

Widgets

Widgets are designed to be used on Dashboards.

Enterprise users have complete freedom to create, update and delete widgets as required.

Creating

When creating a widget, if you select the 'Advanced' button you will have the ability to upload your own custom SQL.

The *SELECT* section of your SQL must contain fully qualified columns and specify certain columns.

Pie charts must return name or my_name, description or my_description and count.

Line charts must return date or my_date, description or my_description and count.

The WHERE section of your SQL must contain **WHERE @filter** so Open-Audit knows to restrict your query to the appropriate Orgs.

SQL not containing this condition will result in the widget failing to be executed.

The SQL cannot contain **WHERE @filter OR**. That SQL will not be executed, however **WHERE @filter AND** queries are allowed.

An example widget SQL showing devices counted per location.

```
SELECT locations.name as `name`,
       locations.id AS `description`,
       count(system.id) AS `count`
FROM locations LEFT JOIN system ON (locations.id = system.location_id)
WHERE @filter
GROUP BY locations.name
```

More elaborate SQL can be used to group attributes within a range. Below shows the SQL for devices not seen in a pie chart grouped by last seen date ranges.

```
SELECT IF ( system.last_seen = "2000-01-01", "unknown", ( IF ( system.last_seen < DATE(NOW() - INTERVAL 180
day), "180 Days or more",
( IF ( system.last_seen < DATE(NOW() - INTERVAL 150 day), "150-180 days", ( IF ( system.last_seen < DATE(NOW()
- INTERVAL 120 day), "120-150 days",
( IF ( system.last_seen < DATE(NOW() - INTERVAL 90 day), "90-120 days", ( IF ( system.last_seen < DATE(NOW() -
INTERVAL 60 day), "60-90 days",
( IF ( system.last_seen < DATE(NOW() - INTERVAL 30 day), "30-60 days", "7-30 days" ) ) ) ) ) ) ) ) ) ) ) ) AS
`my_name`,
IF ( system.last_seen = "2000-01-01", "system.last_seen=",
( IF ( system.last_seen < DATE(NOW() - INTERVAL 180 day), CONCAT("system.last_seen=<", DATE(NOW() - INTERVAL
180 day)),
( IF ( system.last_seen < DATE(NOW() - INTERVAL 150 day), CONCAT("system.last_seen=>", DATE(NOW() - INTERVAL
180 day), "&system.last_seen=<", DATE(NOW() - INTERVAL 150 day)),
( IF ( system.last_seen < DATE(NOW() - INTERVAL 120 day), CONCAT("system.last_seen=>", DATE(NOW() - INTERVAL
150 day), "&system.last_seen=<", DATE(NOW() - INTERVAL 120 day)),
( IF ( system.last_seen < DATE(NOW() - INTERVAL 90 day), CONCAT("system.last_seen=>", DATE(NOW() - INTERVAL 120
day), "&system.last_seen=<", DATE(NOW() - INTERVAL 90 day)),
( IF ( system.last_seen < DATE(NOW() - INTERVAL 60 day), CONCAT("system.last_seen=>", DATE(NOW() - INTERVAL 90
day), "&system.last_seen=<", DATE(NOW() - INTERVAL 60 day)),
( IF ( system.last_seen < DATE(NOW() - INTERVAL 30 day), CONCAT("system.last_seen=>", DATE(NOW() - INTERVAL 60
day), "&system.last_seen=<", DATE(NOW() - INTERVAL 30 day)),
CONCAT("system.last_seen=>", DATE(NOW() - INTERVAL 30 day), "&system.last_seen=<", DATE(NOW() - INTERVAL 7
day))) ) ) ) ) ) ) ) ) ) ) AS `my_description`,
count(system.id) AS `count`
FROM system
WHERE @filter AND DATE(system.last_seen) < DATE(NOW() - INTERVAL 7 day)
GROUP BY `my_name` ORDER BY system.last_seen;
```

Database Schema

The database schema can be found in the application if the user has database::read permission by going to menu: Admin -> Database -> List Tables, then clicking on the details button for the table.

API / Web Access

You can access the collection using the normal Open-Audit JSON based API. Just like any other collection. Please see [The Open-Audit API](#) documentation for further details.

Default Items

Shipped are a set of default items. These can be found by going to menu: Help Defaults Widgets.