405181901",
        "status": "complete",
        "type": "createConnection",
        "requestor": "john.doe@level3.com",
        "requestId": "ef2c8a5f-7d97-46cc-b38b-d9e247e30993",
        "awsRegionId": "us-east-1",
        "customerAccount": "123456",
        "customerAwsAccount": "12341234",
        "customerAsn": "1234",
        "awsAsn": "5678",
        "uniServiceId": "47/KEFN/103458/TWCS",
        "ceVlan": "11",
        "customerBgpPeerAddress": "10.0.0.1/30",
        "awsBgpPeerAddress": "10.0.0.2/30",
        "bgpAuthenticationKey": "secret",
        "connectionId": "dxcon-abcd1234",
        "virtualInterfaceId": "dxvif-wxyz5678",
        "virtualInterfaceName":
        "Internal_NNI_to_Amazon_Web_Services_152420170405181901",
        "virtualGatewayId": "test1234",
```



```

        "arnSecurityToken": "arn:aws:iam::1234:role/xyz",
        "createDateTime": "2016-08-15T12:40:42Z",
        "modifyDateTime": "2016-08-15T12:40:42Z"
    }
}
],
"receivedParams":
{
    "cloudProvider": "aws",
    "billingAccountNumber": "123456"
}
}

```

### Failure Response

A response with some failure HTTP status code is returned after the request is rejected. The following data encoded in JSON format is returned in the response body to carry more details.

Name	Type	Value (example)	Description
httpStatusCode	integer	400	The HTTP status code following HTTP standard.
code	integer	400003	A Level 3 generated code combining HTTP code with sub-category code.
message	string	Missing data	Failure info at summarized level.
detail	string	Missing parameter billingAccountNumber	Info in detail about the failure.

Example:

```

{
  "exception":
  {
    "httpStatusCode": "400",
    "code": "400003",
    "message": "Missing data",
    "detail": "Missing parameter billingAccountNumber"
  }
}

```

## Query Locations by Location ID

### Overview

A user can use this API request via HTTP GET method to query for geo location info. Each location is identified by a unique location ID. A call using this API is synchronous.

### URL

Example:

<https://api.level3.com/Network/v2/DynamicConnection/locations/{locationId}>

### Input Parameters in HTTP GET URL

The input parameters are encoded in URL and sent in HTTP GET.

Name	Type	Requirement	Value (example)	Description
locationId	string	mandatory	LSVMNVVT	The unique ID identifying a geo location in Level 3 Global Location Management system.

Example:

.../LSVMNVVT

### Response

A HTTP response is returned after the request is accepted and processed by the dynamic controller.

#### Successful Response

A response with status code of “200” is returned after a request is accepted and successfully processed. The following data encoded in JSON format is returned in the response body.

Name	Type	Value (example)	Description
receivedParams	JSON object	{"locationId": "LSVMNVVT"}	The echo of the request parameters in JSON.
locations	JSON list object	[ {...}, {...} ]	The list of locations for the given location ID. It should contain only one item.

Each JSON object in the returned list has the following attributes:

Name	Type	Value (example)	Description
address	string	7135 S DECATUR BLVD, LAS VEGAS, NV 89118-4376, UNITED STATES	The postal address.
latitude	string	36.059347	The latitude of the geo location.
longitude	string	-115.208008	The longitude of the geo location.
locationId	string	LSVMNVVT	The unique identifier of the geo location.

Example:

```
{
  "receivedParams":
    {
      "locationId": "LSVMNVVT"
    },
  "locations":
    [
      {
        "address": "7135 S DECATUR BLVD, LAS VEGAS, NV 89118-4376, UNITED STATES",
        "latitude": "36.059347",
        "longitude": "-115.208008",
        "locationId": "LSVMNVVT"
      }
    ]
}
```

## Failure Response

A response with some failure HTTP status code is returned when the request is rejected. The following data encoded in JSON format is returned in the response body to carry more details.

Name	Type	Value (example)	Description
statusCode	integer	400	The HTTP status code following HTTP standard.
code	integer	400003	A Level 3 generated code combining HTTP code with sub-category code.
message	string	Missing data	Failure info at summarized level.
detail	string	The locationId is missing.	Info in detail about the failure.

Example:

