



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 Pulse Deployment Guide

Pulse Restful Web Service API

12/20/2025

Pulse Restful Web Service API

Create external data for display in Pulse by using the Pulse Restful Web Service API.

Important

The API is subject to change at any time without notice.

Methods

- **/api/wbrt/layouts**

[+] GET

Returns an array of layouts for a tenant. If Tenant ID is omitted, returns the layouts for user's tenant and templates.

Example request URIs:

- /api/wbrt/layouts
- /api/wbrt/layouts?tenantId=1
- /api/wbrt/layouts?tenantId=0

Request query parameters

Name	Required	Type	Description
tenantId	not	integer	Specifies tenant to which layouts belong
uscn	not	integer	Specifies USCN, for filtering layouts changed after this USCN (non-inclusive)
type	not	string, available values: ItGENERIC ItPCREGULAR ItPCPERFORMANCE ItDATADEPOT ItOTHER	Specifies layout type

Required permission:

- **Pulse Read All Layouts** - for any request parameters
- **Pulse View Dashboard** - for all templates request (tenantId=0)

Available response representations:

- 200 - application/json with array of layouts
- 400 - Request is not valid, application/json with details:
 - { "message": "INVALID_URL" } - Specified wrong request parameters.
- 403 - User does not have **Pulse Read All Layouts** permission or user has **Pulse View Dashboard** permission, but the request is different from /api/wbrt/layouts?tenantId=0 (request for all templates)

[+] POST

Creates a new layout.

Available request representations:

- application/json with layout configuration

Required permission:

- **Pulse Manage Widgets** for creating layouts
- **Pulse Manage Templates and Default Dashboard** for creating templates

Available response representations:

- 200 - New layout was successfully created, redirect to newly created layout.
- 400 - Provided layout configuration is not valid, application/json with details:
 - { "message": "LAYOUT_PARSE_ERROR" }, when widget has an invalid format
- 403 - User does not have **Pulse Manage Widgets** permission and tries to create layout or user does not have **Pulse Manage Templates and Default Dashboard** permission and tries to create template

• /api/wbrt/layouts/<id>

[+] GET

Returns layout with specified ID.

Required permission:

- **Pulse Read All Layouts** or **Pulse View Dashboard**

Available response representations:

- 200 - application/json with requested layout
- 403 - User does not have **Pulse View Dashboard** or **Pulse Read All Layouts** permissions
- 404 - Layout does not exist or the layout is not accessible by the user

[+] PUT

Updates layout with specified ID. If there is more than one widget for this layout then a new layout is created.

Available request representations:

- application/json with layout configuration

Required permission:

- **Pulse Manage Widgets**, for updating layout

- **Pulse Manage Templates and Default Dashboard**, for updating template

Available response representations:

- 200 - Layout was updated, redirects to new newly created layout
- 400 - Provided layout configuration is not valid, application/json with details:
 - { "message": "LAYOUT_PARSE_ERROR" } - layout has invalid format
- 403 - User does not have **Pulse Manage Widgets** permission and tries to update layout or user does not have **Pulse Manage Templates and Default Dashboard** permission and tries to update template
- 404 - Layout not exists, application/json with details:
 - { "message": "LAYOUT_NOT_FOUND" } - layout does not exist

[+] DELETE

Delete layout with specified ID.

Required permission:

- **Pulse Manage Widgets**, for deleting layout
- **Pulse Manage Templates and Default Dashboard**, for deleting template

Available response representations:

- 200 - Layout was deleted
- 204 - Layout was already deleted
- 403 - User does not have **Pulse Manage Widgets** permission and tries to delete layout or user does not have **Pulse Manage Templates and Default Dashboard** permission and tries delete template

- **/api/wbrt/layouts/<id>/snapshot**

[+] GET

Returns recent snapshot for specified layout.

Note: This method performs additional row filtering based on user access restrictions for snapshots with layout_type = **ITPCREGULAR** and layout_type = **ITPCPERFORMANCE**. Rows with objects not accessible for user are filtered out. It must work as follows: if column _Object\$CfgType exists, then use pair (_Object\$CfgType, _Object\$ID) to check permissions; otherwise attempt the usual combination (_Object\$Type, _Object\$ID).

Required permission:

- **Pulse View Dashboard**

Available response representations:

- 200 - application/json snapshot
- 204 - no content, snapshot does not exist
- 403 - User does not have **Pulse View Dashboard**
- 404 - Related resource is not found, application/json with details:
 - { "message": "LAYOUT_NOT_FOUND" } - layout does not exist
 - { "message": "STATUS_NOT_ACTIVE_YET" } - layout is not active

[+] POST

Saves a snapshot for layout with specified ID. Uses timestamp property from Layout Snapshot to distinguish different snapshots. Any saved snapshot with the same timestamp is overwritten.

