Express Connect

API Reference

API Reference

Introduction

Welcome to Express Connect. You can use this document to get familiar with the relevant API operations of the Express Connect.

Before using these APIs, make sure that you fully understand the product, user agreements, and billing methods.

Glossary

Term	Description
Express Connect	Express Connect uses Alibaba Cloud’s excellent infrastructure to create data transmission channels. It establishes secure and reliable intranet connections between different networks. For example, between VPCs or between VPC and IDC machine room.
Virtual Private Cloud	An Alibaba Virtual Private Cloud (VPC) is a customized virtual private cloud built on the basis of Alibaba Cloud to achieve full logical isolation between Alibaba VPCs. You can create and manage cloud product instances, such as ECS, Intranet Sever Load Balancer, and RDS, in your own VPC.
Physical Connection	A physical connection is the abstraction of a physical line used to directly connect a customer to Alibaba Cloud. Every connection a customer uses to access Alibaba Cloud, is considered as one Alibaba Cloud physical connection object under the customer’s name.
VRouter	A VRouter is a hub in the VPC, connecting all VSwitches in the VPC and serving as a gateway device that connects the VPC to other networks as well. It forwards network traffic according to specific route entries.
Virtual Boarder Router	Customers can create multiple virtual border routers (VBRs) on a physical connection. Each VBR is responsible for forwarding the data of one VLAN on the leased line to an address in

	Alibaba Cloud. Using VBRs, customer data can be directly transmitted to any Alibaba Cloud region.
Router Interface	A Router Interface is a virtual network device. It can be attached to a VRouter to establish an Express Connect connection with another Router Interface. This creates an intranet connection between different networks.
Route Table	A route table refers to a list of route entries on the VRouter.
Route Entry	Each item in the route table is a route entry. A route entry defines the next hop address for the network traffic to be routed to the specified destination CIDR block. Route entries are categorized into system routes and custom routes.

Description of Business Restrictions and Type Constraints

In the interface description section, whenever there is a conflict with the type constraints given on the official site concerning optional parameter values or available specifications, the information on the official site will prevail.

Call Method

API interfaces are called by sending HTTP GET requests to the API server address, with corresponding request parameters included in the request according to the interface instructions. The system returns the processing result according to the request processing status.

Request Structure

Service Address

The service access URL of API is:vpc.aliyuncs.com

Communication Protocol

Supports request communication via HTTP or HTTPS channels. HTTPS is recommended because it

provides better security.

Request Methods

Supports sending request via HTTP GET. In this mode, the request parameter has to be included in the request URL.

Request Parameters

You must specify an action for each request, that is, the 'Action' parameter (such as StartInstance). Every action must include the required public request parameters and the request parameters specified for this action.

Character Encoding

Requests and returned results are both encoded using the UTF-8 character set.

Public Parameters

Public Request Parameters

Public request parameters refer to the request parameters that need to be used by every interface.

Name	Type	Required	Description
Format	String	No	Type of value returned, JSON and XML supported. XML is the default.
Version	String	Yes	The API version, with the date format: YYYY-MM-DD. The current version is 2016-04-28.
AccessKeyId	String	Yes	The key ID Alibaba Cloud issued to a user to access services.
Signature	String	Yes	The signature result string. For details on the signature calculation method, please refer to Signature mechanism.
SignatureMethod	String	Yes	The signature method.

			HMAC-SHA1 currently supported.
Timestamp	String	Yes	Request time stamp. The date is represented according to ISO8601, and UTC is required. Format: YYYY-MM-DDThh:mm:ssZ For example, 2014-05-26T12:00:00Z (equivalent to 20:00:00, May 26, 2014 Beijing time).
SignatureVersion	String	Yes	Signature algorithm version. The current version is 1.0.
SignatureNonce	String	Yes	A unique random number, used to prevent replay attacks. The user must use different random numbers for different requests

Examples:

```
https://vpc.aliyuncs.com/
?Format=xml
&Version=2014-05-26
&Signature=Pc5WB8gokVn0xfeu%2FZV%2BiNM1dgI%3D
&SignatureMethod=HMAC-SHA1
&SignatureNonce=15215528852396
&SignatureVersion=1.0
&AccessKeyId=key-test
&Timestamp=2012-06-01T12:00:00Z
...
```

Public Return Parameters

Each time the user sends a call request to an interface, no matter if it is successful or not, the system will return a unique identification code (RequestId) to the user.

Examples:

XML Examples:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!--Result Root Node-->
<Interface Name+Response>
<--Return Request Tag-->
<RequestId>4C467B38-3910-447D-87BC-AC049166F216</RequestId>
<!--Return Result Data-->
</Interface Name+Response>
```

JSON Examples:

```
{
"RequestId": "4C467B38-3910-447D-87BC-AC049166F216",
/* Return Result Data*/
}
```

Returned Results

After the API service is called, the returned data adopts a uniform format. A returned HTTP status code of 2xx indicates that the call was successful. A returned HTTP status code of 4xx or 5xx indicates that the call was unsuccessful. For successful calls, the supported formats of returned data are XML and JSON. When a request is sent, an external system can pass in parameters to specify the format of returned data. XML is the default format. In this document, examples of results returned are formatted in a way that is easier for users to view. The actual results returned are not formatted with line breaks, indentation, etc.

Successful Results

XML Examples:

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Result Root Node-->
<Interface Name+Response>
<!--Return Request Tag-->
<RequestId>4C467B38-3910-447D-87BC-AC049166F216</RequestId>
<!--Return Result Data-->
</Interface Name+Response>
```

JSON Examples:

```
{
"RequestId": "4C467B38-3910-447D-87BC-AC049166F216",
```

```
/* Return Result Data*/
}
```

Error Results

After an error is encountered in an interface call, no result data will be returned. The caller can pinpoint the causes of error based on the error codes corresponding to each interface and the public error codes that are listed in 2.3.3 below.

When an error occurs in a call, the HTTP request will return an HTTP status code of 4xx or 5xx. The returned message body includes the specific error code and error message. It also contains a globally unique request ID: RequestId and the ID of the site you accessed with this request: HostId. If the caller cannot find the cause of the error, he can contact Alibaba Cloud customer service and provide the HostId and RequestId to help us solve the problem as quickly as possible.

XML Examples:

```
<?xml version="1.0" encoding="UTF-8"?>
<Error>
<RequestId>8906582E-6722-409A-A6C4-0E7863B733A5</RequestId>
<HostId>vpc.aliyuncs.com</HostId>
<Code>UnsupportedOperation</Code>
<Message>The specified action is not supported.</Message>
</Error>
```

JSON Examples:

```
{
"RequestId": "8906582E-6722-409A-A6C4-0E7863B733A5",
"HostId": "vpc.aliyuncs.com",
"Code": "UnsupportedOperation",
"Message": "The specified action is not supported."
}
```

Public Error Codes

Error code	Description	Http status code	Meaning
MissingParameter	The input parameter “Action” that is mandatory for processing this request is not supplied	400	The Action field missing
MissingParameter	The input parameter “AccessKeyId”	400	The AccessKeyId field missing

	that is mandatory for processing this request is not supplied		
MissingParameter	An input parameter “Signature” that is mandatory for processing the request is not supplied.	400	The Signature field missing
MissingParameter	The input parameter “TimeStamp” that is mandatory for processing this request is not supplied	400	The Timestamp field missing
MissingParameter	The input parameter “Version” that is mandatory for processing this request is not supplied	400	The Version field missing
InvalidParameter	The specified parameter “Action or Version” is not valid.	400	Invalid Action value (This API does not exist)
InvalidAccessKeyId.NotFound	The Access Key ID provided does not exist in our records.	400	Invalid AccessKeyId value (This key does not exist)
Forbidden.AccessKeyDisabled	The Access Key is disabled.	403	The AccessKey is disabled
IncompleteSignature	The request signature does not conform to Alibaba Cloud standards.	400	Invalid Signature value (incorrect signature result)
InvalidParamater	The specified parameter “SignatureMethod” is not valid.	400	Invalid SignatureMethod value
InvalidParamater	The specified parameter “SignatureVersion” is not valid.	400	Invalid SignatureVersion value
IllegalTimestamp	The input parameter “Timestamp” that is mandatory for processing this request is not supplied.	400	Invalid Timestamp value (The gap between Timestamp and server time exceeds one hour)
SignatureNonceUse	The request	400	Invalid

d	signature nonce has been used.		SignatureNonce (This SignatureNonce value has been used)
InvalidParameter	The specified parameter “Action or Version” is not valid.	400	Invalid Version value
InvalidOwnerId	The specified OwnerId is not valid.	400	Invalid OwnerId value
InvalidOwnerAccount	The specified OwnerAccount is not valid.	400	Invalid OwnerAccount value
InvalidOwner	OwnerId and OwnerAccount can’t be used at one API access.	400	The OwnerId and OwnerAccount are used at the same time
Throttling	Request was denied due to request throttling.	400	Access is denied due to the system traffic control
Throttling	Request was denied due to request throttling.	400	The call quota for this key is used up
InvalidAction	Specified action is not valid.	403	This key is not authorized to call this API
UnsupportedHTTPMethod	This http method is not supported.	403	The user adopts unsupported Http method (currently **TOP** only supports post and get)
ServiceUnavailable	The request has failed due to a temporary failure of the server.	500	The service is unavailable
UnsupportedParameter	The parameter “<parameter name>” is not supported.	400	Invalid parameter is used
InternalError	The request processing has failed due to some unknown error, exception or failure.	500	Other causes
MissingParameter	The input parameter OwnerId,OwnerAccount that is mandatory for	403	No assigned OwnerId when calling this interface

	processing this request is not supplied.		
Forbidden.SubUser	The specified action is not available for you	403	Not authorized to call the Order class interface
UnsupportedParameter	The parameter “<parameter name>” is not supported.	400	Not authorized to use this parameter
Forbidden.InstanceNotFound	The specified Instance does not exist, so we cannot get enough information to check permission in RAM.	404	RAM is used to provide cross-account access to resources, but the Instance involved in the access does not exist
Forbidden.DiskNotFound	The specified Disk does not exist, so we cannot get enough information to check permission in RAM.	404	RAM is used to provide cross-account access to resources, but the Disk involved in the access does not exist
Forbidden.SecurityGroupNotFound	The specified SecurityGroup does not exist, so we cannot get enough information to check permission in RAM.	404	RAM is used to provide cross-account access to resources, but the SecurityGroup involved in the access does not exist
Forbidden.SnapshotNotFound	The specified Snapshot does not exist, so we cannot get enough information to check permission in RAM.	404	RAM is used to provide cross-account access to resources, but the Snapshot involved in the access does not exist
Forbidden.ImageNotFound	The specified Image does not exist, so we cannot get enough information to check permission in RAM.	404	RAM is used to provide cross-account access to resources, but the Image involved in the access does not exist
Forbidden.RAM	User not authorized to operate on the specified resource, or this API does not support RAM.	403	RAM is used to provide cross-account access to resources, but users are not authorized to perform the

			operation this time
Forbidden.NotSupportRAM	This action does not support accessed by RAM mode.	403	RAM is not allowed to be used to access this interface (use of ResourceOwnerAccount not allowed)
Forbidden.RiskControl	This operation is forbidden by Alibaba Cloud Risk Control system.	403	Access denied by Alibaba Cloud Risk Control system
InsufficientBalance	Your account does not have enough balance.	400	Insufficient balance
IdempotentParameterMismatch	Request uses a client token in a previous request but is not identical to that request.	400	A used ClientToken is used for the request, which is different from the previous request with the used token.
RealNameAuthenticationError	Your account has not passed the real-name authentication yet.	403	Real-name authentication is not performed by the user
InvalidIdempotenceParameter.Mismatch	The specified parameters are different from before	403	Idempotence parameters do not match
LastTokenProcessing	The last token request is processing	403	The previous request is still being processed
InvalidParameter	The specified parameter is not valid	400	Parameter verification failed

Signature Mechanism

Each time it is accessed by a request, the service will perform sender authentication. Therefore, whether HTTP or HTTPS protocol is used to submit the request, the request must contain signature information. By using the Access Key ID and Access Key Secret, the service performs symmetric encryption to authenticate the sender request. The ‘Access Key ID’ and ‘Access Key Secret’ are officially issued to visitors by Alibaba Cloud (visitors can apply for and manage them on the official Alibaba Cloud website). The Access Key ID indicates the identity of the visitor. The Access Key Secret is the secret key used to encrypt the signature string and to verify the signature string on the server.

