Facility Database Admin Help

The Facility Database holds the event logs from each of the RFID stations. It also holds the configuration information for the stations. And, it is the data source for the reporting web site.

In this document we assume you are familiar with PHP, MySQL, phpMyAdmin (or similar tool), and curl.

Database Setup

Create the database

Using your ISP admin console, or similar, create a database called rfidlogs (or your own name). You do not need to make the database available for remote access.

Create users

Create two users:

- rfidreader (Execute SP, Select)
- rfidlogger (Insert, Select)

Database structure

Use the file rfidlogs_schema.sql to create the tables and stored procedures.

Website Setup

Deploy

Copy all the files in datalogging to the root of your website.

Modify the rfidconfig.ini file

- Add the two database uses in the appropriate keys, and their passwords.
- Change the photoDisplay key to have the URL prefix of the server hosting your member's photos.
- Change the file permission to 660

Test

Bring up the web page rfidhome.html

The Database

The rfidlogs database has two tables: rawdata and stationConfig.

rawdata

This table holds the data for every log event that is recorded. These are typically made by an rfid Station, but could be made by any process that calls the correct URL.

rawdata is written to by rfidcheckin.php and rfidlog.php. These pages accept the POST of a JSON package. Each file has a description of the JSON it expects in the header of the file.

rfidcheckin.php

is used to toggle the state of a clientID in the database. Each day, the first time rfidcheckin is called for a particular clientID, a record is entered for Checked In.

The next time that clientID is used a Checked Out record is entered.

rfidlog.php

is a more generic log function. The parameters passed to it are added to the rawdata table. This is useful for logging errors, warnings, restart events, etc.

stationConfig

This table contains configuration information for the different stations you have. In a nutshell, each station has a deviceType. When the station boots it calls fdbGetStationConfig.php passing in its deviceType. The URL returns a JSON package telling the Station what name to use, what eventName to use when logging, the strings it should look for to validate a client for machine use, etc. The fields are each documented in the comments section of the database.

Use phpMyAdmin to insert records to this table to establish your configuration.

Note: We have included sample data in stationConfig_sampledata.sql. You can insert this data into your database if you want to see what our config looks like.

Tests

Note: To facilitate testing we supply two files of sample data: rawdata_sampledata.sql and stationConfig_sampledata.sql. You can load these two sample sets to see what the website reports look like if you are unable to get your own POSTs to work for some reason.

1. Test Logging:
    a. In the header of file rfidlog.php is a sample JSON. Use that JSON to construct several test packages.
    b. Post each to rfidlog.php. Look in the rawlog table to see your data.
    c. Use rfidclientactivity.php to see that the data is presented on the page.
2. Test Check in/out
    a. In the header of file rfidcheckin.php is a sample JSON. Use that JSON to construct several test packages. Use the same clientID in a few of the packages.
    b. Post each to rfidcheckin.php.
    c. Use rfidclientactivity.php to see that the data is presented on the page.
    d. Use checkinreport.php to see that your clientID has been checked in and then post again to see that has been checked out.
3. Other Reports
    a. From the rfidhome.html page, click on each link and see that the report shows the activity you have just performed.
4. Manual Checkout
    a. With at least one client Checked In, bring up checkinreport.php.
    b. Click on one of the names and see that the client is checked out.
5. Station Configuration
    a. Call the URL fdbGetStationConfig.php?cmd=-1
    b. See that the JSON returned has a list of stations and their device types. This is used by the admin configuration app.
    c. Call the URL fdbGetStationConfig.php?cmd=<<any valid device type>>
    d. See that the JSON returned is a set of configuration parameters for the Station.