Genesys Rules System REST API Reference Guide

Overview

5/3/2025

Contents

Overview

Authentication / Security

The API is authenticated using HTTP Basic Authentication.

SSL is strongly recommended by Genesys. This allows the unique authentication token (APIToken) to be safely sent on each request. The APIToken will be passed in the header using HTTP Basic protocol.

A configuration option—`rest-api`—enables a customer to control whether or not they wish to force SSL. SSL might not be required, such as when GRAT server is positioned behind an ELB, and the communications between the ELB and GRAT server are protected behind a firewall.

Request Format

The base URL of the API is `https://<server>:<port>/grs/v1/` .

The "v1" indicates version 1 and allows for future expansion while maintaining compatibility with previous versions of the API.

JSON Bodies

All POST and PUT requests will contain JSON encoded request bodies, and must have have content type of `application/json`, or the API will return a `415 Unsupported Media Type` status code.

HTTP Verbs

We use standard HTTP verbs to indicate intent of a request:

- GET—To retrieve a resource or a collection of resources (all Queries)
- POST—To create a resource (Create rule package, Create rule)
- PUT—To modify a resource (Modify rule package)
- DELETE—To delete a resource (Delete Rule, Delete Calendar)

Responses

All response bodies are JSON-encoded.

A single resource is represented as a JSON object:

```
{
  "field1": "value",
  "field2": true,
  "field3": []
}
```

A collection of resources is represented as a JSON array of objects:

```
[
  {
    "field1": "value",
    "field2": true,
    "field3": []
  },
  {
    "field1": "another value",
    "field2": false,
    "field3": []
  }
]
```

Timestamps are in UTC and formatted as ISO8601.

HTTP Status Codes

We use HTTP status codes to indicate success or failure of a request.

Success codes

- 200 OK—Request succeeded. Response included
- 201 Created—Resource created. Response included of newly created resource
- 204 No Content—Request succeeded, but no response body

Error codes

- 400 Bad Request—Could not parse request
- 401 Unauthorized—No authentication credentials provided or authentication failed
- 403 Forbidden—Authenticated user does not have access. For example, they do not have `PACKAGE_CREATE` permission and attempted to create a package
- 404 Not Found—Resource not found. For example; Query rule package detail and the provided package ID was not found.

- 415 Unsupported Media Type—POST/PUT request occurred without an `application/json` content type.
- 422 Unprocessable Entry—A request to modify or create a resource failed due to a validation error.

In case of validation errors on a POST/PUT/PATCH request, a `422 Unprocessable Entry` status code will be returned. The JSON response body will include an array of error messages.

```
{
  "message": "Validation Failed",
  "errors": [
    {
      "message": "Field is not valid"
    },
    {
      "message": "OtherField is already used"
    }
  ]
}
```

- 500, 501, 502, 503—An internal server error occurred

Configuration

By default, the new REST API is not enabled. You can enable it after upgrading to GRS 8.5.2 by setting configuration option `rest-api`. In addition, this configuration option will enable you to determine whether or not to force only SSL communications. Genesys recommends running over SSL to protect the authentication tokens that flow on each request from compromise. However, there are situations where SSL would not be required (testing labs, positioning server behind firewalls, etc.). In this case, you can use the option to disable the check for SSL.

Configuration Option—`rest-api`
Default Value—`disabled`
Note—Changes to this option will require a restart of the GRAT server to take effect.
Possible Values:

- `disabled`—(default)The REST API is disabled and will not accept any requests.
- `enabled`—The REST API is enabled and will accept both secure (https) and non-secure (http) requests.
- `requireSSL`—The REST API is enabled and will only accept secure (https) requests.