OpenAquaponics

Application SDK

**Table of Contents**

[1. OAServer Database Table Definitions 3](#_Toc347065758)

[1.1. OAServer .OAUserInfo 3](#_Toc347065759)

[1.2. OAServer .OAUserPass 3](#_Toc347065760)

[1.3. OAServer .OASystemCfg 3](#_Toc347065761)

[1.4. OAServer .OANodeCfg 4](#_Toc347065762)

[2. OAData Database Table Definitions 4](#_Toc347065763)

[2.1. OAData.{sNodeId} 4](#_Toc347065764)

[3. API v1 5](#_Toc347065765)

# OAServer Database Table Definitions

## OAServer .OAUserInfo

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Column Name (Default) | Required | Key | Type | Length | Description |
| sUsername | REQUIRED | PRIMARY | VARCHAR | 32 | The username |
| sEmail | REQUIRED | N/A | VARCHAR | 64 | The email address of the user |
| sFirstname | REQUIRED | N/A | VARCHAR | 32 | The first name |
| sLastName | NOT REQUIRED | N/A | VARCHAR | 32 | The last name |
| sAddress | NOT REQUIRED | N/A | VARCHAR | 128 | The address line 1 |
| sAddress2 | NOT REQUIRED | N/A | VARCHAR | 128 | The address line 2 |
| sCity | NOT REQUIRED | N/A | VARCHAR | 32 | The city |
| sState | NOT REQUIRED | N/A | VARCHAR | 32 | The state |
| sZip | NOT REQUIRED | N/A | VARCHAR | 16 | The zip code |
| sPhone | NOT REQUIRED | N/A | VARCHAR | 16 | The phone number |
| bPublic (TRUE) | NOT REQUIRED | N/A | BOOL | -- | The user account is public or private |
| bEnable (TRUE) | NOT REQUIRED | N/A | BOOL | -- | The user account is enabled or disabled |

CREATE TABLE test\_OAServer.OAUserInfo ( sUsername VARCHAR(32) PRIMARY KEY NOT NULL UNIQUE, sEmail VARCHAR(64) NOT NULL UNIQUE, sFirstName VARCHAR(32) NOT NULL, sLastName VARCHAR(32), sAddress VARCHAR(128), sAddress2 VARCHAR(128), sCity VARCHAR(32), sState VARCHAR(32), sZip VARCHAR(16), sPhone VARCHAR(16), bPublic BOOL NOT NULL DEFAULT 1, bEnable BOOL NOT NULL DEFAULT 1);

## OAServer .OAUserPass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Column Name (Default) | Required | Key | Type | Length | Description |
| sUsername | REQUIRED | PRIMARY | VARCHAR | 32 | The username |
| sPassword | REQUIRED | N/A | VARCHAR | 32 | A salted hash of the password |

CREATE TABLE test\_OAServer.OAPassword ( sUsername VARCHAR(32) PRIMARY KEY NOT NULL UNIQUE, sPassword VARCHAR(32));

## OAServer .OASystemCfg

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Column Name (Default) | Required | Key | Type | Length | Description |
| sUsername | REQUIRED | FOREIGN | VARCHAR | 32 | The owner of the system |
| sSystemId | REQUIRED | PRIMARY | VARCHAR | 16 | Unique ID of the system |
| sGroupId | NOT REQUIRED | FOREIGN | VARCHAR | 16 | Unique ID of the group |
| sDescription | NOT REQUIRED | N/A | VARCHAR | 256 | System description |
| bPublic (TRUE) | NOT REQUIRED | N/A | BOOL | -- | The user account is public or private |
| bEnable (TRUE) | NOT REQUIRED | N/A | BOOL | -- | The user account is enabled or disabled |

CREATE TABLE test\_OAServer.OASystemCfg ( sUsername VARCHAR(32) NOT NULL, sSystemId VARCHAR(16) PRIMARY KEY NOT NULL UNIQUE, sGroupId VARCHAR(16), sDescription VARCHAR(256), bPublic BOOL NOT NULL DEFAULT 1, bEnable BOOL NOT NULL DEFAULT 1);

