This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Genesys Rules System REST API Reference Guide

Genesys Rules System 8.5.3

12/29/2021

Table of Contents

Genesys Rules System REST API Reference Guide

API Functions

The GRS REST API provides functions to manage:

- Authentication—login, logout.
- Business Hierarchy
- Rule Templates
- Rule Packages
- Parameters
- Rules
- Business Calendars
- Deployments and Snapshots

Applications or custom user interfaces (which can run in parallel to or instead of the Genesys Rules Authoring Tool web application) can use this API to perform rule authoring and deployment.

This reference can be used to write applications that use the Genesys Rules System Representational state transfer (REST) Application Programming Interface (API). This reference also explains the resources and methods available to developers. Each category presents information about relevant operations, related resources, request parameters, and return values.

Enabling/Disabling the GRS REST API

Use of the API can be enabled or disabled by setting the value of configuration option `rest-api`. In addition, this configuration option will enable you to determine whether or not to force only SSL communications. Genesys strongly recommends running over SSL in order to protect the authentication tokens that flow on each request from compromise. SSL can be disabled where appropriate (for example, testing labs, positioning server behind firewalls, and so on). Valid values are as follows:

- `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.

Platform Support

In this release, the REST API will be available only from the Tomcat application server.

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>/<grat-application-path>/grs/v1/` .

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

For example, the base URL can be:

```
https://localhost:8080/genesys-rules-authoring/grs/v1/
```

where `genesys-rules-authoring` is the application path of Genesys Rules Authoring application.

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

For each valid request, the responses are returned in JSON format.

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 Entity—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 Entity` 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.

Authentication

Login

Method

GET

Syntax

```
/grs/v1/login
```

Request Body

N/A

Response Body

```
{
"apiToken": "749cf21d112cc88cdeb72c441724ec0b
71e53a6a7666e847209cac9593f96f5e",
"lastLogin": 0
}
```

HTTP Status Codes

- 200 OK—Response Body Provided
- 401 Unauthorized—Userid/Password not valid
- 422 Unprocessable Entity

```
{
  "message": "Validation Failed",
  "errors": [
    {
       "code" : 1000,
       "message" : "Password Change Required"
    }
  ]
}
```

- 503—Service Unavailable

Additional Errors:

- 1001—No User Connection
- 1002—Client port bind failed
- 1003—Max Connections Exceeded

Notes

Userid and Password are provided in the basic authorization header.

- `apiToken`—A unique session `apiToken` is returned on a successful login. `Userid` and `apiToken` must be included on subsequent requests via the basic authorization header.
- `lastLogin`—(If available from Configuration Server) is returned as a long representing time in ms since epoch.

Change Password and Login

Method

PUT

Syntax

```
/grs/v1/changepassword
```

Request Body

```
{
   "newPassword":"AnotherPW"
 }
```

Response Body

```
 |
{
"apiToken": "749cf21d112cc88cdeb72c441724ec0b
71e53a6a7666e847209cac9593f96f5e",
"lastLogin": 0
}
```

HTTP Status Codes

- 200 OK—Response Body Provided
- 401Unauthorized—Userid / Original Password not valid
- 503—Service Unavailable

Notes

A user cannot simply change their password at any time. The password must expire or the flag to "reset" password must be enabled in Configuration Server. When a user attempts to log in and gets the "Password Change Required" error, they can then use this API to change their password and log in. UserID and Original Password are provided in the basic authorization header. The new password is passed in the request body. If the password change is successful, the API token and last login time (if available) will be returned (just like normal login).

Logout

Method

GET

Syntax

```
/grs/v1/logout
```

Optional Parameters:

- `&force=true`—log out even if there are locked resources. Any locked resources will automatically be unlocked.

Request Body

N/A

Response Body

N/A

HTTP Status Codes

- 204 No Content—Logout successful
- 401 Unauthorized—User not logged in or API Token not otherwise recognized
- 422 Unprocessable Entity —User has locked resources:

```
{
"message": "User has locked resources",
"resource": [
"dmd:Rule-100",
"package3:DT-163",
"Calendar-200"
]
}
```

- 503—Service Unavailable

Notes

Query Locks

Method

Get

Syntax

```
/grs/v1/querylocks
```

Request Body

N/A

Response Body

```
{
"message": "User has locked resources",
"resource": [
"dmd:Rule-100",
"package1:DT-163",
"Calendar-200"
]
}
```

HTTP Status Codes

- 200 OK—Response body provided
- 204 No Content—User has no locked resources
- 401 Unauthorized—APIToken not valid
- 503—Service Unavailable

Notes

Permissions

Method

GET

Syntax

```
/grs/v1/permissions/{tenantId}
```

Request Body

N/A

Response Body

```
{
   "canCreatePackage":true,
   "canModifyPackage":true,
   "canDeletePackage":true,
   "canDeployPackage":true,
   "canViewRule":true,
   "canCreateRule":true,
   "canModifyRule":true,
   "canEditRuleOnly":false,
   "canDeleteRule":true,
   "canViewCalendar":true,
   "canCreateCalendar":true,
   "canModifyCalendar":true,
   "canDeleteCalendar":true,
   "canViewPackageSnapshot":true,
   "canCreatePackageSnapshot":true,
   "canDeletePackageSnapshot":true,
   "canViewTestScenario":true,
   "canCreateTestScenario":true,
   "canModifyTestScenario":true,
   "canDeleteTestScenario":true,
   "canExecuteTestScenario":true,
   "canCreateTemplate":true,
   "canModifyTemplate":true,
   "canDeleteTemplate":true
}
```

HTTP Status Codes

- 200 OK—Response body provided
- 401 Unauthorized—APIToken not valid
- 403 Forbidden—User does not have permission for the specified tenant
- 503—Service Unavailable

Notes

User must have permission to the specified tenant ID. Use Query Tenants to get a list of valid tenant IDs for this logged in user.

Package-Level Permissions (8.5.302+)

Method

GET

Syntax

```
/grs/v1/permissions/{tenantId}/package/{packageId}
```

Request Body

N/A

Response Body

```
{
   "canViewPackage":true,
   "canModifyPackage":true,
   "canDeletePackage":true,
   "canDeployPackage":true,
   "canViewRule":true,
   "canCreateRule":true,
   "canModifyRule":true,
   "canEditRuleOnly":false,
   "canDeleteRule":true,
   "canViewCalendar":true,
   "canCreateCalendar":true,
   "canModifyCalendar":true,
   "canDeleteCalendar":true,
   "canViewPackageSnapshot":true,
   "canCreatePackageSnapshot":true,
   "canDeletePackageSnapshot":true,
   "canViewTestScenario":true,
   "canCreateTestScenario":true,
   "canModifyTestScenario":true,
   "canDeleteTestScenario":true,
   "canExecuteTestScenario":true,
}
```

HTTP Status Codes

- 200 OK—Response body provided
- 401 Unauthorized—APIToken not valid
- 403 Forbidden—User does not have permission for the specified tenant
- 503—Service Unavailable

Notes

User must have permission to the specified tenant ID. Use Query Tenants to get a list of valid tenant IDs for this logged-in user.

Business Hierarchy

- Query Tenants
- Query Business Hierarchy

Query Tenants

Method

GET

Syntax

```
/grs/v1/tenants
```

Request Body

N/A

Response Body

```
{
   "tenants":[
      {
         "nodeId":1,
         "name":"Environment"
      },
      {
         "nodeId":104,
         "name":"KC Test Environment"
      }
   ]
}
```

HTTP Status Codes

- 200 OK—Response body provided
- 401 Unauthorized—APIToken not valid
- 503—Service Unavailable

Notes

Query Tenants returns the list of tenant names and IDs that this user is allowed to see based on their login credentials.

Query Business Hierarchy

Method

GET

Syntax

```
/grs/v1/tenant/{tenantId}/businesshierarchy
```

Request Body

N/A

Response Body

```
{
   "nodes":[
      {
         "nodeId":346,
         "name":"ACME Solution",
         "children":[
            {
               "nodeId":458,
               "name":"Financial Department",
               "children":null
            },
            {
               "nodeId":351,
               "name":"Sales",
               "children":[
                  {
                     "nodeId":353,
                     "name":"Outbound",
                     "children":null
                  }
               ],
            },
            {
               "nodeId":457,
               "name":"Sales Department",
               "children":null
            }
         ],
      }
   ]
}
```

HTTP Status Codes

- 200 OK—Response body provided
- 401 Unauthorized—APIToken not valid
- 403 Forbidden—User does not have permission for the specified tenant.
- 503—Service Unavailable

Notes

Will return the list of nodes that the userid has permission to view.

Templates

- Query Templates by Type

Query Templates by Type

Method

GET

Syntax

```
/grs/v1/tenant/{tenantId}/templates
```

Optional parms:

- `&type=samples`
- `&maxVersions=3`

Request Body

N/A

Response Body

```
{
   "templates":[
      {
         "name":"test",
         "version":1,
         "type":"iWD",
         "global":false,
         "changeComment":"Initial"
      },
      {
         "name":"ABC",
         "version":2,
         "type":"samples",
         "global":false,
         "changeComment":"Fixed bug"
      },
       {
         "name":"ABC",
         "version":1,
         "type":"samples",
         "global":false,
         "changeComment":"Initial"
      }
   ]
}
```

HTTP Status Codes

- 200 OK—Response body provided
- 401 Unauthorized—APIToken not valid
- 403 Forbidden—User does not have permission for the specified tenant.
- 503—Service Unavailable

Notes

By default, this query returns all versions (up to 100) of all templates published for the specified tenant that the logged in user has permission to view. To further filter, specify: `&type` and/or `&maxVersions` parameters.

Rule Packages

- Query Packages
- Query Package Details
- Create Package
- Modify Package
- Delete Package

Query Packages

Method

GET

Syntax

```
/grs/v1/tenant/{tenantId}/businessnode/{nodeId}/packages
```

Request Body

N/A

Response Body

```
{
   "packages":[
      {
         "packageId":"1fb18a53-17b2-4ae1-81c7-1614f483c754",
         "name":"dmd",
         "type":"samples",
         "businessName":"XYZ",
         "description":"SamplePackage",
         "templates":[
            {
               "name":"XYZ",
               "version":1,
               "type":null,
               "global":false,
               "changeComment":null
            }
         ]
      }
   ]
}
```

HTTP Status Codes

- 200 OK—Response body provided
- 401 Unauthorized—APIToken not valid
- 403 Forbidden—User does not have permission for the specified tenant or business node.
- 503—Service Unavailable

Notes

User must have permission to view the specified tenant and business node.

