

# Quoting Rest Routes

---

- [Address Verification](#)
- [Product Catalog](#)
- [Product Catalog Item](#)
- [Quote](#)
- [Order](#)

## API Flow

---

Product Catalog -> Product Code Details -> Address Verification -> Quote  
-> Order

The Zayo Quoting API is composed of a series of API calls that are intended be leveraged sequentially with each request/response cycle providing the required inputs for the next step in the quoting flow. Although it is best practice to have a typical quote following this lifecycle from beginning to end, it is not required and not the only workflow that can be constructed.

This best practice flow will follow the following lifecycle:

1. Product Catalog call to identify all available Product Codes and Version for the desired quote. (Product Catalog)
2. With productCode and version, it is now possible to get the product code details (Product Catalog Items) in order to identify how many locations will be needed to process the quote and dynamically start building the body of the quote request. (Product Code Details)
3. Once the number of locations needed is parsed from 2, client is now able to validate a building address in order to get a unique Zayo identifier for each of the locations. (Address Verification)
4. With locations in and the body of the quote request built, it is time to process the quote with the selection of desired values for each key. (Quote)
5. The Quote route will return a quoteID that enables the client to proceed with ordering (Order) OR edit the quote parameters and refresh the quote.

For a visual representation and example on how to implement, refer to the "Best Practices Development Example" document shared along with this wiki.

Modifications to existing Quote

Quote (GET) -> Quote (POST)

OR

Quote (GET) -> Quote (POST) -> Order (POST)

Another possible flow consists of making modifications to an existing quote. This can be done by:

1. Retrieve quote. Quote (GET)
2. Build body of request as if it was a new request, ADD quoteID to body under the root of the request and update fields to modify in quote. Quote (POST)
3. Optional, proceed with ordering using the original quoteID. Order (POST)

## Flexibility in lifecycle: independent calls

The best practice flow described above is necessary to ensure most up to date data and to minimize impact of non-breaking changes. The following flows are possible but not encouraged long term, Zayo will support with additional information on changes that might affect these development style, but cannot guarantee notice on changes considered non-breaking that affect results in these models.

Requests without calling Address Verification every iteration

Additional calls for quotes can be made without first calling the address verification and product catalog calls, provided inputs for the quote are already known. However, Zayo's building and locations microservice is dynamic and locations can be added, modified or removed in time. IF calls are being executed without calling the address verification route, THEN it is strongly advised to have a defined cadence (e.g. every 30 days) to refresh locationIDs and check if new locations have been enabled.

Requests without calling the Product Catalog every iteration

The result responses for the Product Catalog are fairly static. The data returned is subject to change as new products and versions become available.

*Changes that break compatibility will be versioned and previous version can be hard coded during transition to avoid breaking compatibility.*

# Product Catalog

---

Returns available product codes and their respective version available for quoting. This route will always return

- URL

GET <https://api.zayo.com/services/quote-management/v1/catalog/product-code>

- URL Params

None

- Data Params

None

- Success Response:

- Code: 200 Ok

Content:

```
{
  "apiVersion": "v1",
  "data": {
    "productCodes": [
      {
        "productCode": "ETH-ELINE-UNI",
        "version": "1.0.0",
        "description": "Metro & Intercity Point-to-Multi-Point service prov
      },
      {
        "productCode": "ETH-ELINE-P2P",
        "version": "1.0.0",
        "description": "Metro and Intercity Point to Point service configur
      },
      {
        "productCode": "ETH-ELINE-NNI",
        "version": "1.0.0",
        "description": "Deliver full connectivity across your network by un
      },
      {
        "productCode": "ETH-ELAN-M2M",
        "version": "1.0.0",
```

```

        "description": "A Carrier Ethernet, multipoint to multipoint config
    },
    {
        "productCode": "ETH-PDN-P2P",
        "version": "1.0.0",
        "description": "Ethernet connectivity across a completely private m
    },
    {
        "productCode": "IP-DIA",
        "version": "1.0.0",
        "description": "Zayo's high performance Dedicated Internet Access (
    },
    {
        "productCode": "IP-DIA-AGG",
        "version": "1.0.0",
        "description": "Zayo's high performance Aggregated Dedicated Intern
    },
    {
        "productCode": "IP-VPN",
        "version": "1.0.0",
        "description": "IPVPN (Virtual Private Network using IP/MPLS backbo
    },
    {
        "productCode": "WAVES-STD-P2P",
        "version": "1.0.0",
        "description": "Dedicated standard point-to-point route with flexib
    }
}
]
}
}

