

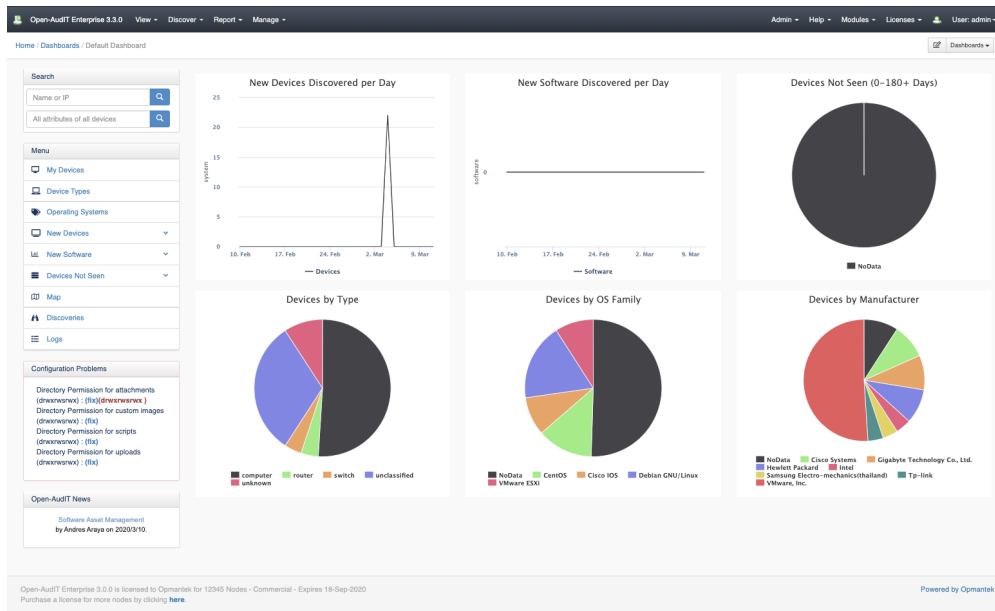
Dashboards

As at Open-Audit 2.2, we now ship with several pre-configured Dashboards. If you are a Professional user, you can select between two of these and set it as your default page when you logon. If you are an Enterprise user, you can not only select between the five preconfigured Dashboards, but you also have the ability to create your own.

The page [How do Dashboards and Widgets work?](#) may be of some assistance here.

Dashboards use [Widgets](#), of which we supply over thirty as a default set. Professional users cannot alter these, but an Enterprise user has complete customisation abilities - including creating your own.

Dashboard provide "at a glance" important visibility for changes occurring on your network as well as providing quick links to device types, manufacturers, and operating systems and so much more.



The screenshot shows the Open-Audit Enterprise 3.3.0 dashboard with a list of available dashboards. The top navigation bar includes 'Admin', 'Help', 'Modules', 'Licenses', and 'User: admin'. The main content area is titled 'Home / Dashboards'. Below the title, there is a search bar and a table of dashboards. The table has columns for 'Execute', 'Details', 'Name', 'Description', and 'Delete'. The table lists six dashboards: 'Clouds Dashboard', 'Default Dashboard', 'Devices Dashboard', 'Discovery Dashboard', 'Network Dashboard', and 'Purchasing Dashboard'. At the bottom, there is a pagination control showing 'Showing 1 to 6 of 6 entries' and buttons for 'First', 'Previous', 'Next', and 'Last'.

Execute	Details	Name	Description	Delete
		Clouds Dashboard	The details of your cloud infrastructure	
		Default Dashboard	The Default Open-Audit Dashboard	
		Devices Dashboard	Devices Information	
		Discovery Dashboard	Discovered Devices	
		Network Dashboard	Networks and Locations	
		Purchasing Dashboard	Purchase Management	

Database Schema

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

```

CREATE TABLE `dashboards` (
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `name` varchar(100) NOT NULL DEFAULT '',
  `org_id` int(10) unsigned NOT NULL DEFAULT '1',
  `type` enum('default','org','user','') NOT NULL DEFAULT '',
  `user_id` int(10) unsigned NOT NULL DEFAULT '0',
  `description` text NOT NULL,
  `sidebar` enum('y','n') NOT NULL DEFAULT 'y',
  `options` text NOT NULL,
  `edited_by` varchar(200) NOT NULL DEFAULT '',
  `edited_date` datetime NOT NULL DEFAULT '2000-01-01 00:00:00',
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

API / Web Access

You can access the /dashboards collection using the normal Open-Audit JSON based API. Just like any other collection (assuming you're an Enterprise user). Please see the API documentation for further details.

Access is provided as part of a roles permissions. Dashboards is a standard resource and can have create, read, update, delete and execute permissions.

The API routes below are usable from both a JSON Restful API and the web interface. The Web application routes are specifically designed to be called from the web interface (a browser).

API Routes

Request Method	ID	Action	Resulting Function	Permission Required	URL Example	Notes	Example Response
GET	n		collection	dashboards::read	/dashboards	Returns a list of dashboards.	
POST	n		create	dashboards::create	/dashboards	Insert a new dashboards entry.	
GET	y		read	dashboards::read	/dashboards/{id}	Returns a dashboards details.	
PATCH	y		update	dashboards::update	/dashboards/{id}	Update the details of a dashboards entry.	
DELETE	y		delete	dashboards::delete	/dashboards/{id}	Delete a dashboards entry.	
EXECUTE	y	execute	execute	dashboards::read	/dashboards/{id}/execute	Execute a dashboards and return the resulting dataset.	

Web Application Routes

Request Method	ID	Action	Resulting Function	Permission Required	URL Example	Notes
GET	n	create	create_form	dashboards::create	/dashboards/create	Displays a standard web form for submission to POST /dashboards.
GET	y	update	update_form	dashboards::update	/dashboards/{id}/update	Show the dashboards details with the option to update details using PATCH to /dashboards/{id}