It must be kept strictly confidential and should only be known to Alibaba Cloud and the end user.

When a user makes a call to the server, the following method is used to sign the request:

The Canonicalized Query String is constructed using the request parameters
(a) The request parameters are ordered alphabetically by the parameter names (this includes the "public request parameters" and custom parameters for the given request interfaces described in this document, but not the Signature parameter mentioned in "Public Request Parameters").
Note: When a request is submitted using the GET method, these parameters are the parameter section of the request URI (i.e. the section in the URI following "?" and connected by "&").

(b) The name and value of each request parameter are encoded. The names and values must undergo URL encoding using the UTF-8 character set. The URL encoding rules are as follows:

i. The characters A-Z, a-z, 0-9, "-", "_", ".", and "~" are not encoded;
ii. Other characters are encoded in "%XY" format, with XY representing the characters' ASCII code in hexadecimal notation. For example, the English double quotes (") are encoded as %22
iii. Extended UTF-8 characters are encoded in "%XY%ZA..." format;
iv. It must be noted that the English space ( ) is encoded as %20, rather than the plus sign (+).

Note: Generally, libraries that support URL encoding (e.g. Java's java.net.URLEncoder) are all encoded according to the rules for the "application/x-www-form-urlencoded" MIME-type. If this encoding method is used, replace the plus signs (+) in the encoded strings with %20, the asterisks (*) with %2A, and change %7E back to the tilde (~) to conform to the encoding rules described above.

(c) Connect the encoded parameter names and values with the English equals sign (\=).

(d) Then, order the parameter name and value pairs connected by equals signs in alphabetical order and connect them with the & symbol to produce the Canonicalized Query String.

Follow the rules below to construct the string used for signature calculation by using the Canonicalized Query String constructed in the previous step:

```
StringToSign=
HTTPMethod + "&" +
percentEncode("/") + "&" +
percentEncode(CanonicalizedQueryString)
```

Here, HTTPMethod is the HTTP method used for request submission, e.g. GET.
percentEncode ("/ ") is the encoded value for the character "/ " according to the URL

encoding rules described in 1.b, namely “%2F”.
percentEncode(CanonicalizedQueryString) is the encoded string of the Canonicalized Query String constructed in Step 1, produced by following the URL encoding rules described in 1.b.

Use the above to calculate the HMAC value of the signature string, as defined in RFC2104.
Note: When calculating the signature, the Key is the Access Key Secret held by the user with the “&” character (ASCII:38) added on the end. The SHA1 hashing algorithm is used.

According to Base64 encoding rules, encode the above HMAC value into a string. This gives you the signature value.

Add the obtained signature value to the request parameters as the ‘Signature’ parameter. This completes the request signing process.
Note: When the obtained signature value is submitted to the server as the final request parameter value, it will undergo URL encoding like other parameters according to RFC3986 rules.

Taking DescribeRegions for example, the request URL before signature is:

```
http://vpc.aliyuncs.com/?TimeStamp=2012-12-26T10:33:56Z&Format=XML&AccessKeyId=testid&Action=DescribeRegions&SignatureMethod=HMAC-SHA1&RegionId=region1&SignatureNonce=NwDAxvLU6tFE0DVb&Version=2014-05-26&SignatureVersion=1.0
```

Thus, the StringToSign is:

```
GET&%2F&AccessKeyId%3Dtestid%26Action%3DDescribeRegions%26Format%3DXML%26RegionId%3Dregion1%26SignatureMethod%3DHMAC-SHA1%26SignatureNonce%3DNwDAxvLU6tFE0DVb%26SignatureVersion%3D1.0%26TimeStamp%3D2012-12-26T10%253A33%253A56Z%26Version%3D2014-05-26
```

If we assume the Access Key ID is “testid”, the Access Key Secret is “testsecret”, and the Key used for HMAC calculation is “testsecret&”, the calculated signature value is:

```
K9fCVP6Jrklpd3rLYKh1pfrrFNo=
```

The signed request URL is (note the added Signature parameter):

```
http://vpc.aliyuncs.com/?TimeStamp=2012-12-26T10%3A33%3A56Z&Format=XML&AccessKeyId=testid&Action=DescribeRegions&SignatureMethod=HMAC-SHA1&RegionId=region1&SignatureNonce=NwDAxvLU6tFE0DVb&Version=2012-09-13&SignatureVersion=1.0&Signature=SDFQNvyH5rtkc9T5Fwo8DOjw5hc%3d
```

Physical Connection Related Interfaces

Apply for Physical Connection access

Description

- When you apply for Physical Connection access, the number of Physical Connections not in **Enabled** state at the same time cannot be greater than five.
- You are not allowed to create more Physical Connection if you have Physical Connection in arrears.
- Created Physical Connections are in Initial state.

Request parameters

Name	Type	Required?	Description
Action	String	Yes	Required parameter, with a value: CreatePhysicalConnection.
RegionId	String	Yes	Region (Region ID for managing instances).
AccessPointId	String	Yes	ID of the access point applied to be connected to the Physical Connection. The selected access point must be connectable.
Type	String	No	Physical Connection type. Default value: VPC; optional value: VPC - VPC leased line.
CircuitCode	String	No	Carrier circuit code corresponding to the leased line interface.
LineOperator	String	Yes	Carrier providing physical lines for access. Optional values:

			- CT - China Telecom - CU - China Unicom - CM - China Mobile - CO - Other (China) - Equinix - Equinix - Other - Other (outside China)
PeerLocation	String	Yes	Physical location of the Physical Connection peer (address). It contains 2 to 256 characters but cannot begin with http:// or https://.
Bandwidth	String	No	Bandwidth of the Physical Connection access interface, in Mbps. Default value: 100; optional values: 2 to 10,000.
PortType	String	No	Type of Physical Connection access port. The default value is 1000Base-T. It can be left empty if the information is unclear. Optional values: - 100Base-T - 100M electrical port - 1000Base-T - 1G electrical port - 1000Base-LX - 1G single-mode optical port (10km) - 10GBase-T - 10G electrical

			port - 10GBase-LR - 10G single-mode optical port (10km)
RedundantPhysicalConnectionId	String	No	Redundant Physical Connection ID. (If a redundant Physical Connection ID is passed in when the second Physical Connection is connected, Alibaba Cloud will allocate to it as independent AD devices as possible to provide higher availability). There must be a corresponding Physical Connection in **Allocated**/**Confirmed**/**Enabled** state.
Name	String	No	Physical Connection name. It contains 2 to 128 characters but cannot begin with http:// or https://.
Description	String	No	Physical Connection description. It contains 2 to 256 characters but cannot begin with http:// or https://.
ClientToken	String	Yes	It is used to ensure the idempotence of the request. The value is generated by a client. It must be unique among all requests and can contain a maximum of 64 ASCII characters.

Return parameters

Name	Type	Description
PhysicalConnectionId	String	ID of the Physical Connection that is approved, in **Initial** state.

Error codes

Error code	Description	HTTP status code	Meaning
InvalidRegionId.NotFound	The RegionId provided does not exist in our records.	404	The specified RegionID does not exist.
InvalidAccessPointId.NotFound	The AccessPointId provided does not exist in our records.	404	The specified AccessPointID does not exist.
InvalidAccessPointId.NotEnabled	The specified AccessPointId is not in Enabled state.	400	The specified AccessPointID is not in Enabled state.
InvalidType.Malformd	The Type provided was invalid.	400	The specified Type value is invalid.
InvalidPeerLocation.Malformd	The PeerLocation provided was invalid.	400	The specified PeerLocation is invalid.
InvalidLineOperator.Malformd	The LineOperator provided was invalid.	400	The specified LineOperator value is invalid.
InvalidPortType.Malformd	The PortType provided was invalid.	400	The specified PortType value is invalid.
InvalidDescription.Malformed	The specified Description is not valid.	400	The specified leased line description is invalid.
InvalidRedundantPhysicalConnectionId.NotFound	The specified RedundantPhysicalConnectionId is not found.	404	The specified redundant leased line does not exist.
InvalidName.Malformed	The specified Name is not valid.	400	The specified leased line name is invalid.
QuotaExceeded.pConnPerAP	Physical connection count per ap quota exceed.	400	The total number of leased lines connected to the access point exceeds the quota limit.
InvalidBandwidth	invalid physical connection banwidth.	400	The specified Bandwidth is invalid.
InvalidRedundantPhysicalConnection	redundant physical connection doesn’t belong to current user.	400	The specified RedundantPhysicalConnection does not belong to the caller.
InvalidRedundantPhysicalConnectionStat	invalid redundant physical connection	400	The specified RedundantPhysicalC

us	status.		onnection status is invalid.
InvalidCircuitCode.Malformed	circuitCode is illegal.	400	The specified CircuitCode is invalid.
QuotaExceeded.freePconnPerAP	free physical connections count excceeded.	400	The number of physical leased lines with no VBR created exceeds the quota limit.

Query the Physical Connection list

Description

Queries the Physical Connection list

- The query result can be filtered through various filters. Currently, the supported filters include PhysicalConnectionID, AccessPointID, Type, LineOperator, Spec, Status, and Name.
- The logical relationship between multiple values of a filter is “or” (the filter criteria are met as long as there is a matching value). The logical relationship between filters is “and” (values are included in the final query result only when they meet the criteria of all filters).

Request parameters

Name	Type	Required?	Description
Action	String	Yes	Required parameter, with a value: DescribePhysicalConnections
RegionId	String	Yes	Region (Region ID for managing instances)
Filter.n.Key	String	No	Type of the nth filter. The n value starts from 1 and is not greater than 5.
Filter.n.Value.m	String	No	The mth value of the nth filter. The m value starts from 1 and is not greater than 5.
PageNumber	Integer	No	Page number of the query result. The

			default value is 1 when the parameter is not specified.Value range: a positive integer. If the PageNumber value is greater than the greatest page number in the actual result, the content of the last page is returned.
PageSize	Integer	No	Page size of the query result. The value range is 1 to 100, and the default value is 10.

Return parameters

The return parameters are all public return parameters. For details, refer to Public Return Parameters.