```

- Error Response:

- Code: 400 Bad Request

Content:

```

{
  "apiVersion": "1.0",
  "error": {
    "code": 400,
    "message": "Bad Request",
    "errors": [{
      "reason": "BadRequestException",
      "message": "The request was invalid or cannot be otherwise served."
    }]
  }
}

```

- Sample Call (Node.js) :

```
var request = require("request");

var options = { method: 'GET',
  url: 'https://api.zayo.com/services/quote-management/v1/catalog/product-code',
  headers:
  { 'Authorization': '1e2264e7-19cf-4b97-a4e8-de6207e99c06',
    'Cache-Control': 'no-cache' } };

request(options, function (error, response, body) {
  if (error) throw new Error(error);

  console.log(body);
});
```

- Notes: as the API is versioned, this route will return latest version available. Backwards compatibility between versions will be provided for a period of time so development teams can transition to new versions before deprecating the previous one.

# Product Code Details

---

Returns configuration details for a given product code.

- URL

GET <https://api.zayo.com/services/quote-management/v1/catalog/product-code-details/> :productCode / :version

- URL Params

Name	Type	Description	Required?
productCode	string	unique Product Code from productCatalog response	Yes
version	string	version of Product Code derived from productCodes	Yes

- Data Params

None

- Success Response:
  - Product Code: WAVES-STD-P2P

Code: 200 Ok

Content:

```
{
  "apiVersion": "v1",
  "data": {
    "productCodeConfigurations": [
      {
        "configuration": {
          "nodeId": "c85fb31a-6c1b-4d21-8b78-0c318737d7d3",
          "productCode": "WAVES-STD-P2P",
          "version": "1.0.0",
          "minCount": 1,
          "maxCount": 1,
        }
      }
    ]
  }
}
```

```
"attrs": [
  {
    "name": "bandwidth",
    "isRequired": true,
    "dataType": "String",
    "options": [
      {
        "value": "1G",
        "defaultValue": false,
        "allowedForControllingFields": []
      },
      {
        "value": "2.5G",
        "defaultValue": false,
        "allowedForControllingFields": []
      },
      {
        "value": "10G",
        "defaultValue": false,
        "allowedForControllingFields": []
      },
      {
        "value": "40G",
        "defaultValue": false,
        "allowedForControllingFields": []
      },
      {
        "value": "100G",
        "defaultValue": false,
        "allowedForControllingFields": []
      }
    ]
  },
  {
    "name": "currencyIsoCode",
    "isRequired": false,
    "dataType": "String",
    "options": [
      {
        "value": "USD",
        "defaultValue": true,
        "allowedForControllingFields": []
      },
      {
        "value": "CAD",
        "defaultValue": false,
        "allowedForControllingFields": []
      },
      {
        "value": "EUR",
        "defaultValue": false,
        "allowedForControllingFields": []
      },
      {
        "value": "GBP",
        "defaultValue": false,
```

```

        "allowedForControllingFields": []
    },
    {
        "value": "HKD",
        "defaultValue": false,
        "allowedForControllingFields": []
    },
    {
        "value": "AUD",
        "defaultValue": false,
        "allowedForControllingFields": []
    },
    {
        "value": "CHF",
        "defaultValue": false,
        "allowedForControllingFields": []
    },
    {
        "value": "JPY",
        "defaultValue": false,
        "allowedForControllingFields": []
    },
    {
        "value": "SGD",
        "defaultValue": false,
        "allowedForControllingFields": []
    }
    ]
}
],
"children": [
    {
        "nodeId": "4f6276ac-0054-47ef-84ca-3b06e695828b",
        "productCode": "WAVES-STD-P2P",
        "version": "1.0.0",
        "minCount": 1,
        "maxCount": 1,
        "attrs": [
            {
                "name": "locationA",
                "isRequired": true,
                "dataType": "Id",
                "value": "Custom Id"
            }
        ],
        "children": []
    },
    {
        "nodeId": "9b45d904-8a15-43ca-8cfb-4e9f66f52643",
        "productCode": "WAVES-STD-P2P",
        "version": "1.0.0",
        "minCount": 1,
        "maxCount": 1,
        "attrs": [
            {
                "name": "locationZ",