```
{
  "exception":
    {
      "statusCode": "400",
      "code": "400003",
      "message": "Missing data",
      "detail": "The locationId is missing."
    }
}
```

## Digest-Based Security

Digest-based security uses two factors to authenticate incoming requests:

- **Client IP**—All requests must come from a white-listed range of IP addresses associated with the provided Application Key. The IP addresses are configured during account setup.
- **Secure Digital Signature**—Calls are authenticated using an encrypted digest signature that must be created using a shared Secret known only to Level 3 and the customer.

With this security method, users are assigned an Application Key and shared Secret during the onboarding process. **Application Keys can be shared publicly, but Secrets should be closely guarded. Only you and Level 3 should know your Secret.** If your Secret becomes compromised, contact the Level 3 Portal Support Center immediately to request a new one.

To access the Level 3 API, your software client must sign each request by generating a digest value using the current epoch time and your Secret. The generated digest value must be placed in the appropriate HTTP header (described below). The Secret value itself should never be included directly in the HTTP headers. For additional tips generating a digest see below: **Implementing Security**

### On-Boarding for Digest Based API Access

At this time, you will need to contact your Level 3 Account Manager in order to obtain Level 3 OpenAPI access.

### Requirements For a Digest-Based Request

Required HTTP header keys for any API call are:

Header Key	Format	Description
X-Level3-Application-Key	String	A valid application key. To obtain the key, please see How to sign up
X-Level3-Digest	binary (base64 encoded)	The SHA-2 encrypted and base64 encoded value generated using your application key, shared secret value, and digest time. See Implementing Security.
X-Level3-Digest-Time	String	The epoch time used to seed the creation of your digest. Epoch time is defined here as the number of seconds elapsed since Jan, 1 1970. See <a href="http://en.wikipedia.org/wiki/Epoch_time">http://en.wikipedia.org/wiki/Epoch_time</a>
Accept	String	application/json. Customers can submit requests and receive results in the format of their choice.
Content-Type	String	Informs the web server about the data format of the incoming request body. Valid values include application/json.

## Generating A Digest

Gather the following required data and libraries:

- A string version of the current epoch time. Epoch time is defined here as the number of seconds elapsed since Jan, 1 1970 (See [Wikipedia](#) for details).
- Your Level 3 Secret
- A cryptographic library that supports your client development language. One example of a freely available library for both Java and C# can be found at [www.bouncycastle.org](http://www.bouncycastle.org)

Next, generate a digest:

- Retrieve the current epoch time value as a string.
- Use the epoch time string and your Secret to seed the SHA2 algorithm. The epoch time string value will be the data field that gets encrypted with the SHA2 algorithm.
- Base64 encode the output binary digest.
- The encoded digest value will be used in the HTTP header: X-Level3-Digest.
- The epoch time string value will be used in the HTTP header: X-Level3-Digest-Time.

## Helpful Tips for Digest Generation

If any error messages are returned from the API regarding the Digest value, here are a few tips that may be useful to help correct the error. If your App Key is provisioned for both Test and Production, there is a separate Secret for each environment. Again, be sure that you are generating the Digest using the correct Secret for the environment you are accessing.

### Error Message: "Digest Time provided not within 15 minutes".

- This means the client input for X-Level3-Digest-Time is not close enough to the current computer system time and has expired. API security provides a grace period of +/- 15 minutes from the current Epoch time. If digest time is outside this window, an error will be returned to the client.
- Verify that your computer's current system time is accurate. Check out [epochconverter.com](http://epochconverter.com) to check the accuracy of your computer system time. If system time is not accurate, you may need to change your computer's time server to point to a more accurate server (like [time.nist.gov](http://time.nist.gov))
- Make sure that the Epoch time value used is in the format of a String-based integer that doesn't contain a decimal point or any other special characters.

### Error Message: "Digest provided does not match".

- Verify you are using the correct App Key and Secret for the environment you are trying to access (Test vs. Production).
- Verify that the X-Level3-Digest value is encoded using Base64.
- Make sure that the Epoch time value used is in the format of a String-based integer that doesn't contain a decimal point or any other special characters.

## Digest Sample Code

### How to Generate a Digest

This section lists code samples for use in generating a digest. For more details on authentication requirements, see the "[API Basics](#)" page.

#### Java Code Sample: Calling A Create Digest Function