## OAServer .OANodeCfg

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Column Name (Default) | Required | Key | Type | Length | Description |
| sUsername | REQUIRED | FOREIGN | VARCHAR | 32 | The owner of the system |
| sSystemId | REQUIRED | FOREIGN | VARCHAR | 16 | Unique ID of the system |
| sNodeId | REQUIRED | PRIMARY | VARCHAR | 16 | Unique ID of the node |
| sChannelNames | NOT REQUIRED | N/A | VARCHAR | 256 | The last name |
| sChannelUnits | NOT REQUIRED | N/A | VARCHAR | 64 | The address line 1 |
| mPollingPeriod (60) | NOT REQUIRED | N/A | INT | -- | The address line 2 |
| sDescription | NOT REQUIRED | N/A | VARCHAR | 256 | The city |
| bPublic (TRUE) | NOT REQUIRED | N/A | BOOL | -- | The user account is public or private |
| bEnable (TRUE) | NOT REQUIRED | N/A | BOOL | -- | The user account is enabled or disabled |

CREATE TABLE test\_OAServer.OANodeCfg ( sUsername VARCHAR(32) NOT NULL , sSystemId VARCHAR(16) NOT NULL, sNodeId VARCHAR(16) PRIMARY KEY NOT NULL UNIQUE, sChannelNames VARCHAR(256), sChannelUnits VARCHAR(64), mPollingPeriod INTEGER NOT NULL DEFAULT 60, sDescription VARCHAR(256), bPublic BOOL NOT NULL DEFAULT 1, bEnable BOOL NOT NULL DEFAULT 1);

# OAData Database Table Definitions

## OAData.{sNodeId}

* sample number (PK) (R, automatic)
* timetag sample (PK) (R) UTC Zulu Time // Time will be the only UID tag I can trigger off of to know when data samples are being resent either because of a duplicate or because OANode was reset in the middle of a bulk transfer
* “sam,sam,sam,sam”

CREATE TABLE "OANodeData" ("mCnt" INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL UNIQUE , "mDatabaseTimetag" DATETIME NOT NULL DEFAULT CURRENT\_TIMESTAMP , "mNodeId" VARCHAR(16) NOT NULL , "mUsername" VARCHAR(32) NOT NULL , "mSampleTimetag" DATETIME, "mVal01" FLOAT, "mVal02" FLOAT, "mVal03" FLOAT, "mVal04" FLOAT, "mVal05" FLOAT, "mVal06" FLOAT, "mVal07" FLOAT, "mVal08" FLOAT, FOREIGN KEY("mUsername") REFERENCES UserInfo("mUsername"), FOREIGN KEY("mNodeId") REFERENCES OANodeConfig("mNodeId"))

# API v1

Paths are relative to: [www.api.opensourceaquaponics.com/v1/](http://www.api.opensourceaquaponics.com/v1/)

