Genesys Rules System REST API Reference Guide

Query Linear Rule Body

12/13/2025

Contents

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.