Query Package Details

Method

GET

Syntax

```
/grs/v1/tenant/{tenantId}/businessnode/{nodeId}/package/{packageId}
```

Request Body

N/A

Response Body

```
{
   "packageId":"1fb18a53-17b2-4ae1-81c7-1614f483c754",
   "name":"dmd",
   "type":"samples",
   "businessName":"XYZ",
   "description":"SamplePackage
Note: Must enable wildcards for this to work
decision-table-enable-wildcards=true",
   "templates":[
      {
         "name":"XYZ",
         "version":1,
         "type":samples,
         "global":false,
         "changeComment":"Initial"
      }
   ]
}
```

HTTP Status Codes

- 200 OK—Response body provided

- 401 Unauthorized—APIToken not valid
- 404 Not Found—could not find specified package ID. Use Query Packages to obtain correct ID.
- 503—Service Unavailable

Create Package

Method

POST

Syntax

```
/grs/v1/tenant/{tenantId}/businessnode/{nodeId}/package
```

Request Body

```
{
   "name":"XYZ",
   "type":"samples",
   "businessName":"XYZ",
   "description":"SamplePackage
Note: Must enable wildcards for this to
work decision-table-enable-wildcards=true",
   "templates":[
      {
         "name":"Existing Template",
         "version":7
      },
  {
         "name":"A new template",
         "version":2
      }
   ]
}
```

Response Body

```
{
"packageId", "f44268af-dd9e-4035-ba1f-acd53d78fc89",
  "name":"XYZ",
   "type":"samples",
   "businessName":"XYZ",
   "description":"SamplePackage",
   "templates":[
      {
         "name":"Existing Template",
         "version":7
      },
  {
         "name":"A new template",
         "version":2
      }
   ]
}
```

HTTP Status Codes

- 201 Created—Response body provided
- 401 Unauthorized—APIToken not valid
- 403 Forbidden—User does not have permission for the specified tenant or business node, or does not have proper permission to perform this operation.
- 422 Unprocessable Entity—Field Validation Error

```
{
  "message": "Validation Failed",
  "errors": [
    {
       "code" : "1000",
       "field" : "name",
       "message": "package name must be specified"
    }
  ]
}
```

Other possible return values:

- 1001—name—package name contains invalid character(s)
- 1002—name—package name cannot contain java keyword(s)
- 1003—name—package name already in use
- 1004—type—package type must be specified
- 1005—templates—specified template: {templatename} version: {version} not found
- 1006—businessName—business name already in use
- 503—Service Unavailable

Modify Package

Method

PUT

Syntax

```
/grs/v1/tenant/{tenantId}/businessnode/{nodeId}/package/{packageId}
```

Request Body

```
{
   "businessName":"New Business Name",
   "description":"New Description",
   "templates":[
      {
         "name":"XYZ",
         "version":1
      },
  {
         "name":"AnotherTemplate",
         "version":2
      }
   ]
}
```

Response Body

```
{
   "name":"XYZ",
   "type":"samples",
  "businessName":"New Business Name",
   "description":"New Description",
   "templates":[
      {
         "name":"SamplePackage",
         "version":1
      },

  {
         "name":"AnotherTemplate",
         "version":2
      }
   ]

}
```

HTTP Status Code

- 200 OK—Response body provided
- 401 Unauthorized—APIToken not valid
- 403 Forbidden—User does not have permission for the specified tenant or business node, or does not have proper permission to perform this operation.
- 422 - Unprocessable Entity - Field Validation Error

[+]Show

```
{

  "message": "Validation Failed",
  "errors": [
    {

       "code" : "1006",

       "field" : "businessName",

       "message": "busnessName already in use"
    }

  ]

}
```

- 1005—templates—specified template: {templatename} version: {version} not found
- 1006—businessName—business name already in use
- 503—Service Unavailable

Notes

User must have `MODIFY_PACKAGE` permission. User can modify the following fields:

- businessName
- description
- templates

Any other fields modified will be ignored.

Delete Package

> **Warning**
>
> Before deleting a rule package, you should Undeploy it, which removes it from service from your GRE instances. Once a rule package is deleted, it cannot be undeployed via the API or via GRAT.

Method

DELETE

Syntax

```
/grs/v1/tenant/{tenantId}/businessnode/{nodeId}/package/{packageId}
```

Request Body

N/A

Response Body

N/A

HTTP Status Codes

- 204 No Content—Returned if successful operation
- 401 Unauthorized—APIToken not valid
- 403 Forbidden—User does not have permission for the specified tenant or does not have proper permission to perform this operation.
- 404 Not Found—Could not find specified package ID.
- 503—Service Unavailable

Notes

User must have PACKAGE_DELETE permission.

Query Orphaned Rule Packages

Method

GET

Syntax

```
/grs/v1/tenant/{tenantId}/orphanpackages
```

Request body

N/A

Response body

```
{
"packages": [
{
"packageId": "fb8627c2-911f-4ec2-90c9-30c2e60c8839",
"name": "my.lost.pkg",
"type": "samples",
"businessName": "My Lost Package",
"description": "Lost but not forgotten",
"businessNodeId": "774",
"templates": [
{ "name": "GRSSplitTest", "version": 1, "type": null, "global": false, "changeComment": null,
"lastModified": 0, "modifiedBy": null }
]
}
]
}
```

HTTP codes

- 200 OK—Response body provided
- 401 Unauthorized—APIToken not valid

- 403 Forbidden—User does not have permission for the specified tenant
- 503—Service Unavailable

Delete Orphaned Rule Package

Method

DELETE

Syntax

```
grs/v1/tenant/{tenantId}/orphanpackage/{packageId}
```

Request body

`N/A

Response body

N/A

HTTP codes

- 204 No Content—Returned if successful operation
- 401 Unauthorized—APIToken not valid
- 403 Forbidden—User does not have permission for the specified tenant or does not have proper permission to perform this operation.
- 404 Not Found—Could not find specified package ID.
- 503—Service Unavailable

Query Rule Package by Name

Method

GET

Syntax

```
/grs/v1/tenant/{tenantId}/packagename/{packageName}
```

Request body

N/A

Response body

```
{
"packageId": "7324b497-9d34-468b-8dcf-84690e4efd3c",
"name": "call.treatment",
"type": "call treatment",
"businessName": "",
"description": "Handle incoming calls & stuff",
"businessNodeId": "346",
"templates": [
{ "name": "CallTreatment", "version": 4, "type": null, "global": false, "changeComment":
null, "lastModified": 0, "modifiedBy": null }
]
}
```

HTTP codes

- 200 OK—Response body provided
- 401 Unauthorized—APIToken not valid
- 404 Not Found—could not find specified package.
- 503—Service Unavailable

Query Rule Package by ID

Method

GET

Syntax

```
/grs/v1/tenant/{tenantId}/packageid/{packageId}
```

Request body

N/A

Response body

```
{
"packageId": "7324b497-9d34-468b-8dcf-84690e4efd3c",
"name": "call.treatment",
"type": "call treatment",
"businessName": "",
"description": "Handle incoming calls & stuff",
"businessNodeId": "346",
"templates": [

{ "name": "CallTreatment", "version": 4, "type": null, "global": false, "changeComment":
null, "lastModified": 0, "modifiedBy": null }
]
}
```

HTTP codes

- 200 OK—Response body provided
- 401 Unauthorized—APIToken not valid
- 404 Not Found—could not find specified package ID. Use Query Packages to obtain correct ID.
- 503—Service Unavailable

Parameters

- Query Parameters

Query Parameters

Method

GET

Syntax

```
/grs/v1/tenant/{tenantId}/package/{packageId}/parameters
```

Request Body

N/A

Response Body

```
{
   "parms":[
      {
         "name":"age",
         "type":"integer",
         "constraints":"0::99",
         "keysAndValues":null
      },
      {
         "name":"education",
         "type":"list",
         "constraints":null,
         "keysAndValues":[
            {
               "key":"dropout",
               "value":"High School Drop-out"
            },
            {
               "key":"ged",
               "value":"G.E.D."
            },
            {
               "key":"hs",
               "value":"High School"
            },
            {
               "key":"associate",
               "value":"Associate Degree"
            },
            {
               "key":"bachelor",
               "value":"Bachelor Degree"
            },
            {
               "key":"master",
               "value":"Master's Degree"
            },
            {
               "key":"phd",
               "value":"phD"
            }
         ]
      }
      {
         "name":"zipCode",
         "type":"string",
         "constraints":">^\\d{5}$|^\\d{5}-\\d{4}$ ",
         "keysAndValues":null
      }
   ]
}
```

HTTP Status Codes

- 200 OK—Response body provided
- 401 Unauthorized—APIToken not valid
- 403 Forbidden—User does not have permission for the specified tenant ID.
- 404 Not Found—could not find specified package ID. Use Query Packages to obtain correct ID.
- 503—Service Unavailable

Notes

Returns a list of parameter information based on the combination of all templates/versions assigned to the specified rule package. For each parameter (eg, "age", "gender", etc) it returns the type, any constraints, and if a list of selectable values are returned, it returns the list of keys/values.

For constraints, integer/numeric parms can have lower/upper bounds rendered as: "constraints" : "0::99"

If there is no upper bound, lower bound 0: "constraints: : "0::"

No lower bound, upper bound 99: "constraints" : "::99"

Strings can have regex expressions such as: "constraints":">^\\d{5}$|^\\d{5}-\\d{4}$ ",

Rules

Query Rules In Package

Method

GET

Syntax

```
/grs/v1/tenant/{tenantId}/package/{packageId}/rules
```

Optional parameters:

- `&nodeId=457`
- `&baseSalience=99000`

Request Body

N/A

Response Body

```
{
   "rules":[
      {
         "ruleId":"f22d35f6-e31e-41f2-8e62-97864c47579a",
         "ruleType":"DECISION_TABLE",
         "extRuleId":"DT-163",
         "name":"DEF",
         "description":"Sample2",
         "phase":"2nd",
         "splitTest": "A",
         "calendarId":"Calendar_117",
         "nodeId":"pkg",
         "locked":false,
         "lockOwner":null,
         "dateEffective":null,
         "dateExpired":null,
         "pendingSnapshot":true
      },
      {
         "ruleId":"820a0434-9e46-4a44-b0e7-fea201a17282",
         "ruleType":"DECISION_TABLE",
         "extRuleId":"DT-186",
         "name":"3rd",
         "description":"",
         "phase":"3rd",
         "splitTest": "A",
         "calendarId":"none",
         "nodeId":"pkg",
         "locked":false,
         "lockOwner":null,
         "dateEffective":null,
         "dateExpired":null,
         "pendingSnapshot":true
      },
      {
         "ruleId":"a60416af-80a9-4378-be58-785c12b3ba8e",
         "ruleType":"LINEAR",
         "extRuleId":"Rule-100",
         "name":"1st Date",
         "description":"Basic criteria",
         "phase":"1st",
         "splitTest": "A",
         "calendarId":"none",
         "nodeId":"pkg",
         "locked":false,
         "lockOwner":null,
         "dateEffective":null,
         "dateExpired":null,
         "pendingSnapshot":true
      },
      {
         "ruleId":"061619bb-de47-4bda-80a1-2e1ade5fa010",
         "ruleType":"LINEAR",
         "extRuleId":"Rule-110",
         "name":"Default",
         "description":"Default is NO WAY BUDDY",
         "phase":"--none--",
         "splitTest": "A",
         "calendarId":"none",
         "nodeId":"pkg",
         "locked":false,
         "lockOwner":null,
         "dateEffective":null,
         "dateExpired":null,
         "pendingSnapshot":true
      },
      {
         "ruleId":"65cda7d5-09b4-47ec-9f7d-d90144be7f89",
         "ruleType":"LINEAR",
         "extRuleId":"Rule-130",
         "name":"Check day",
         "description":"No dating on holidays!",
         "phase":"--none--",
         "splitTest": "A",
         "calendarId":"Calendar_117",
         "nodeId":"pkg",
         "locked":false,
         "lockOwner":null,
         "dateEffective":null,
         "dateExpired":null,
         "pendingSnapshot":true
      }
   ]
```

```
}
```

HTTP Status Codes

- 200 OK—Response body provided
- 401 Unauthorized—APIToken not valid
- 403 Forbidden—User does not have permission for the specified tenant ID, or does not have the correct permission for this operation.
- 404 Not Found—could not find specified package ID. Use Query Packages to obtain correct ID.
- 503—Service Unavailable

Notes

User must have `RULE_VIEW` permission By default, this query returns all rules in the specified package. To further filter, specify: `&nodeId`, which returns only rules at that node within the rule package. Notes: If "nodeId" is set to "pkg", API will return only rules at the package level. For sub-node, set both "nodeId" and "baseSalience" values for the node being queried. Both of these are returned on the "Query Business Hierarchy" API.