Name	Type	Description
PhysicalConnectionSet	PhysicalConnectionSetType	Set of PhysicalConnectionSetType values
TotalCount	Integer	Total number of instances
PageNumber	Integer	Page number of the instance list
PageSize	Integer	Number of rows per page set during input

Error codes

Error code	Description	HTTP status code	Meaning
InvalidRegionId.NotFound	The RegionId provided does not exist in our records.	404	The specified RegionID does not exist.
InvalidFilterKey.ValueNotSupported	Specified filter key is not supported: Filter.X.key	404	The specified filter field is not supported.

Modify Physical Conneciton attributes

Description

Modifies Physical Conneciton attributes.

- The Physical Conneciton enters the Initial state from the Rejected state when its attributes are modified.
- Spec and RedundantPhysicalConnectionID can be modified only in Initial/Rejected state.
- No attribute can be modified when a Physical Conneciton is in Canceled, Allocating, AllocationFailed, or Terminated state.

Request parameters

Name	Type	Required?	Description
Action	String	Yes	Required parameter, with a value: ModifyPhysicalConnectionAttribute
RegionId	String	Yes	Region
PhysicalConnectionId	String	Yes	ID of the Physical Conneciton whose attributes need modification
LineOperator	String	No	Carrier providing Physical Conneciton for access. Optional values: - CT - China Telecom - CU - China Unicom - CM - China Mobile - CO - Other (China) - Equinix - Equinix - Other - Other (outside China)
bandwidth	String	No	Bandwidth of the Physical Conneciton access interface, in Mbps. Optional value: \2 to 10,240\

PeerLocation	String	No	Physical location of the Physical Conneciton peer (address)
PortType	String	No	Type of a Physical Conneciton access port. Optional values: - E1 - 2M synchronous electrical port - 100Base-T - 100M electrical port - 1000Base-T - 1G electrical port - 1000Base-LX - 1G single-mode optical port (10km) - 10GBase-T - 10G electrical port - 10GBase-LR - 10G single-mode optical port (10km) - Other - Unstated port
RedundantPhysicalConnectionId	String	No	Redundant Physical Conneciton ID
CircuitCode	String	No	Carrier circuit code corresponding to the Physical Conneciton interface
Name	String	No	Physical Conneciton name
Description	String	No	Physical Conneciton description
ClientToken	String	No	It is used to ensure the idempotence of the request. The value is generated by a client. It must be unique among all

			requests and can contain a maximum of 64 ASCII characters.

Return parameters

The return parameters are all public return parameters. For details, refer to Public Return Parameters.

Error codes

Error code	Description	HTTP status code	Meaning
InvalidRegionId.NotFound	The RegionId provided does not exist in our records.	404	The specified RegionID does not exist.
InvalidPhysicalConnectionId.NotFound	The PhysicalConnectionId provided does not exist in our records.	404	The specified PhysicalConnectionID does not exist.
InvalidLineOperator.Malformd	The LineOperator provided was invalid.	400	The specified leased line carrier is incorrect.
InvalidPeerLocation.Malformd	The PeerLocation provided was invalid.	400	The specified PeerLocation is invalid.
InvalidPortType.Malformd	The PortType provided was invalid.	400	The specified PortType is invalid.
InvalidDescription.Malformed	The specifid Description is not valid.	400	The entered Description is invalid.
InvalidRedundantPhysicalConnectionId.NotFound	The specified RedundantPhysicalConnectionId is not found.	404	The specified redundant Physical Conneciton ID does not exist.
InvalidName.Malformed	The specified Name is not valid.	400	The specified Physical Conneciton name is invalid.
InvalidStatus	invalid physical connection status.	400	The Physical Conneciton status is invalid.
InvalidBandwidth	invalid physical connection banwidth.	400	The Physical Conneciton bandwidth is invalid.
InvalidRedundantPhysicalConnection	redundant physical connection doesn’t belong to current	400	The specified RedundantPhysicalConnection does not

	user.		belong to the caller.
InvalidRedundantPhysicalConnectionStatus	invalid redundant physical connection status.	400	The specified RedundantPhysicalConnection status is invalid.
InvalidCircuitCode.Malformed	circuitCode is illegal.	400	The specified CircuitCode is invalid.

Cancel Physical Connection access

Description

Cancels Physical Connection access before a Chysical Connection is activated.

- Only a Physical Connection in Initial, Approved, Allocated, or Confirmed state (not in Enabled state) can be cancelled.
- The physical leased line enters the Canceled state when cancellation is finished.

Request parameters

Name	Type	Required?	Description
Action	String	Yes	Required parameter, with a value: CancelPhysicalConnection
RegionId	String	Yes	Region
PhysicalConnectionId	String	Yes	Physical leased line ID

Return parameters

The return parameters are all public return parameters. For details, refer to Public Return Parameters.

Error codes

Error code	Description	HTTP status code	Meaning
InvalidRegionId.NotFound	The RegionId provided does not	404	The specified RegionID does not

	exist in our records.		exist.
InvalidPhysicalConnectionId.NotFound	The PhysicalConnectionId provided does not exist in our records.	404	The specified PhysicalConnectionID does not exist.
Forbidden.NotAllowedInState	The request does not allow in this state.	400	The operation is not supported in the current status.

Activate a Physical Connection

Description

- Only a Physical Connection in Confirmed state can be activated.
- The Physical Connection enters the Enabled state when activation is finished.

Request parameters

Name	Type	Required?	Description
Action	String	Yes	Required parameter, with a value: EnablePhysicalConnection
RegionId	String	Yes	Region
PhysicalConnectionId	String	Yes	Physical Connection ID
ClientToken	String	Yes	It is used to ensure the idempotence of the request. The value is generated by a client. It must be unique among all requests and can contain a maximum of 64 ASCII characters.

Return parameters

The return parameters are all public return parameters. For details, refer to Public Return Parameters.

Error codes

Error code	Description	HTTP status code	Meaning
InvalidRegionId.NotFound	The RegionId provided does not exist in our records.	404	The specified RegionID does not exist.
InvalidPhysicalConnectionId.NotFound	The PhysicalConnectionId provided does not exist in our records.	404	The specified Physical Connection does not exist.
Forbidden.NotAllowedInState	The request does not allow in this state.	400	The specified Physical Connection is not in Confirmed state.

Delete a Physical Connection

Description

Deletes a Physical Connection.

- Only a Physical Connection in Rejected, Canceled, AllocationFailed, or Terminated state can be deleted.

Request parameters

Name	Type	Required?	Description
Action	String	Yes	Required parameter, with a value: DeletePhysicalConnection
RegionId	String	Yes	Region (Region ID for managing instances)
PhysicalConnectionId	String	Yes	Physical Connection ID

Return parameters

The return parameters are all public return parameters. For details, refer to Public Return Parameters.

Error codes

Error code	Description	HTTP status code	Meaning
InvalidRegionId.NotFound	The RegionId provided does not exist in our records.	404	The specified RegionID does not exist.
InvalidPhysicalConnectionId.NotFound	The PhysicalConnectionId provided does not exist in our records.	404	The specified Physical Connection does not exist.
Forbidden.NotAllowedInState	The request does not allow in this state.	400	The specified Physical Connection is not in a state for deletion (Rejected, Canceled, AllocationFailed, or Terminated).
Forbidden.VBRExists	physical connection owner’ s vbr still exists.	400	There are still VBRs created for Physical Connection owners under the specified Physical Connection.
Forbidden.VBRExists	vbr owner’ s vbr still exists.	400	There are still VBRs created for other users under the specified Physical Connection.

Terminate Physical Connection access

Description

Terminates Physical Connection access when a Physical Connection is activated.

- Only a Physical Connection in Enabled state can be terminated.
- If there are VBRs whose owners are the Physical Connection owner, the Physical Connection cannot be terminated. (The Physical Connection owner must delete these VBRs.)
- If there are non-terminated VBRs that do not belong to the Physical Connection owner, the Physical Connection cannot be terminated. (The Physical Connection owner must delete VBRs not accepted by other users and terminate VBRs accepted by other users. VBRs accepted by other users cannot be deleted but can be terminated.)
- The Physical Connection first enters the Terminating state and then enters the Terminated

state when termination is finished.

Request parameters

Name	Type	Required?	Description
Action	String	Yes	Required parameter, with a value: TerminatePhysicalConnection
RegionId	String	Yes	Region (Region ID for managing instances)
PhysicalConnectionId	String	Yes	Physical Connection ID

Return parameters

The return parameters are all public return parameters. For details, refer to Public Return Parameters.

Error codes

Error code	Description	HTTP status code	Meaning
InvalidRegionId.NotFound	The RegionId provided does not exist in our records.	404	The specified RegionID does not exist.
InvalidPhysicalConnectionId.NotFound	The PhysicalConnectionId provided does not exist in our records.	404	The specified Physical Connection (PhysicalConnectionID) does not exist.
Forbidden.NotAllowedInState	The request does not allow in this state.	400	The specified Physical Connection is not in Enabled state.
Forbidden.VbrAttached	cannot terminate physical connection when virtual border routers are still attached.	400	There are still associated VBRs on the physical leased line.

Virtual Border Router Related Interfaces

Create a VirtualBorderRouter

Description

Creates a VBR.

- VBRs can be created only by the physical leased line owner for himself/herself or for other users.
- By default, two VBRs can be created on each physical leased line.
    - VBRs created for the physical leased line owner directly enters the Enabled state.
    - VBRs can be created for users other than the physical leased line owner, which enter the Unconfirmed state after creation.
- VlanID can be specified/modified only by the physical leased line owner.
    - A VlanID under a physical leased line cannot be used by two VBRs simultaneously.
    - VlanID of a VBR in Terminated state is automatically retained in seven days and cannot be used by other VBRs only within these seven days.
- LocalGatewayIP, PeerGatewayIP, and PeeringSubnetMask cannot be configured for VBRs created for users other than the physical leased line owner.
- LocalGatewayIP, PeerGatewayIP, and PeeringSubnetMask must be verified. PeeringSubnetMask supports 24-30 bits. PeeringSubnetMask is in dotted decimal notation ranging from 255.255.255.0 to 255.255.255.252. LocalGatewayIP and PeerGatewayIP must be on the same subnet defined by PeeringSubnetMask.For example:
    - LocalGatewayIp: 192.168.50.17
    - PeerGatewayIp: 192.168.50.18
    - PeeringSubnetMask: 255.255.255.248

Request parameters

Name	Type	Required?	Description
Action	String	Yes	Required parameter, with a value: CreateVirtualBorderRouter
RegionId	String	Yes	Region
PhysicalConnectionId	String	Yes	Physical leased line ID
VbrOwnerId	String	No	User ID of the VBR owner. The default value is the ID of the current user (physical leased line owner).

VlanId	Integer	Yes	802.1Q VLAN ID [1-2999] of VBR leased line interface. It can be specified/modified only by the physical leased line owner.
CircuitCode	String	No	Carrier circuit code corresponding to VBR physical leased line. It can be specified/modified only by the physical leased line owner.
LocalGatewayIp	String	No	IP address of VBR leased line interface. It can be specified/modified only by the VBR owner. It is required when VBRs are created for the leased line owner and cannot be entered when VBRs are created for other users.
PeerGatewayIp	String	No	IP address of the peer VBR leased line interface. It can be specified/modified only by the VBR owner. It is required when VBRs are created for the leased line owner and cannot be entered when VBRs are created for other users.
PeeringSubnetMask	String	No	Subnet mask connecting the local and peer VBR leased line interfaces. It can be specified/modified only by the VBR owner. It is required when VBRs are created for the leased line owner and cannot be entered when VBRs are created for other users.
Name	String	No	VBR name, which contains 2 to 128 characters. It cannot begin with http:// or https://. It can be

			specified/modified only by the VBR owner.
Description	String	No	VBR description. It contains 2 to 256 characters and cannot begin with http:// or https://. It can be specified/modified only by the VBR owner.
ClientToken	String	Yes	It is used to ensure the idempotence of the request. The value is generated by a client. It must be unique among all requests and can contain a maximum of 64 ASCII characters.