```
@Test
public void testCreateDigestFromRequestData() {
    HttpDigestOAuth httpDigester = new HttpDigestOAuth();
    try {
        String epochTime = (String) String.valueOf( new Date().getTime());
        String appKeySecret = "YourAppKeySecretHere";
        String digest = httpDigester.calculateRFC2104HMAC(epochTime,
            appKeySecret );
        System.out.println("Date:" + epochTime());
        System.out.println("App Key Secret:" + appKeySecret );
        System.out.println("Authorization: 1234567:" + digest);
        int len = digest.length();
        Assert.assertTrue(len > 0);
    } catch (Exception e) {
        ...
    }
}
```

#### Java Code Sample: Creating The Digest With Bouncy Castle Libraries

```
...
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import org.bouncycastle.util.encoders.Base64;
...
public String calculateRFC2104HMAC(String epochTime, String appKeySecret) throws
java.security.SignatureException {
    String result;
    // get an hmac_sha2 key from the raw key bytes
    SecretKeySpec signingKey = new SecretKeySpec(appKeySecret.getBytes(), algorithm);
    // get an hmac_sha2 Mac instance and initialize with the signing key
    Mac mac = Mac.getInstance("HmacSHA256");
    mac.init(signingKey);
    // compute the hmac on input epochTime bytes
    byte[] rawHmac = mac.doFinal(epochTime.getBytes());
    // base64-encode the hmac
    result = new String(Base64.encode(rawHmac));
    return result;
}
```

## Perl Code Sample

```
#!/usr/local/bin/perl
use Digest::SHA qw(hmac_sha256_base64);
$appKeySecret = "YourAppKeySecretHere";
$epochTime = "1302641652";
$digest = hmac_sha256_base64( $epochTime, $appKeySecret );
# correction for base64 digest
while ( length($digest) % 4 ) {
    $digest .= '=';
}
print $digest;
```

## Python Code Sample

```
#!/usr/local/bin/python
import hmac
import hashlib
import base64
appKeySecret = "YourAppKeySecretHere"
epochTime = "1302641652"
a = base64.b64encode(hmac.new(appKeySecret, msg=epochTime,
digestmod=hashlib.sha256).digest())
print a
```

## C# Code Sample

```
//
// Function to create TheHMACDigest
//
private void BuildTheHMACDigest()
{
    //key to Encode the Data
    string appKeySecret = "yourAppKeySecretHere";
    //Calculate epoch time
    TimeSpan epochTime = (DateTime.Now - new DateTime(1970, 1, 1));
    //Data to be encoded
    long Data = Convert.ToInt64(epochTime.TotalSeconds);
    //Create instance for ASCIIEncoding
    System.Text.ASCIIEncoding Encoding = new System.Text.ASCIIEncoding();
    //Get encoded key in bytes format
    byte[] KeyByte = Encoding.GetBytes(appKeySecret);
    //Craete instance for HMACSHA256 and assign the key to the algorithm
    HMACSHA256 HMASHA = new HMACSHA256(KeyByte);
    //Get encoded Data in bytes
    byte[] ByteData = Encoding.GetBytes(Data.ToString());
    //Get the hashed Data in bytes
    byte[] HashedDataOutput = HMASHA.ComputeHash(ByteData);
    //Get the output in string format
```