Available request representations:

- application/json with snapshot

Required permission:

- **Pulse Write Snapshot**

Available response representations:

- 200 - snapshot was successfully saved
- 400 - snapshot is not valid, application/json with details:
 - { "message": "SNAPSHOT_PARSE_ERROR" } - snapshot has invalid format
 - { "message": "INVALID_URL" } - URL layout ID does not match layout ID in the snapshot
- 403 - user does not have **Pulse Write Snapshot** permission
- 404 - related resource not found, application/json with details:
 - { "message": "LAYOUT_NOT_FOUND" } - layout does not exist

- **/api/wbrt/layouts/<id>/snapshots**

[+] GET

Returns array of snapshots for specified layout and matched filter period. If the start and end are not specified then returns only latest snapshot.

Example request URI(s):

- /api/wbrt/layouts/1/snapshots
- /api/wbrt/layouts/1/snapshots?start=1411720605
- /api/wbrt/layouts/1/snapshots?start=1411720605&columns=Inbound_Talk_Time,Internal_Talk_Time
- /api/wbrt/layouts/1/snapshots?start=1400000000&frequency=10

Request query parameters

Name	Required	Type	Description
start	not	long	Unix epoch timestamp indicating the start of the period for which snapshots should be returned. If not specified then all snapshots generated before end time are returned.
end	not	long	Unix epoch timestamp indicating the end of the period for which snapshots should be returned. If not specified then all snapshots generated after start time are returned.
frequency	not	integer	Minimum interval, in seconds,

Name	Required	Type	Description
			between two snapshots in history. Needed for reducing the number of snapshots returned by this request. Default value is 0, so all existing snapshots inside the specified period are returned.
columns	not	Array of Strings	Array of Column.id that is to be included to snapshot. If not specified then all columns are included.

Note: This method performs additional row filtering based on user access restrictions for snapshots with layout_type = **ITPCREGULAR** and layout_type = **ITPCPERFORMANCE**. Rows with objects not accessible for user are filtered out. It must work as follows: if the column_Object\$CfgType exists then use pair (_Object\$CfgType, _Object\$ID) to check permissions; otherwise attempt the usual combination (_Object\$Type, _Object\$ID).

Required permission:

- Pulse Write Snapshot

Available response representations:

- 200 - application/json with array of snapshots
- 403 - In cases when user does not have **Pulse View Dashboard**

• /api/wbrt/widgets

[+] POST

Creates a new widget. If the parameter default is not specified then a user creates a widget that cannot be seen by other users. If the parameter default is specified and not equal to false then the default widget is created and can be seen by anyone.

Example request URI(s):

- /api/wbrt/widgets?default=true
- /api/wbrt/widgets?default=false
- /api/wbrt/widgets

Request query parameters

Name	Required	Type	Description
default	not	string	If specified and different from false then a default widget is created.

Note: owner_user_name field is ignored; widget is created for current user or as default widget.

Available request representations:

- application/json with widget configuration

Required permission:

- **Pulse Manage Widgets** - for creating user widget

- **Pulse Manage Templates and Default Dashboard** - for creating default widget

Available response representations:

- 200 - New widget was successfully created, redirect to newly created widget.
- 400 - Provided widget configuration is not valid, application/json with details:
 - { "message": "WIDGET_PARSE_ERROR" } - widget has invalid format
- 403 - In cases when user does not have **Pulse Manage Widgets** and tries to create user widget or **Pulse Manage Templates and Default Dashboard** and tries to create default widget
- 404 - Widget configuration contains nonexistent data, application/json with details:
 - { "message": "LAYOUT_NOT_FOUND" } - widget layout does not exist

- **/api/wbrt/widgets/<id>**

[+] GET

Returns widget configuration for widget with specified ID and belonging to user or default widget.

Required permission:

- **Pulse View Dashboard**

Available response representations:

- 200 - application/json with widget configuration
- 403 - In cases when user does not have **Pulse View Dashboard**
- 404 - Widget with requested ID is not found, application/json with details:
 - { "message": "WIDGET_NOT_FOUND" }

[+] PUT

Update widget configuration for specified widget.

Example request URI(s):

- /api/wbrt/widgets/11?default=true
- /api/wbrt/widgets/12?default=false
- /api/wbrt/widgets/10

Request query parameters

Name	Required	Type	Description
default	not	string	If specified and not "false" then default widget are updated.

Note: owner_user_name field is ignored; widget is updated for current user or default widget.