Return parameters

Name	Type	Description
VbrId	String	VBR ID

Error codes

Error code	Description	HTTP status code	Meaning
InvalidRegionId.NotFound	The specified RegionId is not found.	404	The specified RegionID does not exist.
InvalidPhysicalConnectionId.NotFound	The specified PhysicalConnectionId is not found.	400	The specified physical leased line ID does not exist.
InvalidVlanId.Used	The specified VlanId has been used.	400	The specified VlanID is used.
MissingParameter	The input parameter ‘PhysicalConnectionId’ that is mandatory for processing this request is not supplied.	400	The PhysicalConnectionID value is missing.
InvalidPhysicalConnectionId.NotEnabled	The specified PhysicalConnectionId is not in Enabled	400	The specified physical leased line is not in normal

	state.		state currently.
InvalidVbrOwnerId.NotFound	The specified VbrOwnerId is not valid.	404	The specified VBROwnerID does not exist.
MissingParameter	The input parameter 'VlanId' that is mandatory for processing this request is not supplied.	400	The VlanID value is missing.
InvalidVlanId.Malformed	The specified VlanId is not valid.	400	The specified VlanID format is invalid.
InvalidCircuitCode.Malformed	The specified CircuitCode is not valid.	400	The specified CircuitCode format is invalid.
MissingParameter	The input parameter 'LocalGatewayIp' that is mandatory for processing this request is not supplied.	400	The LocalGatewayIP value is missing.
InvalidLocalGatewayIp.Malformed	The specified LocalGatewayIp is not valid.	400	The specified LocalGatewayIP format is invalid.
MissingParameter	The input parameter 'PeerGatewayIp' that is mandatory for processing this request is not supplied.	400	The PeerGatewayIP value is missing.
InvalidPeerGatewayIp.Malformed	The specified PeerGatewayIp is not valid.	400	The specified PeerGatewayIP format is invalid.
MissingParameter	The input parameter 'PeeringSubnetMask' that is mandatory for processing this request is not supplied.	400	The PeerGatewayIP value is missing.
InvalidPeeringSubnetMask.Malformed	The specified PeeringSubnetMask is not valid.	400	The specified PeerGatewayIP format is invalid.
Forbidden.LocalGatewayIpNotAllowedByCaller	The caller is not allowed to specify the LocalGatewayIp parameter.	403	The caller is not allowed to specify the LocalGatewayIP parameter.
Forbidden.PeerGatewayIpNotAllowedBy	The caller is not allowed to specify	403	The caller is not allowed to specify

Caller	the PeerGatewayIp parameter.		the PeerGatewayIP parameter.
Forbidden.PeeringSubnetMaskNotAllowedByCaller	The caller is not allowed to specify the PeeringSubnetMask parameter.	403	The caller is not allowed to specify the PeeringSubnetMask parameter.
Forbidden.NameNotAllowedByCaller	The caller is not allowed to specify the Name parameter.	403	The caller is not allowed to specify the Name parameter.
Forbidden.DescriptionNotAllowedByCaller	The caller is not allowed to specify the Description parameter.	403	The caller is not allowed to specify the Description parameter.
InvalidName.Malformed	The specified ‘Name’ is not valid.	403	The specified Name format is invalid.
InvalidDescription.Malformed	The specified ‘Description’ is not valid.	400	The specified Description format is invalid.
QuotaExceeded.vbrPerpConn	Virtual boarder router per PhysicalConnection quota exceed.	400	The number of VBRs created on the specified physical leased line exceeds the quota limit.
QuotaExceeded.freevbr	Free virtual boarder router quota exceed.	400	The number of VBRs exceeds the quota limit.
InvalidIP.Malformed	Ip malformed.	400	The IP address format is invalid.
MissingParameter	LocalGatewayIp, peerGatewayIp and peeringSubnetMask must not be null.	400	The value of LocalGatewayIP, PeerGatewayIP, or PeeringSubnetMask is missing.
InvalidIp.NotSameSubnet	Local gateway ip and peer gateway ip are not in the same subnet.	400	LocalGatewayIP and PeerGatewayIP are not on the same subnet.

Delete a VBR

Description

Deletes a VBR.

- A VBR with RouterInterfaces (excluding those on the user client) or routes on it cannot be deleted.
- The VBR owner is allowed to delete a VBR in Unconfirmed, Enabled, or Terminated state.
- The physical leased line owner is allowed to delete VBRs only in Unconfirmed state. A VBR in Active state (accepted by a VBR owner for use) can be deleted only by the VBR owner.

Request parameters

Name	Type	Required?	Description
Action	String	Yes	Required parameter, with a value: DeleteVirtualBorderRouter
RegionId	String	Yes	Region
VbrId	String	Yes	VBR ID

Return parameters

The return parameters are all public return parameters. For details, refer to Public Return Parameters.

Error codes

Error code	Description	HTTP status code	Meaning
InvalidRegionId.NotFound	The specified RegionId is not found.	404	The specified RegionID does not exist.
InvalidVbrId.NotFound	TThe specified VirutalBorderRouter does not exist in our records.	404	The specified VBR does not exist.
InvalidOperation.RouterInterfaceNotDeleted	The specified VirutalBorderRouter still has routerInterface.	400	The specified VBR still has a RouterInterface.
InvalidOperation.OperationNotAllowedInState	The specified VirutalBorderRouter is in invalid state.	400	The specified VBR cannot be deleted.
Forbidden.OperationNotAllowedByUser	The caller is not allowed to delete	403	The caller is not allowed to delete

	the specified VirtualBorderRouter.		the specified VBR.

Query the VBR list

Description

- Queries the list of VBRs whose owner is the caller.
- The query result can be filtered through various filters. Currently, the supported filters include VbrID, PhysicalConnectionID, Status, and Name.
- The logical relationship between multiple values of a filter is “or” (the filter criteria are met as long as there is a matching value). The logical relationship between filters is “and” (values are included in the final query result only when they meet the criteria of all filters).

Request parameters

Name	Type	Required?	Description
Action	String	Yes	Required parameter, with a value: DescribeVirtualBorderRouters
RegionId	String	Yes	Region
Filter.n.Key	String	No	Type of the nth filter. The n value starts from 1 and is not greater than 5.
Filter.n.Value.m	String	No	The mth value of the nth filter. The m value starts from 1 and is not greater than 5.
PageNumber	Integer	No	Page number of the query result. The default value is 1 when the parameter is not specified.Value range: a positive integer. If the PageNumber value is greater than the greatest page number in the actual result, the content of the last page is returned.

PageSize	Integer	No	Page size of the query result. The value range is [1,100], and the default value is 10.

Return parameters

The return parameters are all public return parameters. For details, refer to Public return parameters.

Name	Type	Description
VirtualBorderRouterSet	VirtualBorderRouterSetType	Set of VirtualBorderRouterTypes
TotalCount	Integer	Total number of instances
PageNumber	Integer	Page number of the instance list
PageSize	Integer	Number of rows per page set during input

Error codes

Error code	Description	HTTP status code	Meaning
InvalidRegionId.NotFound	The RegionId provided does not exist in our records.	404	The specified RegionID does not exist.
InvalidFilterKey.ValueNotSupported	Specified filter key is not supported: Filter.X.key	404	The specified filter field is not supported.

View the list of VBRs under a physical leased line

Description

Views the list of VBRs under a physical leased line.

- The physical leased line owner calls this interface to query the VBR list created based on a specified physical leased line of him/her.
- The query result may include VBRs whose owner is the caller and VBRs whose owner is not the caller.

- When this interface is called, fields that only the VBR owner can view are not returned.

Request parameters

Name	Type	Required?	Description
Action	String	Yes	Required parameter, with a value: DescribeVirtualBorderRoutersForPhysicalConnection
RegionId	String	Yes	Region
PhysicalConnectionId	String	Yes	Physical leased line ID
PageNumber	Integer	No	Page number of the query result. The default value is 1 when the parameter is not specified.Value range: a positive integer. If the PageNumber value is greater than the greatest page number in the actual result, the content of the last page is returned.
PageSize	Integer	No	Page size of the query result. The value range is [1,100], and the default value is 10.

Return parameters

The return parameters are all public return parameters. For details, refer to Public return parameters.

Name	Type	Description
VirtualBorderRouterForPhysicalConnectionSet	VirtualBorderRouterForPhysicalConnectionSetType	Set of VirtualBorderRouterForPhysicalConnectionTypes.
TotalCount	Integer	Total number of instances
PageNumber	Integer	Page number of the instance list
PageSize	Integer	Number of rows per page set during input

Error codes

Error code	Description	HTTP status code	Meaning
InvalidRegionId.NotFound	The RegionId provided does not exist in our records.	404	The specified RegionID does not exist.

Modify VBR attributes

Description

Modifies VBR attributes.

- VlanID can be specified/modified only by the physical leased line owner.
  - A VlanID under a physical leased line cannot be used by two VBRs simultaneously.
  - VlanID of a VBR in Terminated state is automatically retained in seven days and cannot be used by other VBRs only within these seven days.
- LocalGatewayIP, PeerGatewayIP, and PeeringSubnetMask cannot be configured for VBRs created for users other than the physical leased line owner.
- LocalGatewayIP, PeerGatewayIP, and PeeringSubnetMask must be verified. PeeringSubnetMask supports 24-30 bits. PeeringSubnetMask is in dotted decimal notation ranging from 255.255.255.0 to 255.255.255.252. LocalGatewayIP and PeerGatewayIP must be on the same subnet defined by PeeringSubnetMask.For example:
  - LocalGatewayIp: 192.168.50.17
  - PeerGatewayIp: 192.168.50.18
  - PeeringSubnetMask: 255.255.255.248

Request parameters

Name	Type	Required?	Description
Action	String	Yes	Required parameter, with a value: ModifyVirtualBorderRouterAttribute
RegionId	String	Yes	Region (Region ID for managing instances)
VBRId	String	Yes	VBR ID
VlanId	String	No	802.1Q VLAN ID (1-2,999) of VBR leased line interface.

CircuitCode	String	No	Carrier circuit code corresponding to VBR physical leased line. It can be specified/modified only by the physical leased line owner.
LocalGatewayIp	String	No	IP address of the local VBR leased line interface. It can be specified/modified only by the VBR owner. You are not allowed to change this attribute of VBRs in Enabled state to empty.
PeerGatewayIp	String	No	IP address of the peer VBR leased line interface. It can be specified/modified only by the VBR owner. You are not allowed to change this attribute of VBRs in Enabled state to empty.
PeeringSubnetMask	String	No	Subnet mask connecting the local and peer VBR leased line interfaces. It can be specified/modified only by the VBR owner. You are not allowed to change this attribute of VBRs in Enabled state to empty.
VBRName	String	No	VBR name
VBRDescription	String	No	VBR description

Return parameters

The return parameters are all public return parameters. For details, refer to Public Return Parameters.

