# Facility Database Reference

This document will cover each table and field in the Facility Database.

Each table will begin on a new page.

# Table: rawData

## Overview

This table holds all the log data that is sent to the Facility Database. Each event gets one record. In the description of the fields we will reference the expected source of the data. Sometimes a PHP page may want to create a log record in addition to the logging passed in to it. For example, `rfidcheckinout.php` logs the original event and then also logs either a checkin or checkout .

## Used By

1. `rfidlog.php` - The main way data is added to this table. A JSON is passed in with data corresponding to each field below.
2. `rfidcheckinout.php` - Check In events come to this URL and it determines if a checkin or checkout is happening.
3. Every report .PHP file selects data from this table.

## Fields

### recNum

This field is unique and set automatically by MySQL when a row is inserted. Do not change this.

### dateCreated

Supplied automatically by MySQL when the row is inserted. The timezone on this date will correspond to the location of the cloud server. Useful for knowing the exact time between inserts. Used in debugging.

### datePublishedAt

If the data posted to one of the URLs comes from a Particle.io webhook, then this will be the time provided by Particle.io, of the timezone where the Particle cloud servers live. Useful in debugging to know the sequence of events from the Particle cloud perspective.

### dateEventLocal

If the JSON posted came from an RFID Station, then this will be the local time of that Argon microprocessor. This is the most useful reporting time as it corresponds to the local time of the facility where the RFID Station is installed.

## coreID

If the JSON posted came from an RFID Station, then this will be the Particle.io assigned unique identifier for that processor.

## deviceFunction

When an RFID Station publishes an event to the cloud, it will include the text of the deviceName it received from stationConfig on boot up.

## clientID

When an RFID Station publishes an event to the cloud, it will include the applicable client ID from the CRM system. Some events, such as a reboot log event, will set the client ID to 0.

## firstName

When the event includes a client ID, the first name of the client is included. The first name comes from the CRM system.

## eventName

When an RFID Station publishes an event, it is really calling a Particle.io webhook. This is the name of that webhook, passed to us by the Particle cloud.

## logEvent

When an RFID Station publishes an “allow” event to the cloud, it will include the text of the logEvent it received from stationConfig on boot up.

## logData

On any log event there is the ability to send free form data for whatever reason. This is not expected to necessarily be machine readable, but useful to humans looking at the data.

## ipAddress

When an event is logged the PHP code will capture the upstream IP address and it gets put in here.

# Table: stationConfig

## Overview

Each type of RFID Station has one active entry in the stationConfig table.

## Used By

1. fdbGetStationConfig.php
    a. RFID Stations: When a Station boots up it determines its type from local non-volatile storage. It then obtains configuration information from one row of this table, based on its deviceType.
    b. Android Admin app: The app obtains the list of valid Station Types from this table.

## Fields

### recNum

An automatically created unique number and should never be changed. If you understand databases, then you understand why we have this. If not, you should be careful about what you do in this database.

### Active

If <> 0 this is the active record for the deviceType. There must be only one active record for each deviceType. This field allows the administrator to change a device type by adding a new record and marking the old record with Active = 0, thus preserving the previous configuration in case you need to quickly revert.

### deviceType

This is the number used by a Station when asking for its configuration parameters. The system will return the active configuration for this deviceType.

### deviceName

When Stations of this type make log entries in the Facility Database they will use this name to identify its type. Typical values are: “checkin”, “admin”, “woodshop”, “3D printer”, etc.

## LCDName

When idle, Stations of this type will display this value on their LCD displays. The value should be less than 16 characters. Typical values are: “Woodshop”, “Laser Cutter”, “Bandsaw”, etc.

## photoDisplay

When a user taps on a Station of this type, the Station will log this value in the Facility Database. This value will eventually be displayed under the member’s name on the Checked In photo display web page. To keep the web page clean, this value should be very short. Typical values are: “Wood”, “3D”, “Laser”, etc.

## OKKeywords

This field provides the tie between your RFID System and your CRM member certification system. You must set up your member certification choices in a way that allows you to define keyword groups that will work in your RFID system to achieve your objectives.

A comma delimited list of keywords. Stations of this type retrieve member certification information from the CRM system (in EZFacility these are called “packages”) and scan that certification information for these keywords as substrings. If any of the keywords is found then the member is assumed to have permission to access the area or equipment associated with this type of RFID Station. For instance, a “woodshop” Station might look for the keyword “wood” in the certification information.

Typical values are: “wood”, “3D”, “consew”, etc.

## logEvent

Devices of this type will use this as the Particle Event Name when they log information to the Facility Database. It may be used in reporting. Typical values are: “wood allowed”, “textile allowed”, etc.

This field may seem redundant at this time, but we include it to allow for possible future expansion of the Station operations.

## dateCreated

Defaults to CURRENT_TIMESTAMP and should not be overwritten.

# Table: clientInfo

## Overview

This table holds data about each clientID, one record per clientID. As of this date the data is used to provide different decoration on the `rfidcurrentcheckins.php` web page. For example, a clientID might have a displayClasses value of “staff”, and that could cause their photo to show up differently in the `rfidcurrentcheckins.php` web page.

Records are inserted or updated in this table by `rfidcheckinout.php`

Frankly, this table could become the basis of a home built CRM system.

## Used By

1. `rfidcheckinout.php` - When this web page is executed the last thing it does is call `sp_insert_update_clientInfo.sql` which determines whether to insert or update a record in this table.
2. `rfidcurrentcheckins.php` - This page displays a grid of the clients that are currently checked in to the facility. It gets decoration information from this table.

## Fields

### recNum

This field is unique and set automatically by MySQL when a row is inserted. Do not change this.

### dateCreated

Supplied automatically by MySQL when the row is inserted. The timezone on this date will correspond to the location of the cloud server. Useful for knowing the exact time between inserts. Used in debugging.

### clientID

The client ID from the CRM system. A unique field in this table.

### firstName

The first name of the client. `sp_insert_update_clientInfo.sql` will update this field.

## lastName

The last name of the client. `sp_insert_update_clientInfo.sql` will update this field.

## dateLastSeen

`sp_insert_update_clientInfo.sql` will update this field. This stored procedure is called by `rfidcheckinout.php` so each checkin/checkout event will update this date.

## isCheckeIn

Not used at this time.

## displayClasses

A string that will be added to the HTML attribute `classes=` for each picture `div` generated by `rfidcurrentcheckins.php`. For example, if this field contains “staff wood” then the photo div for this client will contain at least `classes="staff wood"` This allows CSS style decoration of this client’s particular photo display.