Query Linear Rule Body

Method

GET

Syntax

```
/grs/v1/tenant/{tenantId}/package/{packageId}/linearrulebody/{ruleId}
```

Request Body

N/A

Response Body

```
{
   "ruleId":"a60416af-80a9-4378-be58-785c12b3ba8e",
   "ruleType":"LINEAR",
   "extRuleId":"Rule-100",
   "name":"DEF",
   "description":"Basic criteria",
   "phase":"1st",
   "calendarId":"Calendar_117",
   "nodeId":"pkg",
   "locked":false,
   "lockOwner":null,
   "dateEffective":null,
   "dateExpired":null,
   "pendingSnapshot":true,
   "conditions":[
      {
         "conditionType":"condition",
         "label":"Age is between ",
         "sentence":"Age is between {age_low} and {age_high}",
         "segmentList":[
            "\"{age_low}\"",
            " and ",
            "\"{age_high}\""
         ],
         "parmList":[
            "20",
            "and",
            "24"
         ]
      },
      {
         "conditionType":"condition",
         "label":"Education is ",
         "sentence":"Education is \"{education}\"",
         "segmentList":[
            "\"{education}\""
         ],
         "parmList":[
            "bachelor"
         ]
      },
      {
         "conditionType":"condition",
         "label":"Employed ",
         "sentence":"Employed {employed}",
         "segmentList":[
            "\"{employed}\""
         ],
         "parmList":[
            "true"
         ]
      },
      {
         "conditionType":"condition",
         "label":"Income is at least ",
         "sentence":"Income is at least {income} annually",
         "segmentList":[
            "\"{income}\"",
            " annually"
         ],
         "parmList":[
            "25000.00",
            "annually"
         ]
      },
      {
         "conditionType":"condition",
         "label":"Number of loans less than ",
         "sentence":"Number of loans less than {loans}",
         "segmentList":[
            "\"{loans}\""
         ],
         "parmList":[
            "3"
         ]
      },
      {
         "conditionType":"condition",
         "label":"Savings between ",
         "sentence":"Savings {savings_low} and {savings_high}",
         "segmentList":[
            "\"{savings_low}\"",
            " and ",
            "\"{savings_high}\""
         ],
         "parmList":[
            "160",
            "and",
            "200"
         ]
      }
   ],
   "actions":[
      {
         "label":"Decision is ",
         "sentence":"Decision is \"{decision}\"",
         "segmentList":[
            "\"{decision}\""
         ],
         "parmList":[
            "OK"
         ]
      },
      {
         "label":"Additional action: ",
         "sentence":"Additional action: \"{action}\"",
         "segmentList":[
            "\"{action}\""
         ],
         "parmList":[
            "Reference"
         ]
      }
   ]
}
```

HTTP Status Codes

- 200 OK—Response body provided
- 401 Unauthorized—APIToken not valid
- 403 Forbidden—User does not have permission for the specified tenant ID, or does not have the correct permission for this operation.
- 404 Not Found—could not find specified rule ID.
- 503 —Service Unavailable

Notes

User must have `RULE_VIEW` permission Optional Parameter: `&version=xxx` (specifies an earlier version of the rule) Description of the results:

- `conditionType`—Either "condition" or "label". Label is used for operators like "and", "or", "not"
- `sentence`—The original sentence from the template
- `segmentList`—The original parameters used (eg, `"{age}"`) and any intervening text separated in segments.
- `parmList`—The substituted variables (for example, `"20"`)

Note: Use Query Parameters to fetch the type of each parameter (eg, "age", "name", etc), any constraints and any drop-down values. This information is not repeated for each condition/action but can be queried separately and applied to control what the user is able to type for each value.

Query Decision Table Body

Method

GET

Syntax

```
/grs/v1/tenant/{tenantId}/package/{packageId}/decisiontablebody/{ruleId}
```

Request Body

N/A

Response Body

```
{
"ruleId": "2e11aaa7-12c4-4cf2-8a74-4dbfcf8ed76e",
"ruleType": "DECISION_TABLE",
"extRuleId": "DT-208",
"name": "test dt 2",
"description": "",
"phase": "1st",
"calendarId":"Calendar_117",
"nodeId": "pkg",
"salience": 100000,
"bcLevel": 1,
"locked": false,
"lockOwner": null,
"dateEffective": null,
"dateExpired": null,
"pendingSnapshot": true,
"conditions": [
{
"header": "Debt is less than",
"sentence": "Debt is less than {debt}",
"factType": "Debt is less than {debt}",
"factField": null,
"constraintValueType": 5,
"segmentList": ["\"{debt}\""]
},
{
"header": "Age is between",
"sentence": "Age is between {age_low} and {age_high}",
"factType": "Age is between {age_low} and {age_high}",
"factField": null,
"constraintValueType": 5,
"segmentList": [
"\"{age_low}\"",
" and ",
"\"{age_high}\""
]
}
],
"actions": [
{
"header": "Decision is",
"sentence": "Decision is \"{decision}\"",
"segmentList": ["\"{decision}\""]
},
{
"header": "Stop processing",
"sentence": "Stop processing",
"segmentList": ["Stop processing"]
}
],
"data": [ [
null,
"DTR-209",
"",
"33333",
"22",
"and",
"23",
"OK",
"Stop processing"
]]
}
```

HTTP Status Codes

- 200 OK—Response body provided
- 401 Unauthorized—APIToken not valid
- 403 Forbidden—User does not have permission for the specified tenant ID, or does not have the correct permission for this operation.
- 404 Not Found—could not find specified rule ID.
- 503 —Service Unavailable

Notes

User must have `RULE_VIEW` permission Optional Parameter:

- `&version=<`*xxx*`>` (specifies an earlier version of the rule)

Description of the results: This query will return the condition and action columns to be shown in a decision table, along with the substitution data that makes up each row. For both "condition" and "actions", the following is returned:

- `sentence`—The original sentence from the template: Health is "{health}"
- `segmentList`—The original parameters used (eg, `"{age}"`) and any intervening text separated in segments.

Example: `[ "Health is", "{health}" ]` **Note:** Use Query Parameters to fetch the type of each parameter (eg, "age", "name", etc), any constraints and any drop-down values. This information is not repeated for each condition/action but can be queried separately and applied to control what the user is able to type for each value. The actual decision table data is returned as a two-dimensional array representing the rows and columns of the decision table. The first column is always "NULL" as it represents the generated drools ID and is not modifiable by the user.

Create Rule

Method

POST

Syntax

```
/grs/v1/tenant/{tenantId}/package/{packageId}/rule
```

Request Body

```
{
         "ruleType":"DECISION_TABLE",
         "name":"My New Rule",
         "description":"My Description",
         "phase":"My Phase",
         "splitTest": "A"
         "calendarId":"Calendar_117",
         "nodeId":"pkg",
         "dateEffective":234324234334,
         "dateExpired":32434323434
}
```

Response Body

```
{
        "ruleId":"f22d35f6-e31e-41f2-8e62-97864c47579a",
        "extRuleId":"DT-163",
         "ruleType":"DECISION_TABLE",
         "name":"My New Rule",
         "description":"My Description",
         "phase":"My Phase",
         "splitTest": "A"
         "calendarId":"Calendar_117",
         "nodeId":"pkg",
         "dateEffective":234324234334,
         "dateExpired":32434323434
}
```

HTTP Status Codes

- 201 Created—Response body provided
- 401 - Unauthorized—APIToken not valid
- 403 - Forbidden—User does not have permission for the specified tenant ID, or does not have proper permission to perform this operation.
- 422 - Unprocessable Entity—Field Validation Error

```
{
  "message": "Validation Failed",
  "errors": [
    {
       "code" : "1000",
       "field" : "ruleType",
       "message": "ruleType must be either DECISION_TABLE or LINEAR"
    },
    {
       "code" : "1001",
       "field" : "phase",
       "message": "phase is not valid"
    },
    {
       "code" : "1002",
       "field" : "calendarId",
       "message": "calendarId is not valid"
    },
    {
       "code" : "1003",
       "field" : "nodeId",
       "message": "nodeId is not valid. Must be either pkg or a
node ID that is valid for this user"
    },
    {
       "code" : "1004",
       "field" : "date",
       "message": "dateExpired is before dateEffective"
    }
    {
       "code" : "1005",
       "field" : "splitTest",
       "message": "splitTest is not valid"
    }
  ]
}
* 503–Service Unavailable
```

Notes

User must have `RULE_CREATE` and `RULE_VIEW` permission

- `ruleType`—`DECISION_TABLE` or `LINEAR`
- `nodeId`—Specify the node ID where the rule should be created. This is returned in Query Business

Hierarchy. If the rule should be created at the package level, set this field to **`pkg`**. If the rule should be created in the Split Test Configuration section, specify **`splittest`**.

- `dateEffective`—null or timestamp (ms since EPOCH)
- `dateExpired`—null or timestamp (ms since EPOCH)

Lock Rule

Method

GET

Syntax

```
/grs/v1/tenant/{tenantId}/package/{packageId}/lockrule/{ruleId}
```

Request Body

N/A

Response Body

N/A

HTTP Status Codes

- 204 No Content—Request was successful
- 401 Unauthorized—APIToken not valid
- 403 Forbidden—User does not have permission for the specified tenant ID, or does not have the correct permission for this operation.
- 404 Not Found—could not find specified package ID or rule ID.
- 412 Precondition failed—another user holds the lock for this resource
- 503—Service Unavailable

Notes

Must have `RULE_VIEW` and `RULE_MODIFY` permission.

User can LOCK (obtain exclusive access) to a rule by issuing this API. The LOCK will be in effect until UNLOCKED by the user, or until the user's APIToken times out (session timeout, default 30 min). A lock is required to modify or delete the rule.