Error codes

Error code	Description	HTTP status code	Meaning
InvalidRegionId.Not	The specified	404	The specified

Found	RegionId is not found.		RegionID does not exist.
InvalidVbrId.NotFound	The specified VirutalBorderRouter is not found.	400	The specified VbrID does not exist.
InvalidVlanId.Used	The specified VlanId has been used.	400	The specified VlanID is used.
InvalidCircuitCode.Malformed	The specified CircuitCode is not valid.	400	The specified CircuitCode is invalid.
InvalidVlanId.Malformed	The specified VlanId is not valid.	400	The specified VlanID is invalid.
InvalidIp.Malformed	The specified ip address is not valid.	400	The specified IP address is invalid.
InvalidPeeringSubnetMask.Malformed	The specified PeeringSubnetMask is not valid.	400	The specified PeeringSubnetMask is invalid.
Forbidden.CircuitCodeNotAllowedByCaller	The caller is not allowed to modify.	403	
Forbidden.CircuitCodeNotAllowedByCaller	The caller is not allowed to specify the CircuitCode parameter.	403	The caller is not allowed to specify the CircuitCode parameter.
Forbidden.LocalGatewayIpNotAllowedByCaller	The caller is not allowed to specify the LocalGatewayIp parameter.	403	The caller is not allowed to specify the LocalGatewayIP parameter.
Forbidden.PeerGatewayIpNotAllowedByCaller	The caller is not allowed to specify the PeerGatewayIp parameter.	403	The caller is not allowed to specify the PeerGatewayIP parameter.
Forbidden.PeeringSubnetMaskNotAllowedByCaller	The caller is not allowed to specify the PeeringSubnetMask parameter.	403	The caller is not allowed to specify the PeeringSubnetMask parameter.
Forbidden.VlanIdNotAllowedByCaller	The caller is not allowed to specify the VlanId.	403	The caller is not allowed to specify the VlanID parameter.
Forbidden.NameNotAllowedByCaller	The caller is not allowed to specify the Name parameter.	403	The caller is not allowed to specify the Name parameter.
Forbidden.DescriptionNotAllowedByCalle	The caller is not allowed to specify	403	The caller is not allowed to specify

r	the Description parameter.		the Description parameter.
InvalidName.Malformed	The specified 'Name' is not valid.	400	The specified VBR name is invalid.
InvalidDescription.Malformed	The specifid 'Description' is not valid.	400	The specified description information is invalid.
MissingParameter	LocalGatewayIp, peerGatewayIp and peeringSubnetMask must not be null.	400	The value of LocalGatewayIP, PeerGatewayIP, or PeeringSubnetMask is missing.
InvalidIp.NotSameSubnet	Local gateway ip and peer gateway ip are not in the same subnet.	400	LocalGatewayIP and PeerGatewayIP are not on the same subnet.

Restore a VBR

Description

Restores a VBR.

- Only the physical leased line owner can call this interface to restore a VBR for it to use a leased line.
- Only a VBR in Terminated state can be restored.
- Only a VBR whose physical leased line is in Enabled state can be restored.
- The VBR first enters the Recovering state and then enters the Enabled state when restoration is finished.
- VlanID must be verified:
    - If VlanID is not occupied by other VBRs (in Terminated state within seven days), the VBR can be restored successfully.
    - If VlanID is occupied by other VBRs (in Terminated state after seven days and reused by the physical leased line owner on other VBRs), the VRB cannot be restored successfully.

Request parameters

Name	Type	Required?	Description
Action	String	Yes	Required parameter,

			with a value: RecoverVirtualBorderRouter
RegionId	String	Yes	Region
VbrId	String	Yes	VBR ID

Return parameters

The return parameters are all public return parameters. For details, refer to Public Return Parameters.

Error codes

Error code	Description	HTTP status code	Meaning
InvalidRegionId.NotFound	The RegionId provided does not exist in our records.	404	The specified RegionID does not exist.
InvalidVbrId.NotFound	The specified VirutalBorderRouter is not found.	404	The specified VirutalBorderRouter does not exist.
InvalidOperation.OperationNotAllowedInState	The specified VirutalBorderRouter is in invalid state.	400	The specified VirutalBorderRouter is invalid.
OperationFailed.VlanIdAlreadyInUse	The specifed vlanId has been used.	400	The specified VlanID is occupied.
Forbidden.OperationNotAllowedByUser	The caller is not allowed to recover the specified VirtualBorderRouter.	403	The caller is not allowed to restore the specified VBR.

Terminate a VBR

Description

Terminates a VBR.

- Only the physical leased line owner can call this interface to stop a VBR from using a leased line without deleting the VBR and other content (including RouterInterfaces and routes) on the VBR.
- Only a VBR in Enabled state can be terminated.

- The VBR first enters the Terminating state and then enters the Terminated state when termination is finished.
- VlanID of a VBR in Terminated state is automatically retained in seven days and cannot be used by other VBRs only within these seven days.
- To prevent misoperation, you are allowed to restore a VBR for it to use a leased line. If VlanID is not occupied by other VBRs (in Terminated state within seven days), the VBR can be restored successfully.

Request parameters

Name	Type	Required?	Description
Action	String	Yes	Required parameter, with a value: TerminateVirtualBorderRouter
RegionId	String	Yes	Region
VbrId	String	Yes	VBR ID

Return parameters

The return parameters are all public return parameters. For details, refer to Public Return Parameters.

Error codes

Error code	Description	HTTP status code	Meaning
InvalidRegionId.NotFound	The RegionId provided does not exist in our records.	404	The specified RegionID does not exist.
InvalidVbrId.NotFound	The specified VirutalBorderRouter is not found.	404	The specified VbrID does not exist.
InvalidOperation.OperationNotAllowedInState	The specified VirutalBorderRouter is in invalid state.	400	The specified VirutalBorderRouter is invalid.
Forbidden.OperationNotAllowedByUser	The caller is not allowed to terminate the specified VirtualBorderRouter.	403	The caller is not allowed to terminate the specified VBR.

Router Interface Related Interfaces

Set the RouterInterface status to Active

Description

Sets the status of an inactive RouterInterface to Active.

- This is an asynchronous operation. A successful return of the API call indicates that the RouterInterface enters the Activating state. You can perform round-robin to know whether the RouterInterface enters the Active state.
- This operation is not allowed if the RouterInterface is in arrears.

Request parameters

Name	Type	Required?	Description
Action	string	Yes	Required parameter, with a value: ActivateRouterInterface
RegionId	string	Yes	Region of the instance
RouterInterfaceId	string	Yes	RouterInterface ID

Return parameters

The return parameters are all public return parameters. For details, refer to Public Return Parameters.

Error codes

Error code	Description	HTTP status code	Meaning
InvalidRegionId.NotFound	The specified RegionId does not exist in our records.	404	The specified RegionID does not exist.
InvalidRouterInterfaceId.NotFound	The specified RouterInterfaceId does not exist in our records.	404	The specified RouterInterfaceID does not exist.

IncorrectStatus	RouterInterface can be operated by this action only when its status is Inactive.	400	This operation is allowed only when the RouterInterface is in “Inactive” state.
Forbidden.FinacialLocked	This RouterInterface is finacial locked because of bills outstanding.	400	The instance is locked in arrears.

Initiate a RouterInterface connection

Description

Initiates an open RouterInterface connection request to connect a local RouterInterface to a peer RouterInterface.

NOTE:

- Only the RouterInterface whose role is InitiatingSide can be used to perform this operation. The RouterInterface whose role is AcceptingSide can only be used to configure the connection peer information and wait to be connected.
- The connection operation can be successful only when an AcceptingSide RouterInterface sets the InitiatingSide RouterInterface as the connection peer.
- This is an asynchronous operation. A successful return of the API call indicates that the connection begins to be set up, and the InitiatingSide RouterInterface enters the Connecting state (the AcceptingSide RouterInterface enters the AcceptingConnect state). When the InitiatingSide RouterInterface enters the Active state, the connection is eventually set up.
- The RouterInterface connection means a connection of two routers through a virtual “link”. To transfer messages properly on the link, you need to configure the route table correctly for routers on both sides and configure access control rules correctly for objects in actual communication.

Restrictions for the operation:

- A RouterInterface can be connected to only one RouterInterface.
- A RouterInterface cannot be connected to itself.
- A RouterInterface cannot be connected to another RouterInterface on the same router.
- Only one pair of connected RouterInterfaces is allowed between any two routers.
- This operation can be performed only when RouterInterfaces are in Idle state.
- You are not allowed to perform this operation if you have RouterInterfaces in arrears (that is, you have unpaid RouterInterface bills).

Request parameters

Name	Type	Required?	Description
Action	string	Yes	Required parameter, with a value: ConnectRouterInterface
RegionId	string	Yes	Region.
RouterInterfaceId	string	Yes	ID of the RouterInterface to initiate a connection.

Return parameters

The return parameters are all public return parameters. For details, refer to Public Parameters.

Error codes

Error code	Description	HTTP status code	Meaning
InvalidRegionId.NotFound	The specified RegionId does not exist in our record.	404	The specified RegionID does not exist.
InvalidRouterInterfaceId.NotFound	The specified RouterInterfaceId does not exist in our record.	404	The specified RouterInterfaceID does not exist.
IncorrectRole.NotInitiatingSide	The spedified RouterInterface is not InitiatingSide	400	The specified RouterInterface is not on the InitiatingSide.
IncorrectStatus	RouterInterface can be operated by this action only when its status is Idle.	400	This operation can be performed only when RouterInterfaces are in “Idle” state.
IncorrectOppositeInterfaceInfo.NotSet	OppositeInterfaceId/OppositeRouterId/OppositeRouterType/OppositeRouterOwnerId must be all set.	400	The connection peer information is not configured.
IncorrectOppositeInterfaceInfo	The opposite RouterInterface information specified by OppositeInterfaceId/	400	The peer RouterInterface information is incorrect. Maybe the peer

	OppositeRouterId/OppositeRouterType/OppositeRouterOwnerId is wrong. Maybe the RouterInterface or Router does not exist, or the parameters mismatched, or that RouterInterface is not a AcceptingSide.		RouterInterfaceID/RouterID does not exist, maybe parameter values do not match, or maybe the peer RouterInterface is not an AcceptingSide RouterInterface.
RejectedByOppositeInterface	The opposite RouterInterface does not set this RouterInterface as a OppositeRouterInterface. Meybe it was wrongly set. Please check the configuration of the opposite RouterInterface.	403	The specified RouterInterface is not set as the connection peer on the peer RouterInterface. Maybe the peer information is not configured correctly on the peer RouterInterface. You need to check the configurations on the peer RouterInterface.
Forbidden.BillsOutstanding	You cannot use this action because you have bills outstanding.	400	You are not allowed to perform this operation because you have unpaid bills.

Set the RouterInterface status to Inactive

Description

Sets the status of an active RouterInterface to Inactive.

- This is an asynchronous operation. A successful return of the API call indicates that the RouterInterface enters the Deactivating state. You can perform round-robin to know whether the RouterInterface enters the Inactive state.
- This operation is not allowed if the RouterInterface is in arrears.