Available request representations:

- application/json with widget configuration

Required permission:

- **Pulse Manage Widgets** - for updating user widget
-

- **Pulse Manage Templates and Default Dashboard** - for updating default widget

Available response representations:

- 200 - Widget was successfully updated
- 400 - Provided widget configuration is not valid, application/json with details:
 - { "message": "WIDGET_PARSE_ERROR" } - widget has invalid format
 - { "message": "INVALID_URL" } - widget ID or default parameter in URL does not correspond to ID in the body or widget's owner.
 - { "message": "INVALID_LAYOUT_HASH" } - specified layout has different body_hash than provided in snapshot
- 403 - In cases when user does not have **Pulse Manage Widgets** and tries to update user widget or **Pulse Manage Templates and Default Dashboard** and tries to update default widget
- 404 - Widget configuration contains nonexistent data or widget itself does not exist, application/json with details:
 - { "message": "WIDGET_NOT_FOUND" } - specified widget does not exist
 - { "message": "LAYOUT_NOT_FOUND" } - widget layout does not exist

[+] DELETE

Delete widget with specified ID.

Request query parameters

Name	Required	Type	Description
default	not	string	If specified and not "false" then default widget are deleted.

Required permission:

- **Pulse Manage Widgets** - for deleting user widget
- **Pulse Manage Templates and Default Dashboard** - for deleting default widget

Available response representations:

- 200 - Widget was deleted
- 204 - Widget was already deleted
- 403 - In cases when user does not have **Pulse Manage Widgets** and tries to delete user widget or **Pulse Manage Templates and Default Dashboard** and tries to delete default widget

- **/api/wbrt/dashboards**

[+] GET

Returns list of user's dashboard.

Required permission:

- **Pulse View Dashboard**

Available response representations:

- 200 - application/json with list of user's dashboards

Response body example:

```
[
  {
    "name": "tab1413974752841",
    "idx": 0,
    "title": "My Dashboard"
  },
  {
    "name": "tab1414522307471",
    "idx": 1,
    "title": "New Dashboard"
  }
]
```

- 403 - In cases when user does not have **Pulse View Dashboard**
- 503 - application/json with malfunction details:
 - { "message": "CFG_SERVER_ERROR" } - no connection to conf server

- **/api/wbrt/dashboards/<id>/widgets**

[+] GET

Returns list of widgets placed on the dashboard with specified ID.

ID is a dashboard **Name** or **default** if you want to get widgets from the default dashboard.

Example request URI(s):

- /api/wbrt/dashboards/tab123456789/widgets
- /api/wbrt/dashboards/default/widgets

Required permission:

- **Pulse View Dashboard**

Available response representations:

- 200 - application/json with list of widgets
- 403 - In cases when user does not have **Pulse View Dashboard**
- 404 - application/json with details:
 - { "message": "DASHBOARD_NOT_FOUND" } - specified dashboard does not exist
- 503 - application/json with malfunction details:
 - { "message": "CFG_SERVER_ERROR" } - no connection to conf server

[+] POST

Put widget on the dashboard with specified ID.

ID is a dashboard **Name** or **default** if you want to put widget on the default dashboard.

Example request URI(s):

- /api/wbrt/dashboards/tab123456789/widgets
- /api/wbrt/dashboards/default/widgets

Required permission:

- **Pulse Manage Widgets** - for putting widget to user dashboard

- **Pulse Manage Templates and Default Dashboard** - for putting widget to the default dashboard

Available request representations:

- application/json with dashboard widget configuration

Request Body Example

```
[{
  "widgetId": 683, //widget id to be placed
  "col": 1, //optional, desired column
  "row": 2 //optional, desired row
}]
```

Available response representations:

- 201 - Widget successfully placed to the dashboard
- 403 - In cases when user does not have **Pulse Manage Widgets** and tries to put widget on the user dashboard or **Pulse Manage Templates and Default Dashboard** and tries to put widget on the default dashboard
- 404 - application/json with details:
 - { "message": "WIDGET_NOT_FOUND" } - specified widget does not exist
 - { "message": "DASHBOARD_NOT_FOUND" } - specified dashboard does not exist
- 503 - application/json with malfunction details:
 - { "message": "CFG_SERVER_ERROR" } - no connection to configuration server

• /api/plugins/wbrt/health

[+] GET

Checks Pulse application health.

