ECHOplatform REST API Description

https://campus.barracuda.com/doc/96774500/

The ECHOplatform REST API is a RESTful web service implemented using HTTP and principles such as referencing resources by a URI, manipulating resources by using the standard HTTP methods (GET, PUT, POST, and DELETE), and exchanging representations of the resource in a negotiated media type format. GET requests are safe and GET, PUT, and POST are idempotent.

This article provides the following topics:

- Making Requests
- Objects

Making Requests

After obtaining an access token from the authorization server, the client can make API requests to resources. The access token is sent by including the Authorization header in the HTTP request.

GET /v1/general/time HTTP/1.1

Host: api.intronis.com

Authorization: OAuth {ACCESS_TOKEN}

Accept: application/json

Headers

A request must contain the Authorization header with the OAuth 2 access token. A request must contain the Accept header and possibly the Content-Type header with a value of either application/json or application/xml. Only one media type should be sent and must be encoded in UTF-8. A request can contain the Accept-Encoding to receive a compressed response.

Request Header	Required?	Values
Authorization	Yes	OAuth {ACCESS_TOKEN}
Accept	Yes	application/json or application/xml
Content-Type	Yes – for PUT, POST requests	application/json or application/xml
Accept-Encoding	No	gzip, compress, or deflate

Errors

The client application should be able to handle any 4xx or 5xx HTTP error status code. Specific error status codes are as follows.

Status Code	Description
400 Bad Request	There was an error with the contents of the request. An error object representation will be included in the response's body.
401 Unauthorized	The access token provided is invalid or expired.
403 Forbidden	The access token does not grant scope to the requested resource.
404 Not Found	The requested resource does not exist.
405 Method Not Allowed	The requested method to the resource is not allowed.
406 Not Acceptable	An Accept-* header in the request is not supported by the server.
500 Internal Server Error	An error occurred on the server.
503 Service Unavailable	A dependent service was unavailable. The request can be attempted later.

Objects

Scalar types (int, decimal, string, bool) follow standard definitions. Objects are an unordered set of name/value pairs and the order is not guaranteed. Arrays of objects are ordered and the order is guaranteed between two identical requests.

Collections

Collections are objects that contain a list of objects that has additional attributes to help simplify pagination.

Attributes	• page (int) – Current page number request • page_size (int) – Max size of page requested • count (int) – total number of objects in all pages • links (link objects) – a number of navigation links to first page, next page (if exists), and previous page (if exists) • list (objects) – array of objects returned

The list attribute differs between JSON and XML as XML already includes the links and the list in a root element, where JSON does not.

Content-Type	Collection Example

application/JSON	{
	"page": 1,
	"page_size": 10,
	"count": 1,
	"links": [
	{
	"rel": "first",
	"href": "https://api.intronis.com/v1/general/cancellation_reasons?page=1"
	}
	],
	"list": [
	{
	"reason_id": 1,
	"description": "Customer has gone out of business",
	"requires_note": false
	}
	]
	}

application/XML	<?xml version="1.0" encoding="UTF-8"?> <list page="1" page_size="10" count="1"> <link rel="first" href="https://api.intronis.com/v1/general/cancellation_reasons?page=1"/> <reason> <reason_id>1</reason_id> <description>Customer has gone out of business</description> <requires_note>false</requires_note> </reason> </list>

Link Objects

Link objects always contain 2 attributes: rel and href. These carry the same meanings as the link tag in HTML. The rel attribute value can used to give context to the href value.

rel value	href meaning
self	The location of the resource the link is contained within
first	The location of the first page of the given list.
prev	The location of the next page of the given list.
next	The location of the previous page of the given list.

Content-Type	Link Example

application/JSON	{
	"rel": "self",
	"href": "https://api.intronis.com/v1/partners/barracuda"
	}

application/XML	<link rel="self" href="https://api.intronis.com/v1/partners/barracuda" />

The title attribute can be used in xml to give the link a name.

Content-Type	Link Example
application/JSON	"title: "info", "rel": "related", "href": "https://api.intronis.com/v1/partners/barracuda/info" }
application/XML	<link title="info" rel="self" href="https://api.intronis.com/v1/partners/barracuda/info" />

String Constants

String constants are specially defined strings that are in uppercase. The values are defined for each resource.

Multiple String Constants in Queries

Some Query Select Parameters allow multiple string constants to be sent by separating them by commas.

Description	URL Example
Single String Constant Query	https://api.intronis.com/v1/partners/barracuda/reports/backup_status ?condition_status=FAILURE
Multiple String Constant Query	https://api.intronis.com/v1/partners/barracuda/reports/backup_status ?condition_status=WARNING,FAILURE

Dates

Dates are represented as strings in ISO8601 format (http://www.w3.org/TR/NOTE-datetime).

Error Objects

Error objects always contain 2 string attributes: error_code and error_description.

Content-Type	Link Example

application/JSON	{
	"error_code": "invalid_characters",
	"error_description": "address contains invalid characters"
	}

application/XML	<error> <error_code>invalid_characters</error_code> <error_description>address contains invalid characters</error_description> </error>

© Barracuda Networks Inc., 2024 The information contained within this document is confidential and proprietary to Barracuda Networks Inc. No portion of this document may be copied, distributed, publicized or used for other than internal documentary purposes without the written consent of an official representative of Barracuda Networks Inc. All specifications are subject to change without notice. Barracuda Networks Inc. assumes no responsibility for any inaccuracies in this document. Barracuda Networks Inc. reserves the right to change, modify, transfer, or otherwise revise this publication without notice.