Request parameters

Name	Type	Required?	Description

Action	string	Yes	Required parameter, with a value: DeactivateRouterInterface
RegionId	string	Yes	Region of the instance
RouterInterfaceId	string	Yes	RouterInterface ID

Return parameters

The return parameters are all public return parameters. For details, refer to Public Return Parameters.

Error codes

Error code	Description	HTTP status code	Meaning
InvalidRegionId.NotFound	The specified RegionId does not exist in our records.	404	The specified RegionID does not exist.
InvalidRouterInterfaceId.NotFound	The specified RouterInterfaceId does not exist in our records.	404	The specified RouterInterfaceID does not exist.
IncorrectStatus	RouterInterface can be operated by this action only when its status is Active.	400	This operation can be performed only when the RouterInterface is in “Active” state.
Forbidden.FinacialLocked	This RouterInterface is finacial locked because of bills outstanding.	400	The instance is locked in arrears.

Delete a RouterInterface

Description

Deletes a RouterInterface.

- This operation is allowed only when a RouterInterface is in Idle or Inactive state.
- Before RouterInterface deletion, custom routes directed to it must be all deleted in advance.

Request parameters

Name	Type	Required?	Description
Action	string	Yes	Required parameter, with a value: DeleteRouterInterface
RegionId	string	Yes	
RouterInterfaceId	string	Yes	RouterInterface ID

Return parameters

The return parameters are all public return parameters. For details, refer to Public Return Parameters.

Error codes

Error code	Description	HTTP status code	Meaning
InvalidRegionId.NotFound		404	The specified RegionID does not exist.
IncorrectStatus	This operation would be allowed only when status of this RouterInterface is Idle/Inactive.	400	This operation is allowed only when the RouterInterface is in "Idle" or "Inactive" state.
DependencyViolation.RouterInterfaceReferredByRouteEntry	RouterInterface cannot be deleted when it is referred by a route table entry as next hop.	400	The RouterInterface cannot be deleted when it is referenced by a route entry as the next hop.
InvalidInstanceId.NotFound	The InstanceId provided does not exist in our records.	404	The specified instance does not exist.

Query the RouterInterface list

Description

Queries the RouterInterface list.

The query result can be filtered through various filters. Currently, the supported filters include:

- RouterInterfaceId
- RouterId
- RouterType
- RouterInterfaceOwnerId
- OppositeInterfaceId
- OppositeRouterType
- OppositeRouterId
- OppositeInterfaceOwnerId
- Status
- Name

The logical relationship between multiple values of a filter is "or" (the filter criteria are met as long as a value is fit in the query result).

The logical relationship between filters is "and" (values are included in the final query result only when they meet the criteria of all filters).

Request parameters

Name	Type	Required?	Description
Action	string	Yes	Required parameter, with a value: DescribeRouterInterfaces
RegionId	string	Yes	Region of the instance
Filter.n.Key	string	No	Type of the nth filter. The n value starts from 1 and is not greater than 5.
Filter.n.Value.m	string	No	The mth value of the nth filter. The m value starts from 1 and is not greater than 5.
PageNumber	int	No	Page number of the query result. It is a positive integer, and the default value is 1. If the PageNumber value is greater than the greatest page number in the actual result, the content of

			the last page is returned.
PageSize	int	No	Page size of the query result. The value range is 1 to 100, and the default value is 10.

Return parameters

Name	Type	Description
TotalCount	Integer	Total number of instances
PageNumber	Integer	Page number of the instance list
PageSize	Integer	Number of rows per page set during input
Instances	RouterInterfaceSetType	Returned instance information, in array format of RouterInterfaceSetType

Error codes

Error code	Description	HTTP status code	Meaning
InvalidRegionId.NotFound	The RegionId provided does not exist in our records.	404	The specified RegionID does not exist.
InvalidOppositeRegionId.NotFound	The oppositeRegionId provided does not exist in our records.	404	The specified peer RegionID does not exist.
InvalidFilterKey.ValueNotSupported	Specified filter key is not supported: Filter.X.key	404	The specified filter type is not supported.

Modify RouterInterface attributes

Description

Modifies Name/Description/connection peer information of a RouterInterface.

- Connection peer information

(OppositeInterfaceID/OppositeRouterID/OppositeInterfaceOwnerID) can be modified only when the RouterInterface is in Idle state.

Request parameters

Name	Type	Required?	Description
Action	String	Yes	Required parameter, with a value: ModifyRouterInterfaceAttribute
RegionId	String	Yes	
RouterInterfaceId	String	Yes	RouterInterface ID
Name	String	No	Custom name. It is a string of 2 to 128 Chinese or English characters. It must begin with an uppercase/lowercase letter or a Chinese character and can contain numbers, ".", "_", or "-". It cannot begin with http:// or https://.
Description	String	No	Custom description. It is a string of 2 to 256 characters. The instance description is displayed on Alibaba Cloud console. It cannot begin with http:// or https://.
OppositeInterfaceId	String	No	Interface ID of the connection peer
OppositeRouterId	String	No	Router ID of the connection peer interface. When OppositeRouterType is VBR, the VBR specified by OppositeRouterID must be in the access point specified by OppositeAccessPointID.During RouterInterface creation/modification, OppositeRouterID and OppositeAccessPointID are not verified and

			an error will occur during the connection.
OppositeInterfaceOwnerId	String	No	Owner account ID of the connection peer RouterInterface. You can log on to Alibaba Cloud official website and choose Management Console > User Center > Account Management to view your account ID.
HealthCheckSourceIp	String	No	Source IP address of the packet for leased line HealthCheck in leased line disaster tolerance and ECMP scenarios. It is valid only for a VRouter RouterInterface with a peer on a VBR. The source IP address must be in the VPC of the local VRouter and is not used. HealthCheckSourceIP and HealthCheckTargetIP parameters must be both specified or left unspecified.
HealthCheckTargetIp	String	No	Target IP address of the packet for leased line HealthCheck in leased line disaster tolerance and ECMP scenarios. It is valid only for a VRouter RouterInterface with a peer on a VBR. Usually you can use the CPE IP address of the leased line user’s client (that is, the PeerGatewayIP on the VBR of the peer RouterInterface), you can also specify another IP address of the leased line user’s client as the HealthCheck target IP address. HealthCheckSourceIP

			and HealthCheckTargetIP parameters must be both specified or left unspecified.

Return parameters

The return parameters are all public return parameters. For details, refer to Public Return Parameters.

Error codes

Error code	Description	HTTP status code	Meaning
InvalidRegionId.NotFound	The specified RegionId does not exist in our records.	404	The specified RegionID does not exist.
InvalidRouterInterfaceId.NotFound	The specified RouterInterfaceId does not exist in our records.	404	The specified RouterInterfaceID does not exist.
InvalidName.Malformd	The specified ‘Name’ is not valid.	400	The specified Name is invalid.
InvalidDescription.Malformed	The specifid ‘Description’ is not valid.	400	The specified Description is invalid.
InvalidOppositeRouterType.ValueNotSupported	The specified OppositeRouterType is not valid.	400	The specified OppositeRouterType is invalid.
IncorrectStatus	RouterInterface can be operated by this action only when the status is Active,idle or inactive.	400	This operation is supported when the RouterInterface status is “Active”, “Idle” or “Inactive”.
Forbidden.FinacialLocked	This RouterInterface is finacial locked because of bills outstanding.	400	The RouterInterface is locked in arrears.
Forbbiden	The Router instance owener error	400	The router does not belong to the current caller.
LinkRole.NotSupport	This linkrole is not supported	400	The connection role is not supported.
Forbbiden.ModifyIdAndType	Opposite is VBR, cannot modify the	400	The peer is a VBR and the ID and type

	ID and type		cannot be edited.
InvalidParam.Modify RouterInterface	Modify routerinterface param invalide	400	The parameter for editing a RouterInterface is invalid.

Route Table Related Interfaces

Create a route entry

Description

Creates a RouteEntry.

You can create a RouteEntry in a route table on a VPC VRouter or on a VBR on ExpressConnect.

You can create a RouteEntry in a route table on a VPC VRouter:

- The number of custom route entries in a single route table cannot exceed 48.
- The RouteEntry DestinationCidrBlock cannot be identical to a VSwitch CIDRBlock in the same VPC. It cannot contain or be contained by a VSwitch CIDRBlock, either.
- The RouteEntry DestCidrBlock network segment cannot direct to 100.64.0.0/10 or be contained by 100.64.0.0/10.
- Identical RouteEntry DestCidrBlocks cannot exist under the same route table.
- If the specified DestinationCidrBlock is an IP address, it will be processed based on a 32-bit mask.
- Multiple RouteEntries can direct to the same InstanceID.
- The next hop InstanceID specified by a RouteEntry must belong to the VPC.
- Each VPC can have up to five route entries directing to a HaVip simultaneously.
- The ECMP route can be configured through the NextHopList parameter.

You can create a RouteEntry in a route table on a VBR.

- The number of custom route entries in a single route table cannot exceed 48.
- The NextHopList parameter is not supported.
- The RouteEntry DestCidrBlock network segment cannot direct to 100.64.0.0/10 or be contained by 100.64.0.0/10.
- Identical RouteEntry DestCidrBlocks cannot exist under the same route table.
- If the specified DestinationCidrBlock is an IP address, it will be processed based on a 32-bit

mask.
- Multiple RouteEntries can direct to the same InstanceID.
- The next hop InstanceID specified by a RouteEntry must belong to the VBR.
- Only RouterInterface is supported as the next hop type of RouteEntry.
- A RouteEntry can be created only when the VBR is in Active state and the corresponding physical leased line is in Enabled state and not locked in arrears.

Request Parameters

Name	Type	Required?	Description
Action	String	Yes	Required parameter, with a value:CreateRouteEntry
RouteTableId	String	Yes	RouteTableId
DestinationCidrBlock	String	Yes	The RouteEntry’s target network segment
NextHopType	String	No	The next hop type. Available value options: Instance | Tunnel. The default value is Instance
NextHopId	String	No	The route entry’s next hop
NextHopList	NextHopListType	No	List of route entry’s next hop. If this parameter is appointed, this route entry is an ECMP route entry and the NextHopList contains 2 - 4 the next hop of ECMP route entry. If this parameter is appointed, NextHopType and NextHopId must not be appointed. This parameter can only be appointed while the route entry belongs to a VRouter, and the next hop of ECMP must be RouterInterface which oppsite router type is ‘VBR’.If this parameter is not appointed,

			NextHopType and NextHopId must be appointed.
ClientToken	String	No	Used to ensure the idempotence of the request. The value is generated by a client. It must be unique among all requests and contains a maximum of 64 ASCII characters. For details, refer to the appendix: How to Ensure Idempotence.