```



```

    "isRequired": true,
    "dataType": "Id",
    "value": "Custom Id"
  }
],
"children": []
}
]
},
"preferences": {
  "isRequired": false,
  "attrs": [
    {
      "name": "requestFollowUp",
      "isRequired": false,
      "dataType": "String",
      "options": [
        {
          "value": "No",
          "defaultValue": true,
          "allowedForControllingFields": []
        },
        {
          "value": "Yes",
          "defaultValue": false,
          "allowedForControllingFields": []
        }
      ]
    }
  ]
}
}
]
}
}
}
}
}

```

- Response Attributes

Name	Type	Description
configuration	object	parent container for configuration hierarchy
nodeId	string	unique identifier for a configuration node
productCode	string	product code identifier for node(s)
version	string	product code version for node(s)

<code>minCount</code>	integer	minimum number of instance nodes allowed
<code>maxCount</code>	integer	maximum number of instance nodes allowed
<code>attrs</code>	array	container for attribute definitions for a given node
<code>name</code>	string	name of attribute
<code>isRequired</code>	boolean	indicator of whether attribute is required for node
<code>dataType</code>	string	data type of attribute
<code>options</code>	array	container for all attribute value options
<code>value</code>	string	a specific attribute value
<code>defaultValue</code>	boolean	indicator of attribute or dependent attribute value default
<code>allowedForControllingFields</code>	array	container for attribute controlling values
<code>controllingField</code>	array	specifies parent controller attribute for child attribute
<code>values</code>	array	specifies which child attribute values are valid for parent value(s)

- Error Response:
  - Code: 400 Bad Request

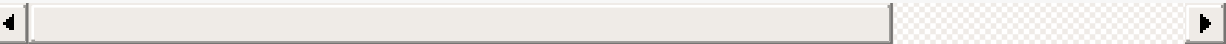
Content:

```
{
  "apiVersion": "1.0",
  "error": {
    "code": 400,
    "message": "Bad Request",
    "errors": [{
      "reason": "BadRequestException",
      "message": "The request was invalid or cannot be otherwise served."
    }]
  }
}
```

```
    }  
  }  
}
```

- Sample Call (Node.js) :

```
var request = require("request");  
  
var options = { method: 'GET',  
  url: 'https://api.zayo.com/services/quote-management/v1/catalog/product-code-data',  
  headers:  
    { 'Authorization': '1e2264e7-19cf-4b97-a4e8-de6207e99c06',  
      'Cache-Control': 'no-cache' } };  
  
request(options, function (error, response, body) {  
  if (error) throw new Error(error);  
  
  console.log(body);  
});
```



- Notes:

# Address Verification

---

Validates and returns unique street address (or addresses) and Zayo's unique identifier for the location to be used in quoting requests. It also provides insight on whether the location is On Zayo Network (On Net) vs Not On Zayo Network (Off Net\*).

\*Off net locations are often still "quotable" since Zayo's pricing engine calculates build cost and prices accordingly. Off net locations do not fall under customer specific pricing and will be higher than On Net rates.