Unlock Rule

Method

GET

Syntax

```
/grs/v1/tenant/{tenantId}/package/{packageId}/unlockrule/{ruleId}
```

Request Body

N/A

Response Body

N/A

HTTP Codes

- 204 No content—Request was successful
- 401 Unauthorized—APIToken not valid
- 403 Forbidden—User does not have permission for the specified tenant ID, or does not have proper permission to perform this operation.
- 404 Not Found—could not find specified package ID or rule ID.
- 503—Service Unavailable

Notes

Unlocks a rule previously locked by this user (APIToken).

Modify Rule

Method

PUT

Syntax

```
/grs/v1/tenant/{tenantId}/package/{packageId}/rule/{ruleId}
```

Request Body

To modify something in rule summary (name, phase, etc):

```
{
         "name":"New Name",
         "description":"New Desc",
         "phase":"New Phase",
         "splitTest": "A"
         "calendarId":"Calendar_117",
         "dateEffective":99999999999,
         "dateExpired":99999999999
}
```

Response Body

```
{
        "ruleId":"f22d35f6-e31e-41f2-8e62-97864c47579a",
        "extRuleId":"DT-163",
         "ruleType":"DECISION_TABLE",
         "name":"New Name",
         "description":"New Description",
         "phase":"New Phase",
         "calendarId":"Calendar_117",
         "nodeId":"pkg",
         "dateEffective":99999999999,
         "dateExpired":99999999999
}}
```

HTTP Status Codes

- 200 Successful—Response body provided
- 401 Unauthorized—APIToken not valid
- 403 Forbidden—User does not have permission for the specified tenant ID, or does not have proper permission to perform this operation.
- 404 Not Found—could not find specified package ID or rule ID.
- 412 Precondition Failed— Rule was not locked first
- 422 Unprocessable Entity—Field Validation Error

```
{
  "message": "Validation Failed",
  "errors": [
    {
       "code" : "1000",
       "field" : "phase",
       "message": "phase is not valid"
    },
    {
       "code" : "1001",
       "field" : "calendarName",
       "message": "calendarName is not valid"
    },
    {
       "code" : "1002",
       "field" : "date",
       "message": "dateExpired is before dateEffective"
    }
    {
       "code" : "1005",
       "field" : "splitTest",
       "message": "splitTest is not valid"
    }
  ]
}
```

- 503—Service Unavailable

Notes

User must have `RULE_MODIFY` and `RULE_VIEW` permission. User must have rule LOCKED. The following fields can modified on the rule summary. Any other fields specified will be ignored:

- `name`
- `description`
- `phase`
- `calendar`

- `effective date`
- `expired date`

Modify LINEAR Rule Body

Method

PUT

Syntax

```
/grs/v1/tenant/{tenantId}/package/{packageId}/linearrulebody/{ruleId}<br/>
```

Optional parameter:

- ?checkInComment=My change (check in comment appears in audit tab and package history tab for the rule)

Request Body

```
{
  "conditions":[
      {
         "conditionType":"condition",
         "label":"Age is between ",
         "sentence":"Age is between {age_low} and {age_high}",
         "segmentList":[
            "\"{age_low}\"",
            " and ",
            "\"{age_high}\""
         ],
         "parmList":[
            "20",
            "and",
            "24"
         ]
      }
   ],
   "actions":[
      {
         "label":"Decision is ",
         "sentence":"Decision is \"{decision}\"",
         "segmentList":[
            "\"{decision}\""
         ],
         "parmList":[
            "OK"
         ]
      }
```

```
     ]
}
```

Response Body

```
{
   "ruleId":"a60416af-80a9-4378-be58-785c12b3ba8e",
   "ruleType":"LINEAR",
   "extRuleId":"Rule-100",
   "name":"1st Date",
   "description":"Basic criteria",
   "phase":"1st",
   "calendarId":"Calendar_117",
   "nodeId":"pkg",
   "locked":true,
   "lockOwner":user01,
   "dateEffective":null,
   "dateExpired":null,
   "pendingSnapshot":true,
   "conditions":[
      {
         "conditionType":"condition",
         "label":"Age is between ",
         "sentence":"Age is between {age_low} and {age_high}",
         "segmentList":[
            "\"{age_low}\"",
            " and ",
            "\"{age_high}\""
         ],
         "parmList":[
            "20",
            "and",
            "24"
         ]
      }
   ],
   "actions":[
      {
         "label":"Decision is ",
         "sentence":"Decision is \"{decision}\"",
         "segmentList":[
            "\"{decision}\""
         ],
         "parmList":[
            "OK"
         ]
      }
     ]
}
```

HTTP Status Codes

- 200 Successful—Response body provided
- 401 Unauthorized—APIToken not valid

- 403 Forbidden—User does not have permission for the specified tenant ID or package ID, or does not have proper permission to perform this operation.
- 404 Not Found—could not find specified rule ID.
- 412 Precondition Failed—Rule was not locked first
- 422 Unprocessable Entity—Rule does not validate

```
{
"message": "Validation Failed",
  "errors": [
    {
      "message": "Validation error 1"
    },
    {
      "message": "Validation error 2"
    }
  ]
}
```

- 503—Service Unavailable

Notes

User must have `RULE_MODIFY` permission. User must have rule LOCKED. To modify something in LINEAR rule body, pass entire rule body (all conditions/actions), regardless of whether they were updated or not (eg, this is a total replacement of rule body). If the change causes the rule to not validate, a 422 will be returned with a list of "compiler errors". The changes have still been committed.

Modify DECISION TABLE Rule Body

Method

PUT

Syntax

```
/grs/v1/tenant/{tenantId}/package/{packageId}/decisiontablebody/{ruleId}<br/>
```

Optional parameter:

- `?checkInComment=My change` (check in comment appears in audit tab and package history tab for the rule)

Notes

- **Warning**—To modify something in the Decision Table body, pass the entire rule body (all conditions/actions/data), regardless of what you updated (this is a total replacement of rule body). So even if you are only changing 1 row of 100, you must effectively replace the contents of all of the data. There is a risk of data loss if you do not submit the entire rule body. There is currently no API for deletion of a single Decision Table row. If the change causes the rule to not validate, a 422 will be returned with a list of "compiler errors". The changes have still been committed.
- User must have `RULE_MODIFY` and `RULE_VIEW` permissions.
- User must have the rule LOCKED.

Request Body

To modify condition/action columns and row data, pass in "conditions", "actions" and "data" elements. If "conditions" or "actions" are not passed, then no changes will be made. "data" must always be passed. If only "data" is changing (values changed, new rows inserted, other rows deleted, etc), just pass in the changed "data" structure (see below).

```
"conditions": [
{
"header": "Debt is less than",
"sentence": "Debt is less than {debt}",
"factType": "Debt is less than {debt}",
"factField": null,
"constraintValueType": 5,
"segmentList": ["\"{debt}\""]
},
{
"header": "Age is between",
"sentence": "Age is between {age_low} and {age_high}",
"factType": "Age is between {age_low} and {age_high}",
"factField": null,
"constraintValueType": 5,
"segmentList": [
"\"{age_low}\"",
" and ",
"\"{age_high}\""
]
}
],
"actions": [
{
"header": "Decision is",
"sentence": "Decision is \"{decision}\"",
"segmentList": ["\"{decision}\""]
},
{
"header": "Stop processing",
"sentence": "Stop processing",
"segmentList": ["Stop processing"]
}
],
"data": [ [
null,
"DTR-209",
"",
"33333",
"22",
"and",
"23",
"OK",
"Stop processing"
]]
}

Data changed only
{
"data": [ [
null,
"DTR-209",
"",
"44444",
"33",
"and",
"44",
"OK",
"Stop processing"
]]
}
```

Response Body

```
{
"ruleId": "2e11aaa7-12c4-4cf2-8a74-4dbfcf8ed76e",
"ruleType": "DECISION_TABLE",
"extRuleId": "DT-208",
"name": "test dt 2",
"description": "",
"phase": "1st",
"calendarId":"Calendar_117",
"nodeId": "pkg",
"salience": 100000,
"bcLevel": 1,
"locked": false,
"lockOwner": null,
"dateEffective": null,
"dateExpired": null,
"pendingSnapshot": true,
"conditions": [
{
"header": "Debt is less than",
"sentence": "Debt is less than {debt}",
"factType": "Debt is less than {debt}",
"factField": null,
"constraintValueType": 5,
"segmentList": ["\"{debt}\""]
},
{
"header": "Age is between",
"sentence": "Age is between {age_low} and {age_high}",
"factType": "Age is between {age_low} and {age_high}",
"factField": null,
"constraintValueType": 5,
"segmentList": [
"\"{age_low}\"",
" and ",
"\"{age_high}\""
]
}
],
"actions": [
{
"header": "Decision is",
"sentence": "Decision is \"{decision}\"",
"segmentList": ["\"{decision}\""]
},
{
"header": "Stop processing",
"sentence": "Stop processing",
"segmentList": ["Stop processing"]
}
],
"data": [ [
null,
"DTR-209",
"",
"33333",
"22",
"and",
"23",
"OK",
"Stop processing"
]]
}
```

HTTP Status Codes

- 200 Successful—Response body provided
- 401 Unauthorized—APIToken not valid
- 403 Forbidden—User does not have permission for the specified tenant ID or package ID, or does not have proper permission to perform this operation.
- 404 Not Found—could not find specified rule ID.
- 412 Precondition Failed—Rule was not locked first
- 422 Unprocessable Entity—Rule does not validate

```
{
"message": "Validation Failed",
  "errors": [
    {
      "message": "Validation error 1"
    },
    {
      "message": "Validation error 2"
    }
  ]
}
```

- 503—Service Unavailable

Delete Rule

Method

DELETE

Syntax

```
/grs/v1/tenant/{tenantId}/package/{packageId}/rule/{ruleId}
```

Request Body

N/A

Response Body

N/A

HTTP Status Codes

- 204 No Content—returned if successful operation
- 401 Unauthorized—APIToken not valid
- 403 Forbidden—User does not have permission for the specified tenant or does not have proper permission to perform this operation.
- 404 Not Found—could not find specified package ID or rule ID.
- 412 Precondition Failed—Rule was not locked first
- 503—Service Unavailable

Notes

User must have `RULE_DELETE` and `RULE_VIEW` permission. User must have rule LOCKED.

Validate Rule

Method

GET

Syntax

```
/grs/v1/tenant/{tenantId}/package/{packageId}/validaterule/{ruleId}
```

Request Body

N/A

Response Body

```
{
"message": "Validation Failed",
  "errors": [
    {
      "message": "Validation error 1"
    },
    {
      "message": "Validation error 2"
    }
  ]
}
```

HTTP Status Codes

- 204 No content—Rule validates successfully
- 401 Unauthorized—APIToken not valid
- 403 Forbidden—User does not have permission for the specified tenant ID, or does not have proper permission to perform this operation.
- 404 Not Found—could not find specified rule ID.

- 422 Unprocessable Entity—Rule does not validate

```
{
"message": "Validation Failed",
  "errors": [
    {
      "message": "Validation error 1"
    },
    {
      "message": "Validation error 2"
    }
  ]
}
```

- 503—Service Unavailable

Notes

User must have `RULE_VIEW` permission.