Return Parameters

All are public return parameters. For details, see Public Return Parameters

Error Code

Error Code	Description	Http status code	Meaning
InvalidCidrBlock.Malformed	Specified CIDR block is not valid.	400	The specified network segment is invalid
InvalidInstanceId.NotFound	Specified instance does not exist.	404	The specified ECS instance does not exist (the instance does not belong tp this VPC)
InvalidRouteTableId.NotFound	Specified route table does not exist.	404	The specified virtual route table does not exist
InvalidCIDRBlock.Duplicate	Specified CIDR block already exists.	400	The specified network segment already exists in the route table
IncorrectVpcStatus	Current VPC status does not support this operation.	400	Creation of the specified VPC has not been completed
QuotaExceeded	Route entry quota exceeded in this route table.	400	The number of custom route entries in a single route table cannot exceed 48
IncorretRouteEntryStatus	Some route entry status blocked this	400	If any route entry or VSwitch in a route

	operation.		table under the same VPC is in an intermediate status (adding, modifying, deleting), a route entry cannot be created
InvalidCidrBlock	Specified CIDR block is not valid.	400	1. This user is not in the white list for the 10.0.0.0/8 route segment and cannot use this segment. 2. The added custom route’ s target segment cannot belong to the network segment for all VSwitches under the same VPC. 3. The network segment cannot be: 100.64.0.0 / 10.
InvalidNextHop.Not Found	Specified next hop does not exist.	404	The specified next hop does not exist

Delete a custom route

Description

Delete the specified RouteEntry

- You can only delete a RouteEntry in the ‘Available’ state
- If the VPC of the RouteTable is creating/deleting a VSwitch or creating/deleting a RouteEntry, you cannot perform the delete RouteEntry operation

Request Parameters

Name	Type	Required?	Description
Action	String	Yes	Required parameter, with a value:DeleteRouteEntry
RouteTableId	String	Yes	The RouteEntry’ s RouteTableId
DestinationCidrBloc	String	Yes	The RouteEntry’ s

k			DestCidrBlock
NextHopId	String	Yes	The RouteEntry’s next hop ID

Return Parameters

All are public return parameters. For details, see Public Return Parameters

Error Code

Error Code	Description	Http status code	Meaning
IncorretRouteEntryStatus	Some route entry status blocked this operation.	400	1. If any route entry in a route table under the same VPC is in an intermediate status (adding, modifying, deleting), a route entry cannot be deleted. 2. If a VSwitch under the same VPC is in an intermediate status (creating, deleting), a route entry cannot be deleted.
InvalidCidrBlock.Malformed	Specified CIDR block is not valid.	400	Incorrect CidrBlock input format
InvalidRouteTableId.NotFound	Specified route table does not exist.	404	The RouteTableId does not exist
OperationDenied	Specified operation is denied as route entry type is system.	400	Cannot delete a System route entry

Query the RouteEntry list

Description

Query the list of route tables under the user name. This interface supports querying by page, with 10 lines per page by default. This interface can only verify the legitimacy of the parameters and cannot verify the dependency between parameters. The returned results are the “intersection” of all

criteria.

Request Parameters

Name	Type	Required?	Description
Action	String	Yes	Required parameter, with a value:DescribeRouteTables
VRouterId	String	Yes	VRouterId
RouteTableId	String	No	RouteTableId
PageNumber	Integer	No	Page number of the instance status list. The start value is 1. The default value is 1
PageSize	Integer	No	The number of lines per page set for paging query. The maximum value is 50 and default value is 10

Return Parameters

Name	Type	Description
RouteTables	RouteTableSetType	RouteTable details, RouteTableSetType collection
TotalCount	Integer	The number of items on the list
PageNumber	Integer	The current page
PageSize	Integer	The number of items on the current page

Error Code

Refer to Public Error Codes.

Region Related Interfaces

Query the Region list

Description

Query the supported Region list

Request Parameters

Name	Type	Required?	Description
Action	String	Yes	Required parameter, with a value:DescribeRegions

Return Parameters

Name	Type	Description
Regions	RegionType	A collection composed of region information RegionType

Error Code

Refer to Public Error Codes.

Querying a zone

Description

Request Parameters

Name	Type	Required?	Description
Action	String	Yes	Required parameter, with a value:DescribeZones

RegionId	String	Yes	Destination RegionId

Return Parameters

Name	Type	Description
Zones	ZoneType	A collection composed of data center information ZoneType

Error Code

Error Code	Description	Http status code	Meaning
MissingParameter	The input parameter “RegionId” that is mandatory for processing this request is not supplied.	400	RegionId value missing (or the user might not be authorized to use the region)

Query Access Points

Description

- Query the supported access point.

Request Parameters

Name	Type	Required	Description
Action	String	Yes	Required parameter, with a value:DescribeAccessPoints
RegionId	String	Yes	Destination RegionId
Type	String	Yes	The type of access point. If this parameter is not appointed, the request will return access point of all types. Optional value: VPC -

			Which indicates that this access point can support a physical leased line connecting to VPC.
PageNumber	Integer	No	Page number of the query result. The default value is 1 when the parameter is not specified. Value range: a positive integer. If the PageNumber value is greater than the greatest page number in the actual result, the content of the last page is returned.
PageSize	Integer	No	Page size of the query result. The value range is 1 to 100, and the default value is 10.

Return Parameters

Name	Type	Description
AccessPointSet	AccessPointSetType	A collection composed of data center information AccessPointType

Error Code

Error Code	Description	Http status code	Meaning
InvalidRegionId.NotFound	The RegionId provided does not exist in our records.	404	The appointed RegionId is not exisit.

Data Type

AccessPointSetType

Description

Set of leased line access point information. It is a list comprised of AccessPointTypes.

Node name

It depends on the interface.

Subnode

AccessPointType

AccessPointType

Description

Type of access point information

Node name

It depends on the interface.

Subnode

Name	Type	Description
AccessPointId	String	Access point ID
Type	String	Type of a leased line access point. Optional value: - VPC - VPC access point, which can be connected to a leased line accessing a VPC network

Status	String	Status of a leased line access point. Optional values: - Recommended - The status is good, and connection is recommended. - Hot - The status is normal, and the number of connected users is large. - Full - The number of connected users reaches the maximum value, and no more users can be connected.
Name	String	Name of a leased line access point
Description	String	Description of a leased line access point
AttachedRegionId	String	Physical region ID of an access point. If this region ID is empty, the access point does not belong to any region physically.
Location	String	Physical location of a leased line access point
HostOperator	String	Machine room carrier of a leased line access point

NextHopItemType

Description

- Data including the specific next hop data of a route.

Node name

NextHopItem

Subnode

Name	Type	Description
NextHopType	String	Type of the next hop target object. Optional values: Instance, Tunnel, and HaVip|RouterInterface
NextHopId	String	ID of the next-hop instance

NextHopListType

Description

- A list including next hop data of multiple routes.
- Supports ECMP routes.

Node name

NextHopList

Subnode

NextHopItemType

PhysicalConnectionSetType

Description

Set of physical leased line information. It is a list comprised of PhysicalConnectionTypes.

Node name

It depends on the interface.

Subnode

PhysicalConnectionType

PhysicalConnectionType

Description

Set of Physical Connection information.

Node name

It depends on the interface.

Subnode

Name	Type	Description
PhysicalConnectionId	String	Physical Connection ID
AccessPointId	String	ID of a leased line access point
Type	String	Leased line type. Optional values: - VPC - VPC leased line - Classic - Classic network leased line, which is used only for AntCloud. When connected, it can be used to access resources in the classic network of AntCloud (which is temporarily not supported).
Status	String	Physical leased line status. Optional values: - Initial - Application in progress - Approved - Application approved - Allocating – Resources

		being allocated - Allocated - Access construction in progress - Confirmed - Waiting for user confirmation - Enabled – Normal - Rejected – Application rejected - Canceled – Canceled - Allocation Failed - Resource allocation failed - Terminated – Terminated
BusinessStatus	String	- Normal - FinancialLocked (locked in arrears) - SecurityLocked (locked in security risk control)
CreationTime	String	Time when a Physical Connection is created
EnabledTime	String	Time when a Physical Connection is activated (time when regular billing starts)
LineOperator	String	Carrier providing physical lines for access. Optional values: - CT - China Telecom - CU - China Unicom - CM - China Mobile - CO - Other (China) - Equinix - Equinix - Other - Other (outside China)
Spec	String	Physical Connection specifications
PeerLocation	String	Physical location of the leased line peer (address)
PortType	String	Type of a Physical Connection access port. Optional values: - E1 - 2M synchronous

		electrical port - 100Base-T - 100M electrical port - 1000Base-T - 1G electrical port - 1000Base-LX - 1G single-mode optical port (10km) - 10GBase-T - 10G electrical port - 10GBase-LR - 10G single-mode optical port (10km) - Other - Unstated port
RedundantPhysicalConnectionId	String	Redundant Physical Connection ID
Name	String	Physical Connection name
Description	String	Physical Connection description
ADLocation	String	Physical location of the leased line access device (rack location)
PortNumber	String	Number of the port allocated for (connected to) the leased line access device
CircuitCode	Long	Circuit code of a corresponding carrier
Bandwidth	String	Physical Connection bandwidth

RouterInterfaceItemType

Description

Router interface description.

Node name

It depends on the interface.

Subnode

Name	Type	Description
RouterInterfaceId	string	Object ID
AccessPointId	string	Local access point ID
OppositeRegionId	string	Region of the connection peer
OppositeAccessPointId	string	Access point ID of the peer
Role	string	Role in the connection: initiator or receiver
Spec	string	Specifications
Name	string	Custom name
Description	string	Custom description
RouterId	string	Router ID
RouterType	string	Router type Currently, VRouter is the only value.
CreationTime	string	Creation time
Status	string	Status
BusinessStatus	string	Normal, FinancialLocked (locked in arrears), and SecurityLocked (locked in security risk control)
ConnectedTime	string	Connection time
OppositeInterfaceId	string	Interface ID of the connection peer
OppositeInterfaceSpec	string	RouterInterface specifications of the connection peer
OppositeInterfaceStatus	string	RouterInterface status of the connection peer
OppositeInterfaceBusiness Status	string	RouterInterface business status of the connection peer
OppositeRouterId	string	Router ID of the connection peer RouterInterface
OppositeRouterType	string	Router type of the connection peer RouterInterface
OppositeInterfaceOwnerId	string	RouterInterface owner ID of the connection peer
HealthCheckSourceIp	string	Health check source IP address
HealthCheckTargetIp	string	Health check target IP address

RouterInterfaceSetType

Description

Set of router interface information. It is a list comprised of RouterInterfaceItemTypes.

Node name

It depends on the interface.

Subnode

RouterInterfaceItemType

VirtualBorderRouterForPhysicalConnectionSetType

Description

Set of information of the virtual border router (VBR) on a physical leased line. It is a list comprised of VirtualBorderRouterForPhysicalConnectionTypes.

Node name

It depends on the interface.

Subnode

VirtualBorderRouterForPhysicalConnectionType

VirtualBorderRouterForPhysicalConnectionTy

pe

Description

Set of VBR information on a physical leased line

Node name

It depends on the interface.

Subnode

Name	Type	Description
VbrId	String	VBR ID
VbrOwnerUid	String	UID of a VBR owner. If the VBR owner is the physical leased line owner, an empty value is returned.
CreationTime	String	Time when a VBR is created
ActivationTime	String	Time when a VBR is activated (accepted by a VBR owner) for the first time
TerminationTime	String	Time when a VBR is terminated (by a physical leased line owner) most recently
RecoveryTime	String	Time when a VBR is activated from the terminated state (recovered by a physical leased line owner) most recently
VlanId	String	802.1Q VLAN ID (1-2,999) of VBR leased line interface
CircuitCode	String	Carrier circuit code corresponding to VBR physical leased line

VirtualBorderRouterSetType

Description

Set of VBR information. It is a list comprised of VirtualBorderRouterTypes.

Node name

It depends on the interface.