- URL

`POST` <https://api.zayo.com/services/location-management/v1/buildings/validation>

- URL Params

None

- Data Params

Name	Type	Description	Required?
<code>address</code>	string	A string that represents the first line of a street address. In most cases this field will be used for street name and house number input, but it can also be used to input building name or place name.	Yes
<code>neighborhood</code>	string	The smallest administrative division associated with an address, typically, a neighborhood or a section of a larger populated place. A neighborhood is a subdivision of a city.	No
<code>city</code>	string	The next largest administrative division associated with an address, typically, a city or municipality. A city is a subdivision of a subregion or a region.	No
<code>subregion</code>	string	The next largest administrative division associated with an address. Depending	No

		on the country, a subregion can represent a county, state, or province.	
<code>region</code>	string	The largest administrative division associated with an address, typically, a state or province.	No
<code>countryCode</code>	string	Three letter ISO 3166-1 alpha-3 country code	Yes
<code>postal</code>	string	The standard postal code for an address, typically, a 3–6-digit alphanumeric code.	No

```

{
  "buildings": [
    {
      "address": "1 Galaxy Way",
      "neighborhood": "",
      "city": "Monroe",
      "subregion": "Snohomish",
      "region": "Washington",
      "countryCode": "USA",
      "postal": "98272"
    },
    {
      "address": "60 Hudson Street",
      "neighborhood": "",
      "city": "",
      "subregion": "",
      "region": "",
      "countryCode": "USA",
      "postal": "10013"
    }
  ]
}

```

## Response:

The JSON error format supports single or multiple errors. A response should contain either a data object or an error object. All attempts to return HTTP Status Codes are made.

HTTP Status Codes ([https://en.wikipedia.org/wiki/List\\_of\\_HTTP\\_status\\_codes](https://en.wikipedia.org/wiki/List_of_HTTP_status_codes))

Status	Text	Description
2xx	Ok	The request was successfully received, understood, and accepted
3xx	Redirection	Further action needs to be taken in order to complete the request
4xx	Client Error	The request contains bad syntax or cannot be fulfilled
5xx	Server Error	The server failed to fulfill an apparently valid request

- Success Response:

- Code: 200 Ok

Content:

`"matched_address"`: returns `"locationIdDefault"` to be used in QUOTE route.

`"unmatchedAddresses"`: building was a match but no quotable location is available for the address provided.

```
{
  "apiVersion": "v1",
  "data": {
    "matchedAddresses": [
      {
        "matched_address": "60 Hudson St, New York, 10013",
        "buildingId": "a0W600000009mrkEAA",
        "locationIdDefault": "a0b60000000DnN3AAK",
        "address": "60 Hudson Street",
        "neighborhood": null,
        "city": null,
        "subregion": null,
        "region": null,
      }
    ]
  }
}
```

```

        "country": "USA",
        "postal": "10013",
        "zayoNetworkStatus": "On Zayo Network"
    }
],
"unmatchedAddresses": [
    {
        "buildingId": null,
        "locationIdDefault": null,
        "address": "1 Galaxy Way",
        "neighborhood": null,
        "city": "Monroe",
        "subregion": "Snohomish",
        "region": "Washington",
        "country": "USA",
        "postal": "98272",
        "zayoNetworkStatus": null,
        "message": "A quotable location id is not yet available for this ad
    }
]
}
}

```

- Error Response:

- Code: 304 Not Modified

Content:

```

{
  "apiVersion": "1.0",
  "error": {
    "code": 304,
    "message": "Not Modified",
    "errors": [{
      "reason": "NotModifiedException",
      "message": "The request was invalid or cannot be otherwise served."
    }]
  }
}

```

- Sample Call (Node.js) :

```

var request = require("request");

```

```
var options = { method: 'POST',
  url: 'https://api.zayo.com/services/location-management/v1/buildings/validation',
  body: {
    "buildings": [
      {
        "address": "1 Galaxy Way",
        "neighborhood": "",
        "city": "Monroe",
        "subregion": "Snohomish",
        "region": "Washington",
        "countryCode": "USA",
        "postal": "98272"
      },
      {
        "address": "60 Hudson Street",
        "neighborhood": "",
        "city": "",
        "subregion": "",
        "region": "",
        "countryCode": "USA",
        "postal": "10013"
      }
    ]
  },
  headers:
  { 'Authorization': '1e2264e7-19cf-4b97-a4e8-de6207e99c06',
    'Cache-Control': 'no-cache',
    'Content-Type': 'application/json' },
  request(options, function (error, response, body) {
    if (error) throw new Error(error);

    console.log(body);
  });
```

- Notes:



# Quote

---

**POST** Creates/Updates a quote and returns a pricing and configuration object given configuration details. This route is set up to optionally pass a quoteId value. If a valid quoteId is passed, an update will be performed to the existing quote. If no quoteId is passed, an insert will be performed. In both cases, the quoteId and quoteNumber will be returned in the response.