Business Calendars

API	Method	Syntax	Request Body	Response Body	HTTP Status Codes	Notes

Query Business Calendars	GET	**[+]Show**
		`/grs/v1/tenant/{tenantId}/package/{packageId}/calendars`
			N/A	**[+]Show**
				`"calendars": [`
				`{`
				`"calendarId":`
				`"dcc4a39b-3442-4408-a907-b85c44663681",`
				`"extCalendarId":`
				`"Calendar_117",`
				`"name": "Dad's Dating Calendar",`
				`"locked": false,`
				`"lockOwner": null,`
				`"weekStarts": 6,`
				`"weekEnds": 7,`
				`"dayStartsHour": 17,`
				`"dayStartsMin": 30,`
				`"dayEndsHour": 23,`
				`"dayEndsMin": 0,`
				`"timeZoneId": "EST"`
				`},`
				`{`
				`"calendarId":`
				`"fdc4a39b-3442-4408-a907-b85c44663622",`
				`"extCalendarId":`
				`"Calendar_190",`
				`"name": "Another Calendar",`
				`"locked": false,`
				`"lockOwner": null,`
				`"weekStarts": 1,`
				`"weekEnds": 7,`
					• 200 OK—Response body provided
					• 401 Unauthorized—APIToken not valid
					• 403 Forbidden—User does not have permission for the specified tenant ID, or does not have the correct permission for this operation.
					• 404 Not Found—could not find specified package ID.
					• 503—Service Unavailable
						Need `CALENDAR_VIEW` permission.

API	Method	Syntax	Request Body	Response Body	HTTP Status Codes	Notes
				"dayStartsHour": 8, "dayStartsMin": 30, "dayEndsHour": 11, "dayEndsMin": 0, "timeZoneId": "America/Cayman" }]}		
Query Business Calendar Body	GET	**[+]Show** /grs/v1/tenant/{tenantId}/package/{packageId}/businesscalendarbody/{calendarId}	N/A	**[+]Show** { "calendarId":"dcc4a39b-3442-4408-a907-b85c44663681", "externalId":"Calendar_117", "name":"Dad's Dating Calendar", "locked":false, "lockOwner":null, "weekStarts":6, "weekEnds":7, "dayStartsHour":17, "dayStartsMin":30, "dayEndsHour":23, "dayEndsMin":0,	• 200 OK—Response body provided • 401 Unauthorized—APIToken not valid • 403 Forbidden—User does not have permission for the specified tenant ID, or does not have the correct permission for this operation. • 404 Not Found—could not find specified	For exceptions, the following applies: type • HOLIDAY (entire day is non-work) • TIME_CHANGE (working range specified) placement • FIXED - A single date (Jan 1, 2015) • fixedDate contains timestamp • ANNUAL - Same day each year

API	Method	Syntax	Request Body	Response Body	HTTP Status Codes	Notes
				"timeZoneId":"EST", "exceptions":[{ "name":"Christmas", "type":"HOLIDAY", "placement":"FIXED", "fixedDate":30949200000, "annualMonth":0, "annualDay":0, "relativeType":null, "relativeDayOfWeek":0, "relativeMonth":0, "timeChangeStartHour":0, "timeChangeStartMin":0, "timeChangeStopHour":0, "timeChangeStopMin":0 }, { "name":"It's New Year's Day",	calendar ID - 503—Service Unavailable	(Dec 25) - annualMonth: 1-12 - annualDay: 1-31 - RELATIVE - 1st Thursday of each month - relativeType - EVERY - FIRST - SECOND - THIRD - FOURTH - LAST - relativeDayOfWeek: 1-7 (1=Sun) - relative Month: 1-31, 0 if all months For TIME_CHANGE:

API	Method	Syntax	Request Body	Response Body	HTTP Status Codes	Notes
			"type":"HOLIDAY", "placement":"ANNUAL", "fixedDate":null, "annualMonth":1, "annualDay":1, "relativeType":null, "relativeDayOfWeek":0, "relativeMonth":0, "timeChangeStartHour":0, "timeChangeStartMin":0, "timeChangeStopHour":0, "timeChangeStopMin":0 }, { "name":"Thanksgiving", "type":"HOLIDAY", "placement":"RELATIVE", "fixedDate":null, "annualMonth":0, "annualDay":0,		• timeChangeStartHour: 0-23 • timeChangeStartMin: 0-59 • timeChangeStopHour: 0-23 • timeChangeStopMin: 0-59

API	Method	Syntax	Request Body	Response Body	HTTP Status Codes	Notes
				"relativeType":"FOURTH", "relativeDayOfWeek":5, "relativeMonth":11, "timeChangeStartHour":0, "timeChangeStartMin":0, "timeChangeStopHour":0, "timeChangeStopMin":0 }, { "name":"My Dad's birthday !", "type":"HOLIDAY", "placement":"ANNUAL", "fixedDate":null, "annualMonth":5, "annualDay":20, "relativeType":null, "relativeDayOfWeek":0, "relativeMonth":0, "timeChangeStartHour":0,		

API	Method	Syntax	Request Body	Response Body	HTTP Status Codes	Notes
				"timeChangeStartMin":0, "timeChangeStopHour":0, "timeChangeStopMin":0 }, { "name":"A Time Change", "type":"TIME_CHANGE", "placement":"ANNUAL", "fixedDate":null, "annualMonth":1, "annualDay":9, "relativeType":null, "relativeDayOfWeek":0, "relativeMonth":0, "timeChangeStartHour":12, "timeChangeStartMin":30, "timeChangeStopHour":16, "timeChangeStopMin":22 }, {		

API	Method	Syntax	Request Body	Response Body	HTTP Status Codes	Notes
				"name":"tc3", "type":"TIME_CHANGE", "placement":"FIXED", "fixedDate":1428984000000, "annualMonth":0, "annualDay":0, "relativeType":null, "relativeDayOfWeek":0, "relativeMonth":0, "timeChangeStartHour":1, "timeChangeStartMin":33, "timeChangeStopHour":7, "timeChangeStopMin":33 }, { "name":"TC2", "type":"TIME_CHANGE", "placement":"RELATIVE", "fixedDate":null, "annualMonth":0,		

API	Method	Syntax	Request Body	Response Body	HTTP Status Codes	Notes
				"annualDay":0, "relativeType":"EVERY", "relativeDayOfWeek":1, "relativeMonth":1, "timeChangeStartHour":3, "timeChangeStartMin":0, "timeChangeStopHour":19, "timeChangeStopMin":0 }] }		
Create Calendar	POST	**[+]Show** /grs/v1/tenant/{tenantId}/package/{packageId}/calendar	**[+]Show** { "name": "Dad's Calendar", "weekStarts": 6, "weekEnds": 7, "dayStartsHour": 17, "dayStartsMin": 30, "dayEndsHour": 23, "dayEndsMin": 0, "timeZoneId": "EST"	**[+]Show** { "calendarId": "Calendar_117", "name": "Dad's Calendar", "locked": false, "lockOwner": null, "weekStarts": 6, "weekEnds": 7, "dayStartsHour": 17, "dayStartsMin": 30,	• 201 Created—Response body provided • 401 Unauthorized—APIToken not valid • 403 Forbidden—User does not have permission for the specified tenant, or does not have proper	User must have CALENDAR_CREATE and CALENDAR_VIEW permissions. Response will include calenderId which can be used on subsequent requests. WeekStarts / WeekEnds 1 - Sunday 2 - Monday ... 7 - Saturday

API	Method	Syntax	Request Body	Response Body	HTTP Status Codes	Notes
			}	"dayEndsHour": 23, "dayEndsMin": 0, "timeZoneId": "EST" }	permission to perform this operation. • 422 Unprocessable Entity—Field Validation Error **[+]Show** { "message": "Validation Failed", "errors": [{ "code" : "1000", "field" : "name", "message": "name must be specified" }, { "code" : "1001", "field" : "weekStarts", "message": "weekStarts is invalid. Must be between 1 and 7."	dayStartsHour : 0-23 dayStartsMin : 0-59 dayEndsHour : 0-23 dayEndsMin : 0-59 timeZoneId - As returned by Java's TimeZone class

API	Method	Syntax	Request Body	Response Body	HTTP Status Codes	Notes
					}, { "code" : "1002", "field" : "weekEnds", "message": "weekEnds is invalid. Must be between 1 and 7." }, { "code" : "1003", "field" : "dayStartsHour", "message": "dayStartsHour is invalid. Must be between 0 and 23." }, { "code" : "1004", "field" : "dayEndsHour", "message": "dayEndsHour is invalid. Must be between 0 and 23." }, {	

API	Method	Syntax	Request Body	Response Body	HTTP Status Codes	Notes

					"code" : "1005",
					"field" : "dayStartsMin",
					"message": "dayStartsMin is invalid. Must be between 0 and 59."
					},
					{
					"code" : "1006",
					"field" : "dayEndsMin",
					"message": "dayEndsMin is invalid. Must be between 0 and 59."
					},
					{
					"code" : "1007",
					"field" : "timeZoneID",
					"message": "Unrecognized time zone ID."
					}
					]
					}
					• 503—Service	

API	Method	Syntax	Request Body	Response Body	HTTP Status Codes	Notes
					Unavailable	
Lock Calendar	GET	**[+]Show** `/grs/v1/tenant/{tenantId}/package/{packageId}/lockcalendar/{calendarId}`	N/A	N/A	• 204 No Content—Request was successful • 401 Unauthorized—APIToken not valid • 403 Forbidden—User does not have permission for the specified tenant ID, or does not have the correct permission for this operation. • 404 Not Found—could not find specified package ID or calendar ID. • 412 Precondition failed—another user holds the lock for this resource	User must have `CALENDAR_VIEW` and `CALENDAR_MODIFY` permissions. User can LOCK (obtain exclusive access) to a calendar by issuing this API. The LOCK will be in effect until UNLOCKED by the user, or until the user's APIToken times out (session timeout, default 30 min). A lock is required to modify or delete the calendar.

API	Method	Syntax	Request Body	Response Body	HTTP Status Codes	Notes
					- 503—Service Unavailable	
Unlock Calendar	GET	**[+]Show** `/grs/v1/tenant/{tenantId}/package/{packageId}/unlockcalendar/{calendarId}`	N/A	N/A	- 204 No Content—Request was successful - 401 Unauthorized—APIToken not valid - 403 Forbidden—User does not have permission for the specified tenant ID. - 404 Not Found—could not find specified package ID or calendar ID. - 503—Service Unavailable	Unlocks a calendar previously locked by this user (APIToken).