Subnode

VirtualBorderRouterType

VirtualBorderRouterType

Description

Set of VBR information

Node name

It depends on the interface.

Subnode

Name	Type	Description
VbrId	String	VBR ID
CreationTime	String	Time when a VBR is created
ActivationTime	String	Time when a VBR is activated (accepted by a VBR owner) for the first time
TerminationTime	String	Time when a VBR is terminated (by a physical leased line owner) most recently
RecoveryTime	String	Time when a VBR is activated from the terminated state (recovered by a physical leased line owner) most recently
Status	String	VBR status

VlanId	String	802.1Q VLAN ID (0-2,999) of VBR leased line interface. The value 0 indicates there is no VLAN encapsulation.
CircuitCode	String	Carrier circuit code corresponding to VBR physical leased line
RouteTableId	String	VBR route table ID
VlanInterfaceId	String	ID of VBR leased line interface. It can be used as the next hop of the route on VBR.
LocalGatewayIp	String	IP address of the local VBR leased line interface
PeerGatewayIp	String	IP address of the peer VBR leased line interface
PeeringSubnetMask	String	Subnet mask connecting the local and peer VBR leased line interfaces
Name	String	VBR name
Description	String	VBR description
PhysicalConnectionId	String	Physical leased line ID
PhysicalConnectionStatus	String	Physical leased line status (The Deleted status indicates that the leased line is deleted.)
PhysicalConnectionBusinessStatus	String	Physical leased line business status (The physical leased line is deleted if its business status is empty.)
PhysicalConnectionOwnerUid	String	UID of the physical leased line owner (The physical leased line owner is deleted if its UID is empty.)
AccessPointId	String	Access point ID

Appendix

Time Format ISO8601

ISO 8601 Data elements and interchange formats – Information interchange – Representation of dates and times is an international standard issued by ISO. ECS Open API adopts ISO8601 time format. For details, please refer to ISO8601.

Error Code Table

Client Errors

For details, refer to the error codes of each interface

Server Errors

Error Code	Description	Http Status Code	Meaning
ServiceUnavailable	The request has failed due to a temporary failure of the server.	503	The service is unavailable
InternalError	The request processing has failed due to some unknown error, exception or failure.	500	Internal error

How to Ensure Idempotence

If a request timeout or internal server error is encountered when calling the CreateInstance interface to create an ECS in the ECS, the client might try to resend the request. At this time, the client can prevent the server from creating more instances than expected by providing the optional 'ClientToken' parameter. Providing this parameter will also ensure the idempotence of the request. 'ClientToken' is a unique, case sensitive string generated by the client that cannot exceed 64 ASCII characters.

If a user uses the same 'ClientToken' value to call the CreateInstance interface, the server will return identical request results, including the same 'InstanceId'. Therefore, when the user encounters an error and tries again, providing the same 'ClientToken' value can ensure that the ECS creates only one instance and obtains this 'InstanceId'.

If the user provides a 'ClientToken' that has already been used, but with different request parameters, the ECS will return the 'IdempotentParameterMismatch' error code. However, you

must note that the 'SignatureNonce', 'TimeStamp', and 'Signature' parameters must be changed when trying to submit the request again. Because the ECS uses 'SignatureNonce' to prevent replay attacks and 'TimeStamp' to mark the time of each request, the duplicate request must provide different 'SignatureNonce' and 'TimeStamp' parameter values. This will also produce a different 'Signature' value.

Generally, clients will only need to retry an operation if they encounter a 500 ('InternalError') or 503 ('ServiceUnAvailable') error or cannot receive a response. When 200 is returned, retrying the operation will give you the same results as the last time, but will not have any effect on the server status. When a 4xx error code is returned, unless the message explicitly says "try it later", retrying the operation will usually not be successful.

How to Call Interfaces

The service interface is called by sending an HTTP request to the server (you can send it through the HTTP or HTTPS channel) and receiving the server's response to this request. After the server receives a user request, it will perform necessary authentication and parameter verification on the request. After all verifications are successful, it submits or completes the associated operation based on the parameters specified for the request. Then, it returns the processing results to the caller in the form of an HTTP response.

Request Composition

Requests are composed of the following parts:

HTTP method - Currently, all service interfaces only support GET method calls.

Request URL - The request's service address, name of the operation to be executed, operation parameters, and public request parameters are included in the request URL.

Server address: The service domain names arehttp://vpc.aliyuncs.com/and https://vpc.aliyuncs.com/. In order to ensure the security of the request, we strongly recommend that you use the HTTPS channel. (HTTPS has added the SSL layer for encrypted communication and can prevent communication interception leading to the leakage of sensitive information.)

Operation name: Each interface requires that the name of the operation to be executed be specified, i.e. the Action parameter.

Operation parameter: Different operation parameters must be set for different operations to be executed. For details, see the instructions for each interface.

Public request parameters: Parameters such as TimeStamp and Signature must be included in each request.

For the server to correctly authenticate the user's identity and authorize request execution, the

request must be signed before submission for processing. For signature rules, refer to the signature mechanism section.

After the server finishes processing the request, it will return the response results. Response results are categorized into successful results and error messages. For an illustration of the format, refer to the Returned Results section. The client can parse the response message body to obtain the execution results.

Call Example

Taking the ‘DescribeRegions’ interface for example:

The corresponding Action is ‘DescribeRegions’, and the required operation parameter is ‘RegionId’ (all RegionIds can be obtained using the interface for querying the available region list). After adding all the public request parameters (except ‘Signature’), the request URL will be (for ease of reading, this is the URL before URL encoding):

```
http://ecs.aliyuncs.com/?TimeStamp=2012-12-
26T10:33:56Z&Format=XML&AccessKeyId=testid&Action=DescribeRegions&SignatureMethod=HMAC-
SHA1&SignatureNonce=NwDAxvLU6tFE0DVb&Version=2014-05-26&SignatureVersion=1.0
```

According to the signature calculation rules, the Canonicalized Query String is constructed first, as shown below:

```
http://ecs.aliyuncs.com/?TimeStamp=2012-12-
26T10:33:56Z&Format=XML&AccessKeyId=testid&Action=DescribeRegions&SignatureMethod=HMAC-
SHA1&SignatureNonce=NwDAxvLU6tFE0DVb&Version=2014-05-26&SignatureVersion=1.0
```

Then, ‘StringToSign’ is constructed as a signature string with a value:

```
GET&%2F&AccessKeyId%3Dtestid&Action%3DDescribeRegions&Format%3DXML&SignatureMethod%3DHMAC-
SHA1&SignatureNonce%3DNwDAxvLU6tFE0DVb&SignatureVersion%3D1.0&TimeStamp%3D2012-12-
26T10%253A33%253A56Z&Version%3D2014-05-26
```

The following Java sample code demonstrates how public request parameters are added, how the Canonicalized Query String is constructed from the request parameters, and how the ‘StringToSign’ is constructed. This example assumes that all request parameters are placed in a Map<String, String> object and the Access Key ID is “testid”.

```
    final String HTTP_METHOD = "GET";

Map<String, String> parameters = new HashMap<String, String>();
// Add public request parameters
parameters.put("Action", "DescribeRegions");
parameters.put("Version", "2014-05-26");
parameters.put("AccessKeyId", "testid");
```

```
parameters.put("TimeStamp", formatIso8601Date(new Date()));
parameters.put("SignatureMethod", "HMAC-SHA1");
parameters.put("SignatureVersion", "1");
parameters.put("SignatureNonce", UUID.randomUUID().toString());
parameters.put("Format", "XML");

// Sort the parameters
String[] sortedKeys = parameters.keySet().toArray(new String[]{});
Arrays.sort(sortedKeys);

final String SEPARATOR = "&";

// Generate the stringToSign
StringBuilder stringToSign = new StringBuilder();
stringToSign.append(HTTP_METHOD).append(SEPARATOR);
stringToSign.append(percentEncode("/")).append(SEPARATOR);

StringBuilder canonicalizedQueryString = new StringBuilder();
for(String key : sortedKeys) {
// Be sure to encode the key and value
canonicalizedQueryString.append("&")
.append(percentEncode(key)).append("=")
.append(percentEncode(parameters.get(key)));
}

// Be sure to encode the canonicalizedQueryString
stringToSign.append(percentEncode(
canonicalizedQueryString.toString().substring(1)));
```

In this example, please note that the 'TimeStamp' parameter must comply with the ISO8601 norms. Furthermore, please note that UTC time is used. Otherwise, the request will encounter an error. The following sample code demonstrates how a compliant TimeStamp string is generated:

```
    private static final String ISO8601_DATE_FORMAT = "yyyy-MM-dd'T'HH:mm:ss'Z'";
private static String formatIso8601Date(Date date) {
SimpleDateFormat df = new SimpleDateFormat(ISO8601_DATE_FORMAT);
df.setTimeZone(new SimpleTimeZone(0, "GMT"));
return df.format(date);
}
```

When generating a Canonicalized Query String (the canonicalizedQueryString variable in the example) and 'StringToSign', both must be encoded. The encoding rules are described in detail in the signature mechanism section. The following sample code demonstrates how the java.net.URLEncoder class is used to complete encoding:

```
    private static final String ENCODING = "UTF-8";

private static String percentEncode(String value) throws UnsupportedEncodingException {
return value != null ? URLEncoder.encode(value, ENCODING).replace("+", "%20").replace("*", "%2A").replace("%7E",
"~") : null;
}
```

If we assume the “Access Key ID” is “testid”, the “Access Key Secret” is “testsecret”, and the Key used for HMAC calculation is “testsecret&”, the calculated signature value is:

```
SDFQNvyH5rtkc9T5Fwo8DOjw5hc=
```

Sample code for signature calculation (Java):

```
// The following is a sample code for signature calculation
final String ALGORITHM = "HmacSHA1";
final String ENCODING = "UTF-8";
key = "testsecret&";

Mac mac = Mac.getInstance(ALGORITHM);
mac.init(new SecretKeySpec(key.getBytes(ENCODING), ALGORITHM));
byte[] signData = mac.doFinal(stringToSign.getBytes(ENCODING));

String signature = new String(Base64.encodeBase64(signData));
```

After adding the Signature parameter, please perform URL encoding according to the RFC3986 rules and obtain

```
http://ecs.aliyuncs.com/?TimeStamp=2012-12-
26T10%3A33%3A56Z&Format=XML&AccessKeyId=testid&Action=DescribeRegions&SignatureMethod=HMAC-
SHA1&RegionId=region1&SignatureNonce=NwDAxvLU6tFE0DVb&Version=2012-09-
13&SignatureVersion=1.0&Signature=SDFQNvyH5rtkc9T5Fwo8DOjw5hc%3d
```

Next, send an HTTP request to the above URL and receive a response to the request from the server, as shown below:

```
<DescribeRegionsResponse>
<Regions>
<Region>
<LocalName>Qingdao node</LocalName>
<RegionId>cn-qingdao</RegionId>
</Region>
<Region>
<LocalName>Hangzhou node</LocalName>
<RegionId>cn-hangzhou</RegionId>
</Region>
</Regions>
<RequestId>833C6B2C-E309-45D4-A5C3-03A7A7A48ACF</RequestId>
</DescribeRegionsResponse>
```

By parsing this XML output, you can obtain an ID and LocalName list for all available regions. If, when submitting a request, the specified Format parameter is JSON, and then the result will also be returned in the JSON format.