UID = User ID

SID = System ID

NID = Node ID

TID = Task ID

AID = Accounting ID

|  |  |  |  |
| --- | --- | --- | --- |
| TYPE | RESOURCE | AUTH | DESCRIPTION |
| -------------------------------------------------------------------------------------------------------------------------------------------------------- | | | |
| GET | /OACommunity | FALSE | High level statistics, number of users, total up time, number of groups, number of systems, number of nodes, a sample plot from a public account, etc. |
| GET | /OACommunity/OAUsers | FALSE | Gets a listing of the public users, should support some kind of search feature. |
| -------------------------------------------------------------------------------------------------------------------------------------------------------- | | | |
| GET | /OAGroups | FALSE | List of groups users can be members of (ie: personal, commercial, research, Utah Aquaponics Grows Association, etc) |
| PUT | /OAGroups | TRUE | Create a new group |
| -------------------------------------------------------------------------------------------------------------------------------------------------------- | | | |
| GET | /{UID}/OAUser | FALSE | Gets basic user information (nothing sensitive) |
| GET | /{UID}/OAUser | TRUE | Gets all user information (potentially sensitive) |
| PUT | /{UID}/OAUser | TRUE | Puts the updated user information |
| POST | /{UID}/OAUser | FALSE | Attempts to create new user and authenticates via email; Locks out IP after ‘x’ attempts, etc. |
| -------------------------------------------------------------------------------------------------------------------------------------------------------- | | | |
| GET | /{UID}/OASystems | TRUE/FALSE | Get the list of OASystems attached to this user; Can be public/private |
| POST | /{UID}/OASystems | TRUE | Create a new OASystem IDs for this system, system name, system description, group association, public/private, etc. |
| GET | /{UID}/OASystems/{SID} | TRUE/FALSE | Get the list of OANode IDs for this system, system name, system description, group association, etc; Can be public/private |
| PUT | /{UID}/OASystems/{SID} | TRUE | Update the list of OANode IDs for this system, system name, system description, group association, public/private, etc. |
| DELETE | /{UID}/OASystems/{SID} | TRUE | Deletes a system pairing of OANodes and gives the OANode no grouping |
| -------------------------------------------------------------------------------------------------------------------------------------------------------- | | | |
| GET | /{UID}/OANodes | TRUE/FALSE | Get the list of OANodes attached to this user; Can be public/private |
| POST | /{UID}/OANodes | TRUE | Create a new OANode ID for this system, system name, system description, group association, public/private, etc. Also creates the OANode\data MySQL table. |
| GET | /{UID}/OANodes/{NID} | TRUE/FALSE | Get the OANode IDs, node name, node description, hardware configuration (pins, units, etc), get the OATasks attached to this node, group association, etc; Can be public/private |
| PUT | /{UID}/OANodes/{NID} | TRUE | Update the node name, node description, hardware configuration (pins, units, etc), get the OATasks attached to this node, group association, public/private, etc. |
| DELETE | /{UID}/OANodes/{NID} | TRUE | Deletes a OANode from the database and all associated data |
| GET | /{UID}/OANodes/{NID}/data | TRUE/FALSE | Get the OANode IDs sampled data, allows some type of data thresholding and selection; Can be public/private |
| POST | /{UID}/OANodes/{NID}/data | TRUE | Creates OANode sample data onto the server |
| PUT | /{UID}/OANodes/{NID}/data | TRUE | Modifies OANode sample data onto the server |
| DELETE | /{UID}/OANodes/{NID}/data | TRUE | Deletes the OANode data from the database, can support some data selection (ie: by date, by row, etc) |
| -------------------------------------------------------------------------------------------------------------------------------------------------------- | | | |
| GET | /{UID}/OATasks | TRUE | List all of the monitoring tasks associated with the user |
| POST | /{UID}/OATasks | TRUE | Create a new monitoring task associated with a OANode or list of OANodes |
| GET | /{UID}/OATasks/{TID} | TRUE | Get the monitoring task associated with a OANode or list of OANodes |
| PUT | /{UID}/OATasks/{TID} | TRUE | Update a particular monitoring task associated with a OANode |
| DELETE | /{UID}/OATasks/{TID} | TRUE | Delete a particular monitoring task |
| -------------------------------------------------------------------------------------------------------------------------------------------------------- | | | |
| GET | /{UID}/OAAccounting | TRUE | List all of the monetary transactions of the user; supports selectable sets by UID, SID, NID, date, amount, etc |
| POST | /{UID}/OAAccounting | TRUE | Create new transaction in the list |
| GET | /{UID}/OAAccounting/{AID} | TRUE | Get the particular transaction |
| PUT | /{UID}/OAAccounting/{AID} | TRUE | Update transaction |
| DELETE | /{UID}/OAAccounting/{AID} | TRUE | Delete a transaction from the list |
| -------------------------------------------------------------------------------------------------------------------------------------------------------- | | | |
|  | | | |
|  | | | |
|  | | | |
|  |  |  |  |

---------------------------------------------------------------

Question:

* What do you do when an OANode is reconfigured? How do you update the database? Give it another unique tag and mark the node as inactive?

R = required

NR – not required

PK – Primary Key

FK – Foreign Key

OANodeEvent (this is also a way to create automate events)

* event number (PK) (R, automatic)
* timetag database (R, automatic)
* node ID (FK) (R)
* event type (ENUM: add plant, add fish, triggers (water, temp, humidity), execution time overrun, ex?) (R) (FK)
* severity (R) – (info, warning, critical, failure, etc)
* send SMS (R)
* send Twitter (R)
* send email (R)
* description (NR)
* ??

Event Type

* Event number (PK) (R, automatic)
* Name (R)
* Description (NR)

Accounting

* transaction ID (PK) (R, automatic)
* username (FK) (R)
* node ID (FK) (NR)
* transaction amount (R)
* description (NR)

CREATE TABLE "Accounting" ("mCnt" INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL UNIQUE , "mUsername" VARCHAR(32) NOT NULL , "mNodeId" VARCHAR(16), "mAmount" FLOAT NOT NULL DEFAULT 0.0, "mDescription" VARCHAR(256), FOREIGN KEY("mUsername") REFERENCES UserInfo("mUsername"), FOREIGN KEY("mNodeId") REFERENCES OANodeConfig("mNodeId"))