Modify Calendar	PUT	**[+]Show**
		`/grs/v1/tenant/{tenantId}/package/{packageId}/calendar/{calendarId}`
			[+]Show
			`{`
			`"weekStarts": 4,`
			`"weekEnds": 5,`
				[+]Show
				`{`
				`"calendarId":`
				`"Calendar_117",`
					- 201 Created—Response body provided
					- 401
						User must have `CALENDAR_MODIFY` permission. User must have LOCK. The following fields can be updated on a calendar (any

API	Method	Syntax	Request Body	Response Body	HTTP Status Codes	Notes

			`"dayStartsHour": 14,`
			`"dayEndsMin": 30,`
			`}`
				`"name": "Dad's Dating Calendar",`
				`"locked": false,`
				`"lockOwner": null,`
				`"weekStarts": 4,`
				`"weekEnds": 5,`
				`"dayStartsHour": 14,`
				`"dayStartsMin": 30,`
				`"dayEndsHour": 23,`
				`"dayEndsMin": 30,`
				`"timeZoneId": "EST"`
				`}`
					Unauthorized—APIToken not valid
					- 403 Forbidden—User does not have permission for the specified tenant, or does not have proper permission to perform this operation.
					- 412 Precondition Failed—Calendar was not locked first
					- 422 - Unprocessable Entity—Field Validation Error
					[+]Show
					`{ "message": "Validation Failed", "errors": [ { "code" : "1000", "field" :`
						other fields passed in will be ignored):
						- name
						- weekStarts
						- weekEnds
						- dayStartsHour
						- dayStartsMin
						- dayEndsHour
						- dayEndsMin
						- timeZoneId

API	Method	Syntax	Request Body	Response Body	HTTP Status Codes	Notes
					"weekStarts", "message": "weekStarts is invalid. Must be between 1 and 7." }, { "code" : "1001", "field" : "weekEnds", "message": "weekEnds is invalid. Must be between 1 and 7." }, { "code" : "1002", "field" : "dayStartsHour", "message": "dayStartsHour is invalid. Must be between 0 and 23." }, { "code" : "1003", "field" : "dayEndsHour", "message":	

API	Method	Syntax	Request Body	Response Body	HTTP Status Codes	Notes
					"dayEndsHour is invalid. Must be between 0 and 23." }, { "code" : "1004", "field" : "dayStartsMin", "message": "dayStartsMin is invalid. Must be between 0 and 59." }, { "code" : "1005", "field" : "dayEndsMin", "message": "dayEndsMin is invalid. Must be between 0 and 59." }, { "code" : "1006", "field" : "timeZoneID", "message": "Unrecognized	

API	Method	Syntax	Request Body	Response Body	HTTP Status Codes	Notes

					`time zone ID."`
					`    }`
					`  ]`
					`}`
					- 503—Service Unavailable	

Modify Calendar Body	PUT	**[+]Show**
		/grs/v1/tenant/{tenantId}/package/{packageId}/businesscalendarbody/{calendarId}
			[+]Show
			`{`
			`"exceptions":[`
			`      {`
			`"name":"Christmas",`
			`"type":"HOLIDAY",`
			`"placement":"FIXED",`
			`"fixedDate":309492000000,`
			`"annualMonth":0,`
			`"annualDay":0,`
			`"relativeType":null,`
			`"relativeDayOfWeek":0,`
			`"relativeMonth":0,`
			`"timeChangeStartHour":0,`
			`"timeChangeStartMin":0,`
				[+]Show
				`{`
				`"calendarId":`
				`"Calendar_117",`
				`"name": "Dad's`
				`Dating Calendar",`
				`"locked": false,`
				`"lockOwner":`
				`null,`
				`"weekStarts": 4,`
				`"weekEnds": 5,`
				`"dayStartsHour":`
				`10,`
				`"dayStartsMin":`
				`30,`
				`"dayEndsHour":`
				`23,`
				`"dayEndsMin": 30,`
				`"timeZoneId":`
				`"EST",`
				`"exceptions":[`
				`      {`
				`"name":"Christmas",`
				`"type":"HOLIDAY",`
					- 201 Created—Response body provided
					- 401 Unauthorized—APIToken not valid
					- 403 Forbidden—User does not have permission for the specified tenant, or does not have proper permission to perform this operation.
					- 412 Precondition Failed—Calendar was not locked first
					- 422 Unprocessable Entity—Field
						User must have `CALENDAR_MODIFY` permission. User must have LOCK. User should pass in structure containing entire calendar body (exceptions).
						This is a complete replacement of current calendar body (not just changes). Updated structure is returned in response

API	Method	Syntax	Request Body	Response Body	HTTP Status Codes	Notes

			"timeChangeStopHour":0,
			"timeChangeStopMin":0
			},
			{
			"name":"It's New Year's Day",
			"type":"HOLIDAY",
			"placement":"ANNUAL",
			"fixedDate":null,
			"annualMonth":1,
			"annualDay":1,
			"relativeType":null,
			"relativeDayOfWeek":0,
			"relativeMonth":0,
			"timeChangeStartHour":0,
			"timeChangeStartMin":0,
			"timeChangeStopHour":0,
			"timeChangeStopMin":0
			}
			}
				"placement":"FIXED",
				"fixedDate":30949200000,
				"annualMonth":0,
				"annualDay":0,
				"relativeType":null,
				"relativeDayOfWeek":0,
				"relativeMonth":0,
				"timeChangeStartHour":0,
				"timeChangeStartMin":0,
				"timeChangeStopHour":0,
				"timeChangeStopMin":0
				},
				{
				"name":"It's New Year's Day",
				"type":"HOLIDAY",
				"placement":"ANNUAL",
				"fixedDate":null,
				"annualMonth":1,
				"annualDay":1,
					Validation Error
					[+]Show
					{
					"message": "Validation Failed",
					"errors": [
					{
					"code" : "1000",
					"field" : "type",
					"message": "Invalid type. Must be HOLIDAY or TIME_CHANGE"
					},
					{
					"code" : "1001",
					"field" : "placement",
					"message": "Invalid placement. Must be ANNUAL, FIXED or RELATIVE"
					}
					]
					}
					Repeated for each field:	

API	Method	Syntax	Request Body	Response Body	HTTP Status Codes	Notes
				"relativeType":null, "relativeDayOfWeek":0, "relativeMonth":0, "timeChangeStartHour":0, "timeChangeStartMin":0, "timeChangeStopHour":0, "timeChangeStopMin":0 } }	• 1002—fixedDate—fixedDate must be specified with type of FIXED • 1003—annualMonth—Invalid value. Must be between 1 and 12 • 1004—annualDay—Invalid value. Must be between 1 and 31. • 1005—relativeType—Invalid value, must be EVERY, FIRST, SECOND, THIRD, FOURTH, or LAST • 1006—relativeDayOfWeek—Invalid value. Must be between 1 and 7 • 1007—relativeMonth—Invalid value. Must be between 1 and	

API	Method	Syntax	Request Body	Response Body	HTTP Status Codes	Notes
					12 - 1008—timeChangeStartHour—Invalid value. Must be between 0 and 23 - 1009—timeChangeStartMin—Invalid value. Must be between 0 and 59 - 1010—timeChangeStopHour—Invalid value. Must be between 0 and 23 - 1011—timeChangeStopMin—Invalid value. Must be between 0 and 59 - 503—Service Unavailable	
Delete Calendar	DELETE	**[+]Show**	N/A	N/A	- 204 No	Must have CALENDAR_DELETE permission. Must

API	Method	Syntax	Request Body	Response Body	HTTP Status Codes	Notes
		/grs/v1/ tenant/{tenantId}/package/{packageId}/calendar/{calendarId}			Content—returned if successful operation • 401 Unauthorized—APIToken not valid • 403 Forbidden—User does not have permission for the specified tenant or does not have proper permission to perform this operation. • 404 Not Found—could not find specified package ID or calendarID. • 412 Precondition Failed—Calendar was not locked first • 503—Service Unavailable	have lock on Calendar.

Query Business Calendars

Method

GET

Syntax

```
/grs/v1/tenant/{tenantId}/package/{packageId}/calendars
```

Request Body

N/A

Response Body

```
"calendars": [
{
"calendarId": "dcc4a39b-3442-4408-a907-b85c44663681",
"extCalendarId": "Calendar_117",
"name": "Dad's Dating Calendar",
"locked": false,
"lockOwner": null,
"weekStarts": 6,
"weekEnds": 7,
"dayStartsHour": 17,
"dayStartsMin": 30,
"dayEndsHour": 23,
"dayEndsMin": 0,
"timeZoneId": "EST"
},
{
"calendarId": "fdc4a39b-3442-4408-a907-b85c44663622",
"extCalendarId": "Calendar_190",
"name": "Another Calendar",
"locked": false,
"lockOwner": null,
"weekStarts": 1,
"weekEnds": 7,
"dayStartsHour": 8,
"dayStartsMin": 30,
"dayEndsHour": 11,
"dayEndsMin": 0,
"timeZoneId": "America/Cayman"
}
]}
```

HTTP Status Codes

- 200 OK—Response body provided
- 401 Unauthorized—APIToken not valid
- 403 Forbidden—User does not have permission for the specified tenant ID, or does not have the correct permission for this operation.
- 404 Not Found—could not find specified package ID.
- 503—Service Unavailable

Notes

Need `CALENDAR_VIEW` permission.

Query Business Calendar Body

Method

GET

Syntax

```
/grs/v1/tenant/{tenantId}/package/{packageId}/businesscalendarbody/{calendarId}
```

Request Body

N/A

Response Body

```
{
   "calendarId":"dcc4a39b-3442-4408-a907-b85c44663681",
   "externalId":"Calendar_117",
   "name":"Dad's Dating Calendar",
   "locked":false,
   "lockOwner":null,
   "weekStarts":6,
   "weekEnds":7,
   "dayStartsHour":17,
   "dayStartsMin":30,
   "dayEndsHour":23,
   "dayEndsMin":0,
   "timeZoneId":"EST",
   "exceptions":[
      {
         "name":"Christmas",
         "type":"HOLIDAY",
         "placement":"FIXED",
         "fixedDate":30949200000,
         "annualMonth":0,
         "annualDay":0,
         "relativeType":null,
         "relativeDayOfWeek":0,
         "relativeMonth":0,
         "timeChangeStartHour":0,
         "timeChangeStartMin":0,
         "timeChangeStopHour":0,
         "timeChangeStopMin":0
      },
      {
         "name":"It's New Year's Day",
         "type":"HOLIDAY",
         "placement":"ANNUAL",
         "fixedDate":null,
         "annualMonth":1,
         "annualDay":1,
         "relativeType":null,
         "relativeDayOfWeek":0,
         "relativeMonth":0,
         "timeChangeStartHour":0,
         "timeChangeStartMin":0,
         "timeChangeStopHour":0,
         "timeChangeStopMin":0
      },
      {
         "name":"Thanksgiving",
         "type":"HOLIDAY",
         "placement":"RELATIVE",
         "fixedDate":null,
         "annualMonth":0,
         "annualDay":0,
         "relativeType":"FOURTH",
         "relativeDayOfWeek":5,
         "relativeMonth":11,
         "timeChangeStartHour":0,
         "timeChangeStartMin":0,
         "timeChangeStopHour":0,
         "timeChangeStopMin":0
      },
      {
         "name":"My Dad's birthday !",
         "type":"HOLIDAY",
         "placement":"ANNUAL",
         "fixedDate":null,
         "annualMonth":5,
         "annualDay":20,
         "relativeType":null,
         "relativeDayOfWeek":0,
         "relativeMonth":0,
         "timeChangeStartHour":0,
         "timeChangeStartMin":0,
         "timeChangeStopHour":0,
         "timeChangeStopMin":0
      },
      {
         "name":"A Time Change",
         "type":"TIME_CHANGE",
         "placement":"ANNUAL",
         "fixedDate":null,
         "annualMonth":1,
         "annualDay":9,
         "relativeType":null,
         "relativeDayOfWeek":0,
         "relativeMonth":0,
         "timeChangeStartHour":12,
         "timeChangeStartMin":30,
         "timeChangeStopHour":16,
         "timeChangeStopMin":22
      },
      {
         "name":"tc3",
         "type":"TIME_CHANGE",
         "placement":"FIXED",
         "fixedDate":1428984000000,
         "annualMonth":0,
         "annualDay":0,
         "relativeType":null,
         "relativeDayOfWeek":0,
         "relativeMonth":0,
         "timeChangeStartHour":1,
         "timeChangeStartMin":33,
         "timeChangeStopHour":7,
         "timeChangeStopMin":33
      },
      {
         "name":"TC2",
         "type":"TIME_CHANGE",
         "placement":"RELATIVE",
         "fixedDate":null,
         "annualMonth":0,
         "annualDay":0,
         "relativeType":"EVERY",
         "relativeDayOfWeek":1,
         "relativeMonth":1,
         "timeChangeStartHour":3,
         "timeChangeStartMin":0,
         "timeChangeStopHour":19,
         "timeChangeStopMin":0
      }
   ]
}
```

HTTP Status Codes

- 200 OK—Response body provided
- 401 Unauthorized—APIToken not valid
- 403 Forbidden—User does not have permission for the specified tenant ID, or does not have the correct permission for this operation.
- 404 Not Found—could not find specified calendar ID
- 503—Service Unavailable

Notes

For exceptions, the following applies:
type

- HOLIDAY (entire day is non-work)
- TIME_CHANGE (working range specified)

placement

- `FIXED` - A single date (Jan 1, 2015)
    - `fixedDate` contains `timestamp`
- `ANNUAL` - Same day each year (Dec 25)
    - `annualMonth: 1-12`
    - `annualDay: 1-31`
- `RELATIVE` - 1st Thursday of each month
    - `relativeType`
        - `EVERY`
        - `FIRST`
        - `SECOND`
        - `THIRD`
        - `FOURTH`
        - `LAST`
    - `relativeDayOfWeek: 1-7` (1=Sun)
        - `relativeMonth: 1-31`, 0 if all months

For `TIME_CHANGE`:

- `timeChangeStartHour: 0-23`
- `timeChangeStartMin: 0-59`
- `timeChangeStopHour: 0-23`
- `timeChangeStopMin: 0-59`

Create Calendar

Method

POST

Syntax

```
/grs/v1/tenant/{tenantId}/package/{packageId}/calendar
```

Request Body

```
{
"name": "Dad's Calendar",
"weekStarts": 6,
"weekEnds": 7,
"dayStartsHour": 17,
"dayStartsMin": 30,
"dayEndsHour": 23,
"dayEndsMin": 0,
"timeZoneId": "EST"
}
```

Response Body

```
{
"calendarId": "Calendar_117",
"name": "Dad's Calendar",
"locked": false,
"lockOwner": null,
"weekStarts": 6,
"weekEnds": 7,
"dayStartsHour": 17,
"dayStartsMin": 30,
"dayEndsHour": 23,
"dayEndsMin": 0,
"timeZoneId": "EST"
}
```

HTTP Status Codes

- 201 Created—Response body provided
- 401 Unauthorized—APIToken not valid
- 403 Forbidden—User does not have permission for the specified tenant, or does not have proper permission to perform this operation.
- 422 Unprocessable Entity—Field Validation Error

```
{
  "message": "Validation Failed",
  "errors": [
    {
       "code" : "1000",
       "field" : "name",
       "message": "name must be specified"
    },
    {
       "code" : "1001",
       "field" : "weekStarts",
       "message": "weekStarts is invalid. Must be between 1 and 7."
    },
    {
       "code" : "1002",
       "field" : "weekEnds",
       "message": "weekEnds is invalid. Must be between 1 and 7."
    },
    {
       "code" : "1003",
       "field" : "dayStartsHour",
       "message": "dayStartsHour is invalid. Must be between 0 and 23."
    },
    {
       "code" : "1004",
       "field" : "dayEndsHour",
       "message": "dayEndsHour is invalid. Must be between 0 and 23."
    },
   {
       "code" : "1005",
       "field" : "dayStartsMin",
       "message": "dayStartsMin is invalid. Must be between 0 and 59."
    },
   {
       "code" : "1006",
       "field" : "dayEndsMin",
       "message": "dayEndsMin is invalid. Must be between 0 and 59."
    },
 {
       "code" : "1007",
       "field" : "timeZoneID",
       "message": "Unrecognized time zone ID."
    }
  ]
}
```

- 503—Service Unavailable

Notes

User must have `CALENDAR_CREATE` and `CALENDAR_VIEW` permissions. Response will include `calenderId` which can be used on subsequent requests. WeekStarts / WeekEnds 1 - Sunday
2 - Monday
...
7 - Saturday

dayStartsHour : 0-23
dayStartsMin : 0-59
dayEndsHour : 0-23
dayEndsMin : 0-59
timeZoneId - As returned by Java's TimeZone class

Lock Calendar

Method

GET

Syntax

```
/grs/v1/tenant/{tenantId}/package/{packageId}/lockcalendar/{calendarId}
```

Request Body

N/A

Response Body

N/A

HTTP Status Codes

- 204 No Content—Request was successful
- 401 Unauthorized—APIToken not valid
- 403 Forbidden—User does not have permission for the specified tenant ID, or does not have the correct permission for this operation.
- 404 Not Found—could not find specified package ID or calendar ID.
- 412 Precondition failed—another user holds the lock for this resource
- 503—Service Unavailable

Notes

User must have `CALENDAR_VIEW` and `CALENDAR_MODIFY` permissions. User can LOCK (obtain exclusive access) to a calendar by issuing this API. The LOCK will be in effect until UNLOCKED by the user, or until the user's APIToken times out (session timeout, default 30 min). A lock is required to modify or delete the calendar.

Unlock Calendar

Method

GET

Syntax

```
/grs/v1/tenant/{tenantId}/package/{packageId}/unlockcalendar/{calendarId}
```

Request Body

N/A

Request Body

N/A

HTTP Status Codes

- 204 No Content—Request was successful
- 401 Unauthorized—APIToken not valid
- 403 Forbidden—User does not have permission for the specified tenant ID.
- 404 Not Found—could not find specified package ID or calendar ID.
- 503—Service Unavailable

Notes

Unlocks a calendar previously locked by this user (APIToken).

Modify Calendar

Method

PUT

Syntax

```
/grs/v1/tenant/{tenantId}/package/{packageId}/calendar/{calendarId}
```

Request Body

```
{
"weekStarts": 4,
"weekEnds": 5,
"dayStartsHour": 14,
"dayEndsMin": 30,
}
```

Response Body

```
{
"calendarId": "Calendar_117",
"name": "Dad's Dating Calendar",
"locked": false,
"lockOwner": null,
"weekStarts": 4,
"weekEnds": 5,
"dayStartsHour": 14,
"dayStartsMin": 30,
"dayEndsHour": 23,
"dayEndsMin": 30,
"timeZoneId": "EST"
}
```

HTTP Status Codes

- 201 Created—Response body provided

- 401 Unauthorized—APIToken not valid
- 403 Forbidden—User does not have permission for the specified tenant, or does not have proper permission to perform this operation.
- 412 Precondition Failed—Calendar was not locked first
- 422 - Unprocessable Entity—Field Validation Error

```
{
  "message": "Validation Failed",
  "errors": [
    {
       "code" : "1000",
       "field" : "weekStarts",
       "message": "weekStarts is invalid. Must be between 1 and 7."
    },
    {
       "code" : "1001",
       "field" : "weekEnds",
       "message": "weekEnds is invalid. Must be between 1 and 7."
    },
    {
       "code" : "1002",
       "field" : "dayStartsHour",
       "message": "dayStartsHour is invalid. Must be between 0 and 23."
    },
    {
       "code" : "1003",
       "field" : "dayEndsHour",
       "message": "dayEndsHour is invalid. Must be between 0 and 23."
    },
   {
       "code" : "1004",
       "field" : "dayStartsMin",
       "message": "dayStartsMin is invalid. Must be between 0 and 59."
    },
   {
       "code" : "1005",
       "field" : "dayEndsMin",
       "message": "dayEndsMin is invalid. Must be between 0 and 59."
    },
 {
       "code" : "1006",
       "field" : "timeZoneID",
       "message": "Unrecognized time zone ID."
    }
  ]
}
```

- 503—Service Unavailable

Notes

User must have `CALENDAR_MODIFY` permission. User must have LOCK. The following fields can be updated on a calendar (any other fields passed in will be ignored):

- `name`
- `weekStarts`
- `weekEnds`
- `dayStartsHour`
- `dayStartsMin`
- `dayEndsHour`
- `dayEndsMin`
- `timeZoneId`

Modify Calendar Body

Method

PUT

Syntax

```
/grs/v1/tenant/{tenantId}/package/{packageId}/businesscalendarbody/{calendarId}
```

Request Body

```
{
   "exceptions":[
      {
         "name":"Christmas",
         "type":"HOLIDAY",
         "placement":"FIXED",
         "fixedDate":30949200000,
         "annualMonth":0,
         "annualDay":0,
         "relativeType":null,
         "relativeDayOfWeek":0,
         "relativeMonth":0,
         "timeChangeStartHour":0,
         "timeChangeStartMin":0,
         "timeChangeStopHour":0,
         "timeChangeStopMin":0
      },
      {
         "name":"It's New Year's Day",
         "type":"HOLIDAY",
         "placement":"ANNUAL",
         "fixedDate":null,
         "annualMonth":1,
         "annualDay":1,
         "relativeType":null,
         "relativeDayOfWeek":0,
         "relativeMonth":0,
         "timeChangeStartHour":0,
         "timeChangeStartMin":0,
         "timeChangeStopHour":0,
         "timeChangeStopMin":0
      }
}
```

Response Body

```
{
"calendarId": "Calendar_117",
"name": "Dad's Dating Calendar",
"locked": false,
"lockOwner": null,
"weekStarts": 4,
"weekEnds": 5,
"dayStartsHour": 14,
"dayStartsMin": 30,
"dayEndsHour": 23,
"dayEndsMin": 30,
"timeZoneId": "EST",
   "exceptions":[
      {
         "name":"Christmas",
         "type":"HOLIDAY",
         "placement":"FIXED",
         "fixedDate":30949200000,
         "annualMonth":0,
         "annualDay":0,
         "relativeType":null,
         "relativeDayOfWeek":0,
         "relativeMonth":0,
         "timeChangeStartHour":0,
         "timeChangeStartMin":0,
         "timeChangeStopHour":0,
         "timeChangeStopMin":0
      },
      {
         "name":"It's New Year's Day",
         "type":"HOLIDAY",
         "placement":"ANNUAL",
         "fixedDate":null,
         "annualMonth":1,
         "annualDay":1,
         "relativeType":null,
         "relativeDayOfWeek":0,
         "relativeMonth":0,
         "timeChangeStartHour":0,
         "timeChangeStartMin":0,
         "timeChangeStopHour":0,
         "timeChangeStopMin":0
      }
}
```