- URL

**POST** <https://api.zayo.com/services/quote-management/v1/quotes/quote>

- URL Params

None

- Data Params

```
{
  "configuration": {
    "nodeId": "4C2BF0C9-285A-42FC-A639-24810A30DDD6",
    "productCode": "IP-DIA",
    "version": "1.0.0",
    "currencyIsoCode": "USD",
    "bandwidth": "10G",
    "burstableService": "Yes",
    "children": [
      {
        "productCode": "IP-DIA",
        "version": "1.0.0",
        "nodeId": "4F5AB389-2A09-4CF6-976B-26078046DFF9",
        "portHandoffSpeed": "10GigE",
        "locationA": "a0b6000000W6LjAAK"
      }
    ]
  },
  "preferences": {
    "requestFollowUp": "No"
  }
}
```

- Success Response:
  - Code: 200 Ok

Content:

```
{
  "apiVersion": "v1",
  "data": {
    "configuration": {
      "nodeId": "4C2BF0C9-285A-42FC-A639-24810A30DDD6",
      "productCode": "IP-DIA",
      "version": "1.0.0",
      "children": [
        {
          "productCode": "IP-DIA",
          "version": "1.0.0",
          "nodeId": "4F5AB389-2A09-4CF6-976B-26078046DFF9",
          "portHandoffSpeed": "10GigE",
          "locationA": "a0b6000000W6LjAAK"
        }
      ],
      "currencyIsoCode": "USD",
      "bandwidth": "10G",
      "burstableService": "Yes",
      "bandwidthType": "Standard",
      "aggregatedBandwidth": "No",
      "product": "DIA",
      "productCategory": "DIA",
      "locations": [
        {
          "locationId": "a0b6000000W6LjAAK",
          "address": "393 Inverness Pkwy Englewood, CO 80112",
          "floorSuiteRoom": "Fl-1 & 2/Ste-zColo",
          "buildingNetworkStatus": "On Zayo Network",
          "locationNetworkStatus": "On Zayo Network"
        }
      ]
    },
    "preferences": {
      "requestFollowUp": "No"
    },
    "quoteId": "94f96839-5b25-49d9-841d-9daba79d03e8",
    "quoteNumber": "Quote-655960",
    "financialTerms": [
      {
        "term": "12",
        "totalMrc": 5718.75,
        "totalNrc": 5000
      },
      {
        "term": "24",
        "totalMrc": 5261.25,
        "totalNrc": 5000
      }
    ]
  }
}
```

```

        "term": "36",
        "totalMrc": 4575,
        "totalNrc": 5000
    },
    {
        "term": "48",
        "totalMrc": 3888.75,
        "totalNrc": 5000
    },
    {
        "term": "60",
        "totalMrc": 3431.25,
        "totalNrc": 5000
    }
]
}
}

```

- Error Response:
  - Code: 400 Bad Request

Content:

```

{
  "apiVersion": "1.0",
  "error": {
    "code": 400,
    "message": "Bad Request",
    "errors": [{
      "reason": "BadRequestException",
      "message": "The request was invalid or cannot be otherwise served."
    }]
  }
}

```

- Sample Call (Node.js) :

```

var request = require("request");

var options = { method: 'POST',
  url: 'https://api.zayo.com/services/quote-management/v1/quotes/quote',
  headers:
    { 'Authorization': '1e2264e7-19cf-4b97-a4e8-de6207e99c06',
      'Cache-Control': 'no-cache',
      'Content-Type': 'application/json' },
  body:
    { quoteId: 'b87cbfe0-15a1-4953-a0a1-827fa0a85236',