Required permission:

- no permissions required

Available response representations:

- 200 - Pulse application works fine
- 503 - Pulse health check failed, application/json with malfunction details:
 - { "message": "HEALTH_CHECK_FAILED" } - no details are available
 - { "message": "COLLECTOR_DOES_NOT_RESPOND" } - collector stopped updating heartbeat file
 - { "message": "DATABASE_ERROR" } - no connection to database
 - { "message": "CFG_SERVER_ERROR" } - no connection to conf server

Examples: Generic Layout Definition

Generic Layout Definition Example

```
[{
  "definition": {
    "tenant_dbid": 1,
    "refresh_interval": 15,
    "layout_type": "ltGENERIC",
    "column": [
      {
        "category": "ccDIMENSION",
        "is_delta_key": true,
        "id": "_Object$ID"
      },
      {
        "category": "ccDIMENSION",
        "is_delta_key": true,
        "id": "_Object$Type"
      },
      {
        "category": "ccDIMENSION",
        "id": "_Object$Name",
        "format": "string"
      },
      {
        "category": "ccMEASURE",
        "vt": "vINT",
        "id": "First_column",
        "type": "ctGENERIC",
        "format": "integer",
        "label": "First"
      },
      {
        "category": "ccMEASURE",
        "vt": "vINT",
        "id": "Second_column",
        "type": "ctGENERIC",
        "format": "time",
        "label": "Second"
      },
      {
        "category": "ccMEASURE",
        "vt": "vSTR",
        "id": "Third_column",
        "type": "ctGENERIC",
        "format": "string",
        "label": "Third"
      }
    ],
    "default_widget": {
      "threshold": [
        {
          "value": [
            0,
            0,
            0
          ],
          "direction_up": false,
          "column_id": "First_column"
        }
      ],
      "view": [
        {
```

```
        "column_selector": [
            "Second_column"
        ],
        "sorting": [
            {
                "is_asc": true
            }
        ],
        "type": "BarView"
    }
],
"id": -1,
"size": "1x4",
"label": "Test  Widget"
},
"name": "Third party source",
"description": "Example of third party source"
},
"id": 3,
"state": {
    "requested_status": "stACTIVE",
    "body_hash_1": 583854940,
    "body_hash_2": 583854940
}
}
```

Layout Snapshot

Layout Snapshot Example

```
[{
  "layout_id": 3,
  "col": [
    {
      "v": [
        101,
        102,
        103,
        104,
        105,
        106,
        107,
        108,
        109,
        110
      ],
      "col": {
        "category": "ccDIMENSION",
        "is_delta_key": true,
        "vt": "vINT",
        "id": "_Object$ID"
      }
    },
    {
      "v": [
        "Agent",

```

```
        "Agent",
        "Agent",
        "Agent",
        "Agent",
        "Agent",
        "Agent",
        "Agent",
        "Agent",
        "Agent"
    ],
    "col": {
        "category": "ccDIMENSION",
        "vt": "vSTR",
        "id": "_Object$Type"
    }
},
{
    "v": [
        "Agent1",
        "Agent2",
        "Agent3",
        "Agent4",
        "Agent5",
        "Agent6",
        "Agent7",
        "Agent8",
        "Agent9",
        "Agent10"
    ],
    "col": {
        "category": "ccDIMENSION",
        "vt": "vSTR",
        "id": "_Object$Name"
    }
},
{
    "v": [
        60,
        120,
        180,
        234,
        123,
        531,
        521,
        231,
        541,
        634
    ],
    "col": {
        "category": "ccMEASURE",
        "vt": "vINT",
        "id": "First_column",
        "type": "ctGENERIC",
        "format": "integer",
        "label": "First"
    }
},
{
    "v": [
        523,
        0,
        312,
        543,
```

```
        631,
        434,
        423,
        642,
        743,
        135
    ],
    "col": {
        "category": "ccMEASURE",
        "vt": "vINT",
        "id": "Second_column",
        "type": "ctGENERIC",
        "format": "time",
        "label": "Second"
    }
},
{
    "v": [
        "Value1",
        "Value2",
        "Value3",
        "Value4",
        "Value5",
        "Value6",
        "Value7",
        "Value8",
        "Value9",
        "Value10"
    ],
    "col": {
        "category": "ccMEASURE",
        "vt": "vSTR",
        "id": "Third_column",
        "type": "ctGENERIC",
        "format": "string",
        "label": "Third"
    }
}
],
"state": {
    "current_status": "stACTIVE",
    "body_hash_1": 583854940,
    "body_hash_2": 583854940
},
"layout_type": "ltGENERIC",
"timestamp": 1408098466350
}]
```

Widget

Widget Example

```
[{
    "id": 2,
    "layout_id": 3,
    "label": "MyWidget",
```

```
"size": "1x4",
"owner_user_name": "default",
"view": [{
  "type": "BarView",
  "column_selector": ["Login_Duration"],
  "sorting": [{
    "is_asc": false
  }]
}]
}]
```