HTTP Status Codes

- 201 Created—Response body provided
- 401 Unauthorized—APIToken not valid
- 403 Forbidden—User does not have permission for the specified tenant, or does not have proper permission to perform this operation.
- 412 Precondition Failed—Calendar was not locked first

- 422 Unprocessable Entity—Field Validation Error

[+]Show

```
{
  "message": "Validation Failed",
  "errors": [
    {
       "code" : "1000",
       "field" : "type",
       "message": "Invalid type. Must be HOLIDAY or TIME_CHANGE"
    },
    {
  "code" : "1001",
       "field" : "placement",
       "message": "Invalid placement. Must be ANNUAL, FIXED or RELATIVE"
    }
]
}
```

Repeated for each field:

- `1002`—`fixedDate`—`fixedDate` must be specified with type of `FIXED`
- `1003`—`annualMonth`—Invalid value. Must be between 1 and 12
- `1004`—`annualDay`—Invalid value. Must be between 1 and 31.
- `1005`—`relativeType`—Invalid value, must be `EVERY, FIRST, SECOND, THIRD, FOURTH`, or `LAST`
- `1006`—`relativeDayOfWeek`—Invalid value. Must be between 1 and 7
- `1007`—`relativeMonth`—Invalid value. Must be between 1 and 12
- `1008`—`timeChangeStartHour`—Invalid value. Must be between `0` and 23
- `1009`—`timeChangeStartMin`—Invalid value. Must be between `0` and 59
- `1010`—`timeChangeStopHour`—Invalid value. Must be between `0` and 23
- `1011`—`timeChangeStopMin`—Invalid value. Must be between `0` and 59

- 503—Service Unavailable

Notes

User must have `CALENDAR_MODIFY` permission. User must have LOCK. User should pass in structure containing entire calendar body (exceptions). This is a complete replacement of current calendar body (not just changes). Updated structure is returned in response

Delete Calendar

Method

DELETE

Syntax

```
/grs/v1/tenant/{tenantId}/package/{packageId}/calendar/{calendarId}
```

Request Body

N/A

Response Body

N/A

HTTP Status Codes

- 204 No Content—returned if successful operation
- 401 Unauthorized—APIToken not valid
- 403 Forbidden—User does not have permission for the specified tenant or does not have proper permission to perform this operation.
- 404 Not Found—could not find specified package ID or calendarID.
- 412 Precondition Failed—Calendar was not locked first
- 503—Service Unavailable

Notes

Must have `CALENDAR_DELETE` permission. Must have lock on Calendar.

Deployment/Snapshots

- Query Snapshots
- Create Snapshot
- Delete Snapshot
- Query Deployment History
- Query Deployment Targets
- Deploy Rule Package or Snapshot
- Undeploy Rule Package or Snapshot

Query Snapshots

Method

GET

Syntax

```
/grs/v1/tenant/{tenantId}/package/{packageId}/snapshots
```

Request Body

N/A

Response Body

```
"snapshots":[
      {
        "snapshotId", "dcc4a39b-3442-4408-a907-b85c44663681",
  "name":"Snap 01",
         "comment":"Create prior to QA",
         "createDate":30949200000,
      },
      {
         "name":"Snap 02",
         "comment":"UAT",
         "createDate":31989200000,
      }
]
}
```

HTTP Status Codes

- 200 OK—Response body provided
- 401 Unauthorized—APIToken not valid
- 404 Not Found—could not find specified package ID.

Notes

User must have SNAPSHOT_VIEW permission.

Create Snapshot

Method

POST

Syntax

```
/grs/v1/tenant/{tenantId}/package/{packageId}/snapshot
```

Request Body

```
{
  "name":"Snap 01",
  "comment":"Create prior to QA",
}
```

Response Body

```
{
   "snapshotId", "dcc4a39b-3442-4408-a907-b85c44663681",
 "name":"Snap 01",
  "comment":"Create prior to QA",
  "createDate":30949200000,
}
```

HTTP Status Codes

- 201 - Created - Response body provided
- 401 - Unauthorized - APIToken not valid
- 403 - Forbidden - User does not have permission for the specified tenant, or does not have proper permission to perform this operation.
- 422 - Unprocessable Entity - Field Validation Error

```
{
  "message": "Validation Failed",
  "errors": [
    {
       "code" : "1000",
       "field" : "name",
       "message": "snapshot name is not valid"
}
]
}
```

- 422 - Unprocessable Entity - Rules in package do not validate

```
{
"message": "Package Validation Failed",
  "errors": [
    {
      "message": "Validation error 1"
    },
    {
      "message": "Validation error 2"
    }
  ]
}
```

Notes

User must have `SNAPSHOT_CREATE` permission.

- name—Snapshot name cannot contain special characters such as: { } [ ] . * | / : < > ' & due to internal restrictions.

Delete Snapshot

Method

DELETE

Syntax

```
/grs/v1/tenant/{tenantId}/package/{packageId}/snapshot/{snapshotId}
```

Request Body

N/A

Response Body

N/A

HTTP Status Codes

- 204 No Content—returned if successful operation
- 401 Unauthorized—APIToken not valid
- 403 Forbidden—User does not have permission for the specified tenant or does not have proper permission to perform this operation.
- 404 Not Found—could not find specified package ID.

Notes

User must have `SNAPSHOT_DELETE` permission.

Query Deployment History

Method

GET

Syntax

```
/grs/v1/tenant/{tenantId}/package/{packageId}/deploy
```

Request Body

N/A

Response Body

Up to 8.5.300

```
"deployments":[
{
    "deploymentId" : 34234,
    "packageVersion" : 1,
    "snapshotName":"Snap 01",
    "target", "GRECluster",
    "deployedBy", "username",
    "comments":"Deploy to QA servers",
    "deployScheduleDateTime": 34243432432,
    "status" : "SUCCESS"
}
]
```

From 8.5.301

```
[
   {
      "deploymentId": 34234,
      "packageVersion": 1,
      "snapshotName": "Snap 01",
      "target": "GRECluster",
      "deployedBy": "username",
      "deployedFrom": "GRAT_INSTANCE_1",
      "comments": "Deploy to QA servers",
      "deployScheduledDateTime": 1453228140000,
      "status": "SUCCESS"
   }
]
```

From 8.5.303

```
[
  {
  "deploymentId": 1493228481864,
  "packageVersion": 0,
  "snapshotName":  LATEST,
  "target": "GRE_850",
  "deployedBy": "demo",
  "deployedFrom": null,
  "comments": "Undeploy to QA servers",
  "deployScheduledDateTime": 1493228481851,
  "status": "PENDING",
  "operation": “UNDEPLOYED”
}
]
```

HTTP Status Codes

- 200 OK—Response body provided
- 401 Unauthorized—APIToken not valid
- 403 Forbidden—User does not have permission for the specified tenant.
- 404 Not Found—could not find specified package ID.

Notes

Status

- SUCCESS—deployment was successful
- FAILED—deployment failed
- PARTIAL—deployment was partially successful
- PENDING—deployment is still processing
- SCHEDULED—deployment is scheduled

Operation

- DEPLOYED—deployment was successful
- UNDEPLOYED—undeployment was successful

Query Deployment Targets

Method

GET

Syntax

```
/grs/v1/tenant/{tenantId}/deploytargets
```

Request Body

N/A

Response Body

```
"deploytargets":[
  {
"name" : "GRE_1",
"type" : SINGLE
},
{
"name" : "GRE_2",
"type" : SINGLE
},

{

"name" : "GRE_CLUSTER",

"type" : CLUSTER

}

]
}
```

HTTP Status Codes

- 200 OK—Response body provided
- 401 Unauthorized—APIToken not valid
- 403 Forbidden—User does not have permission for the specified tenant.

Notes

Type

- SINGLE—represents a single GRE node target
- CLUSTER—represents a cluster of GRE nodes

Deploy Rule Package or Snapshot

Method

POST

Syntax

```
/grs/v1/tenant/{tenantId}/package/{packageId}/deploy
```

Request Body

```
{
  "target": "GRE_Cluster",
  "snapshotName":"Snap 01",
  "comments":"Deploy to QA servers",
  "deployScheduledDateTime": 34243432432
}
```

Response Body

```
{
"target": "GRE_Cluster",  "deploymentId" : 34234,
  "snapshotName":"Snap 01",
  "comments":"Deploy to QA servers",
  "deployScheduleDateTime": 34243432432
}
```

HTTP Status Codes

- 201 Created—Response body provided with deployment ID
- 401 Unauthorized—APIToken not valid
- 403 Forbidden—User does not have permission for the specified tenant or does not have proper permission to perform this operation.
- 404 Not Found—could not find specified package ID or snapshot name

- 422 Unprocessable Entity—Field Validation Error

```
{
  "message": "Validation Failed",
  "errors": [
    {
       "code" : "1000",
       "field" : "name",
       "message": "snapshot name is not found"
}
    {
       "code" : "1001",
       "field" : "deployScheduleDateTime",
       "message": "scheduled date/time is in the past."
}
]
}
```

- 422 Unprocessable Entity—Rules in package do not validate

```
{
"message": "Package Validation Failed",
  "errors": [
    {
      "message": "Validation error 1"
    },
    {
      "message": "Validation error 2"
    }
  ]
}
```

Notes

User must have `PACKAGE_DEPLOY` and `SNAPSHOT_VIEW` permissions.

- `target`—Indicate which GRE or GRE Cluster you want to deploy to. Use "Query Deploy Targets" to get the list of valid targets.
- `snapShotName`—To deploy "latest" package, specify "LATEST" as snapshot name (or omit the field).
- `deployScheduledDateTime`—To schedule deployment for a future time, pass in `scheduledDateTime` field in UTC ms since EPOCH. To schedule immediately, omit this field.

Undeploy Rule Package or Snapshot

Method

POST

Syntax

```
/grs/v1/tenant/{tenantId}/package/{packageId}/undeploy
```

Request Body

```
{
  "target": "GRE_Cluster",
  "comments":"Undeploy to QA servers"
}
```

Response Body

```
{
  "target": "GRE_Cluster",
  "deploymentId" : 3423322234,
  "comments":"Undeploy to QA servers"
}
```

HTTP Status Codes

- 201 Created—Response body provided with deployment ID
- 401 Unauthorized—APIToken not valid
- 403 Forbidden—User does not have permission for the specified tenant or does not have proper permission to perform this operation.
- 404 Not Found—could not find specified package ID

Notes

User must have `PACKAGE_UNDEPLOY` and `SNAPSHOT_VIEW` permissions.

- `target`—Indicate which GRE or GRE Cluster you want to undeploy from. Use "Query Deploy Targets" to get the list of valid targets.