```

```

configuration:
  { nodeId: '4C2BF0C9-285A-42FC-A639-24810A30DDD6',
    productCode: 'IP-DIA',
    version: '1.0.0',
    currencyIsoCode: 'USD',
    bandwidth: '1G',
    burstableService: 'Yes',
    children:
      [ { productCode: 'IP-DIA',
          version: '1.0.0',
          nodeId: '4F5AB389-2A09-4CF6-976B-26078046DFF9',
          portHandoffSpeed: 'GigE',
          locationA: 'a0b6000000DnN3AAK' } ] } },
json: true };

request(options, function (error, response, body) {
  if (error) throw new Error(error);

  console.log(body);
});

```

**GET** Returns an existing quote object. This should be used to retrieve the quote prior to making a POST call to update quote properties or to invoking Order.

- URL

**GET** <https://api.zayo.com/services/quote-management/v1/quotes/quote/:quoteId>

- URL Params

Name	Type	Description	Required?
quoteId	string	unique quote identifier	Yes

- Data Params

None

- Success Response:

- Code: 200 Ok

Content:

```

{
  "apiVersion": "v1",

```

```
"data": {
  "quoteId": "b87cbfe0-15a1-4953-a0a1-827fa0a85236",
  "configuration": {
    "nodeId": "4C2BF0C9-285A-42FC-A639-24810A30DDD6",
    "productCode": "IP-DIA",
    "version": "1.0.0",
    "children": [
      {
        "productCode": "IP-DIA",
        "version": "1.0.0",
        "nodeId": "4F5AB389-2A09-4CF6-976B-26078046DFF9",
        "portHandoffSpeed": "GigE",
        "locationA": "a0b6000000DnN3AAK"
      }
    ],
    "currencyIsoCode": "USD",
    "bandwidth": "1G",
    "burstableService": "Yes",
    "product": "DIA",
    "productCategory": "DIA",
    "aggregatedBandwidth": "No",
    "bandwidthType": "Standard",
    "locations": [
      {
        "locationId": "a0b6000000DnN3AAK",
        "address": "60 Hudson St New York, NY 10013",
        "floorSuiteRoom": "Fl-12/Ste-1211",
        "buildingNetworkStatus": "On Zayo Network",
        "locationNetworkStatus": "On Zayo Network"
      }
    ]
  },
  "quoteNumber": "Quote-577123",
  "financialTerms": [
    {
      "term": "12",
      "totalMrc": 1518.75,
      "totalNrc": 1000
    },
    {
      "term": "24",
      "totalMrc": 1397.25,
      "totalNrc": 1000
    },
    {
      "term": "36",
      "totalMrc": 1215,
      "totalNrc": 1000
    },
    {
      "term": "48",
      "totalMrc": 1032.75,
      "totalNrc": 1000
    },
    {
      "term": "60",
```

```
        "totalMrc": 911.25,  
        "totalNrc": 1000  
      }  
    ]  
  }  
}
```

- Error Response:

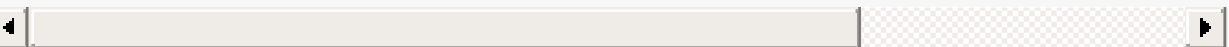
- Code: 400 Bad Request

Content:

```
{  
  "apiVersion": "1.0",  
  "error": {  
    "code": 400,  
    "message": "Bad Request",  
    "errors": [{  
      "reason": "BadRequestException",  
      "message": "The request was invalid or cannot be otherwise served."  
    }]  
  }  
}
```

- Sample Call (Node.js) :

```
var request = require("request");  
  
var options = { method: 'GET',  
  url: 'https://api.zayo.com/services/quote-management/v1/quotes/quote/b87cbfe0-15a',  
  headers:  
    { 'Authorization': '1e2264e7-19cf-4b97-a4e8-de6207e99c06',  
      'Cache-Control': 'no-cache' } }];  
  
request(options, function (error, response, body) {  
  if (error) throw new Error(error);  
  
  console.log(body);  
});
```



- Notes: