Locations (pre v2)

- Introduction
- How Does it Work?
- Creating a Location
- View Location Details
- Database Schema
- Example Database Entry
- API / Web Access
    - API Routes
    - Web Application Routes

Introduction

A location is a physical address that can have devices associated with it. You can assign it coordinates and if there are devices assigned, the location will appear on the Map in Open-AudIT Enterprise.

How Does it Work?

Locations are used to assign devices to and will appear on the map in Open-AudIT Enterprise if they have any devices.

Creating a Location

To make another location go to menu -> Admin -> Locations -> Create Location.

View Location Details

Go to menu -> Admin -> Locations -> List Locations.

You will see a list of available locations. You can view a network by clicking on the it's ID (in green). You can also edit or delete the script (if delete is permitted). You cannot delete the Default Location.

Database Schema

```
CREATE TABLE `locations` (
  `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('Airforce','Airport','Ambulance Station','Army','Cloud','Conference Center','Court House','Data
Center','Depot','District','Doctors Office','Embassy','Factory','Fire Station','Guard','Head Office','High
School','Hospital','Hotel','House','Library','Light House','Marina','Mobile Phone
Tower','Motel','Navy','Newsagent','Nursing Home','Observatory','Office','Oil Rig','Police','Power Plant','Power
Substation','Prison','Radio Station','Regional','Resort','Retail','School','Security','Stadium','Train
Station','Travel Agency','TV Station','University','Warehouse','Workshop') NOT NULL DEFAULT 'Office',
  `room` varchar(100) NOT NULL DEFAULT '',
  `suite` varchar(100) NOT NULL DEFAULT '',
  `level` varchar(100) NOT NULL DEFAULT '',
  `address` varchar(100) NOT NULL DEFAULT '',
  `suburb` varchar(100) NOT NULL DEFAULT '',
  `city` varchar(100) NOT NULL DEFAULT '',
  `district` varchar(100) NOT NULL DEFAULT '',
  `region` varchar(100) NOT NULL DEFAULT '',
  `area` varchar(100) NOT NULL DEFAULT '',
  `state` varchar(100) NOT NULL DEFAULT '',
  `postcode` varchar(10) NOT NULL DEFAULT '',
  `country` varchar(100) NOT NULL DEFAULT '',
  `tags` varchar(250) NOT NULL DEFAULT '',
  `phone` varchar(20) NOT NULL DEFAULT '',
  `picture` varchar(100) NOT NULL DEFAULT '',
  `latitude` float(10,6) NOT NULL,
  `longitude` float(10,6) NOT NULL,
  `geo` varchar(200) NOT NULL DEFAULT '',
  `comments` varchar(100) NOT NULL DEFAULT '',
  `icon` varchar(100) NOT NULL DEFAULT '',
  `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;
```

Example Database Entry

Locations are stored in the database in the "locations" table. A typical entry will look as below.

```
       id: 4
     name: The Cunninghams
     type: House
     room:
    suite:
    level:
  address: 565 North Clinton Drive
   suburb:
     city: Milwaukee
 district:
   region:
     area:
    state: Wisconsin
 postcode: 4000
  country: America
     tags:
    phone:
  picture:
 latitude: 42.556568
longitude: -88.874382
      geo:
 comments:
     icon:
 group_id: 0
```

API / Web Access

You can access the /locations collection using the normal Open-AudIT JSON based API. Just like any other collection. Please see the API documentation for further details.

API Routes

Request Method	ID	Action	Resulting Function	Required Permission	URL Example	Notes	Example Response
GET	n		collection	locations::read	/locations	Returns a list of locations.	locations_collection.json
GET	y		read	locations::read	/locations/{id}	Returns a location's details.	locations_read.json
PATCH	y		update	locations::update	/locations/{id}	Update an attribute of a locations entry.	locations_patch.json
POST	n		create	locations::create	/locations	Insert a new locations entry.	locations_create.json
DELETE	y		delete	locations::delete	/locations/{id}	Delete a locations entry.	locations_delete.json

Web Application Routes

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