```
string HashedStringOutPut = ByteToString(HashedDataOutput);
//Get the output in base64 format
string Base64output = Convert.ToBase64String(HashedDataOutput);
}
//
// Function to Convert HashedByte Output to String
//
// byte
// string
private string ByteToString(byte[] buff)
{
    string sbinary = "";
    for (int i = 0; i < buff.Length; i++)
    {
        sbinary += buff[i].ToString("X2"); // hex format
    }
    return (sbinary);
}
```

## Troubleshooting

Issues sometimes occur in submitting or receiving data in the API. In addition to the support below, error-related resources in the API may appear. For more information, see [Error Resources](#).

### Why Did My Call Fail?

Unauthenticated API requests are rejected with an appropriate HTTP status code and error message. Some common reasons for rejecting calls:

- The request was not sent over HTTPS, which is a requirement.
- The request was received from an IP address outside the authorized white-listed range.
- The digest value is not Base64 encoded.
- The digest value is not in the correct HTTP header: X-Level3-Digest
- Additional digest generation tips are available in the section above entitled [Helpful Tips for Digest Generation](#)
- Your application key is not authorized for the requested Level 3 resource.
- The URI capitalization is incorrect. Since the URI parser is case-sensitive, please observe those requirements.

### Interpreting Results: Mediation Id

- All responses from the Level 3 API include a header: X-Level3-Mediation-Id
- This Mediation-Id is a globally unique identifier used to identify the API call that was just made. If there is ever a need to contact Level 3 and track down an issue related to a call, this is the identifier you will need to provide.
- It is strongly recommended that you include this Mediation-Id in your audit logs along with any other useful troubleshooting details.



## OAuth2-Based Security

---

The OAuth 2.0 Specification is widely recognized as an industry standard method of authorizing a client application to access a protected resource (or HTTP service) on behalf of a resource owner. The resource owner is an entity (or end user) that is capable of granting access to a protected resource. Although not yet available in production, it should be noted the OAuth2 specification is the go-forward method of authorization for all OpenAPI services offered by Level 3. For planning purposes, please note that the currently supported Digest Authorization method will be deprecated.

### On-Boarding for OAuth2 OpenAPI Access

Once the OAuth2 access method is released in production (Q4), you will need to contact your Level 3 Account Manager to obtain Level 3 OpenAPI access.

### OAuth Production and Test Environments

Access to all environments is protected through the use of your OAuth credentials. The Level 3 API provides two environments: one for testing your requests and one for making requests of your actual data in production. The Test environment does not contain reliable data, so please do not use these results to inform your business. **However, Production data is stable** and should be used after your requests have been sufficiently tested.

For access to the Test environment, please contact either your Account Manager or the Level 3 Portal Support Center (PSC).

#### OAuth Request URI: Test Environment

After receiving your **OAuth** credentials for the Test environment, use the following base URI to make your API requests:

<https://den1-apibeta.level3.com>

Note that if you have generated any data in Level 3's Test systems, that data will not be present in the Production environment. Please keep this in mind when developing any test or regression scripts and be sure you can regenerate any test data you will need. Production data is persistent and unaffected.

#### OAuth Request URI: Production Environment

All requests in the production environment for OAuth2 enabled services are sent to this base URI:

<https://den1-api.level3.com>

## Accessing OAuth-Based API Services

Each API resource request must include a set of HTTP headers, which includes an **access token** which is obtained from the appropriate Level 3 OAuth2 token server. The following token server URI's for test and production are provided:

#### Access Token Server - TEST

<https://den1-apibeta.level3.com/auth/oauth/v2/token>

#### Access Token Server - PRODUCTION

<https://den1-api.level3.com/auth/oauth/v2/token>

## Making an OAuth-Based API Request

Making an API request using the OAuth2 method is a two-stage process:

1. Make a request to obtain an access token.
2. Make the actual API request using the token value returned from step 1.

### Step 1 – Access Token Request

For the first step, you will need to make a **POST** request to the server with the following required parameters:

#### POST Parameters for Access Token Request

parameter	value
client_id	[ <i>your client_id</i> ]
client_secret	[ <i>your secret</i> ]
grant_type	client_credentials
scope	[ <i>your scope(s)</i> ]

The scope parameter defines which API service(s) are granted to the client application and must be passed as a parameter to the token authorization server. If there is more than one scope required for a set of resources, the scope values must be separated by spaces.

## OAuth Sample Code

Example **POST** request:

```
curl -X POST https://api-test.level3.com/auth/oauth/v2/token
-d "client_id=123&client_secret=ABC&grant_type=client_credentials&scope=TICKET QUOTE"
-H "Content-Type: application/x-www-form-urlencoded"
```

Example:

```
{
  "access_token": "f67a352d-2cf2-4d5a-936d-ea943b6eeda1",
  "token_type": "Bearer",
  "expires_in": 3600,
  "refresh_token": "c6a152a5-6885-44dd-b84b-20a39a30e630",
  "scope": "TICKET QUOTE"
}
```

## Step 2 - The API Request

Once you have the **access\_token** from step 1 above, you will then be able to make the actual API request by putting the value into an **HTTP Authorization** header as a **Bearer** token type. When making the API request the client should also include the Content-Type and Accept headers.

Example GET request for the Ticketing API:

```
curl -X GET https://api.level3.com/OA/v1/Ticketing/11813838?customerNumber=123456
-H "Content-Type: application/json"
-H "Accept: application/json"
-H "Authorization: Bearer f67a352d-2cf2-4d5a-936d-ea943b6eeda1"
```

Your application should re-use the same Bearer access token on subsequent requests until that token expires. Once the token expires, your application should obtain a new one by sending another POST a request to the token server and passing the refresh\_token value within the grant\_type parameter which is described next.

### Using the Refresh Token

In the initial response returned from the access token server you may have noticed the **refresh\_token** attribute. Your client application will need to cache this refresh\_token or somehow make it re-usable. Your client application also needs to contain some logic to know if the token is expired. If the token is expired, you need to get a new one by making another request to the token server and use the **refresh\_token** value in the **grant\_type** parameter.

This refresh token POST request requires two HTTP headers: Content-Type and Authorization Basic. The Content-Type header value must be application/x-www-form-urlencoded, and the Authorization Basic header must be the Base64 encoding of your "client\_id:client\_secret".

### Java Spring Example for Basic Authorization header

```
protected String getBasicAuthorizationHeaderValue() {
    String auth = clientId + ":" + clientSecret;
    byte[] encodedAuth = Base64.encodeBase64(auth.getBytes(Charset.forName("UTF-8")));
    return "Basic " + new String(encodedAuth);
}
```

Example POST request using the refresh\_token method:

```
curl -X POST https://api.level3.com/OA/v1/Ticketing
-d "grant_type=refresh_token&refresh_token=c6a152a5-6885-44dd-b84b-20a39a30e630"
-H "Content-Type: application/x-www-form-urlencoded"
-H "Authorization: Basic
YzdIODJkMjctNjM1OC00ZWNILTg0MWUtZDI4MWE4YzZcwMzUwOjgzYWNiMmUyLTY1NzMDg0LTY1
NDIIMmZmNmI5Mg=="
```

## Interpreting API Results: HTTP Status Codes

- 200-range codes are successful server calls.
- 400-range codes are generally a client-side error. This generally means the problem lies in the client application calling the service (web request may be improperly formatted or the resource name they requested may not exist).
- 500-range codes are server-side error. This means the problem is due to an error that occurred on a Level 3 server (server or database could be down or temporarily unavailable).
- For all Client and Server error codes (4xx and 5xx) a Level3Response will be returned along with a description of the error.

### Standard HTTP Status Codes

HTTP Status Code	Description
200 OK	The standard response for a successful HTTP request.
201 CREATED	The request has been fulfilled and resulted in a new resource being created (such as a new trouble ticket). A link to the newly created resource is returned in the Location header.
202 ACCEPTED	The request has been accepted for processing, but the processing has not yet been completed.
204 NO CONTENT	The server successfully processed the request, but is not returning any content.
400 BAD REQUEST	The request cannot be processed due to improper syntax. Refer to documentation that outlines the correct format for a given resource request.
401 UNAUTHORIZED	The requested resource requires authentication and it has failed or has not yet been provided. ESP may have to verify that the customer has been granted access to the resource.
403 FORBIDDEN	Usually means that the provided credentials were successfully authenticated but still do not grant the client permission to access the resource.
404 NOT FOUND	Requested resource not found at the location specified by the client.
405 METHOD NOT ALLOWED	A request was made of a resource using an unsupported method. For example, the client may have tried to use a GET request on a resource that requires the POST method.
406 NOT ACCEPTABLE	The resource cannot generate content that is acceptable to the Accept headers provided.
408 REQUEST TIMEOUT	The server timed out waiting for the client to produce a request within the timeframe allowed by the server.
415 UNSUPPORTED MEDIA TYPE	Client request contains a format that isn't supported by the server. For example, the server may only support XML and JSON response formats.
500 INTERNAL SERVER ERROR	A generic error message to indicate something went wrong on the server end. In this case, a Level3Response will be returned along with details of the error including the source system experiencing issues.
501 NOT IMPLEMENTED	Either the server lacks the ability to fulfill the request or doesn't recognize the request method. For example GET and POST are supported, but currently PATCH is not. A Level3Response will be returned with error details.
503 SERVICE UNAVAILABLE	The server is currently unavailable, possibly because it is overloaded, offline, or down for maintenance. A Level3Response will be returned with error details. Escalate to Tier 4.
504 GATEWAY TIMEOUT	Error to internal service client. The server did not receive a timely response from the upstream server.
505 HTTP VERSION NOT SUPPORTED	Server doesn't support the HTTP protocol version used in the request.