

Ignition 8.0.12: Using Database User Source in Manual Mode

Pat Dixon, PE, PMP, www.DPAS-INC.com

8/5/20

The following are my observations for setting up an Ignition v8.0.12 user source as database in manual mode. This is incomplete and there are several points that require further research and possible correction, but this approach is working and no other documentation or information has been found on this subject.

There needs to be a database connection to tables with the following fields:

List of USERS:

FIELD	PRIMARY KEY	TYPE	DESCRIPTION
id	YES	INT	A unique key identifier for the user
username		TEXT	The login username
fname		TEXT	The first name of the user
lname		TEXT	The last name of the user
passwd		VARCHAR(255)	The password, which should be encrypted
schedule		TEXT	An assignment of the schedule for which this user can login

List of ROLES:

FIELD	PRIMARY KEY	TYPE	DESCRIPTION
id	YES	INT	A unique key identifier for the role
role_name		TEXT	The name of the role

List of ROLE ASSIGNMENTS, allows multiple roles assigned to a user:

FIELD	PRIMARY KEY	TYPE	DESCRIPTION
user_id		INT	Identifier for username
role_id		INT	Identifier for role

List of CONTACT INFO, allows multiple contacts to a user:

FIELD	PRIMARY KEY	TYPE	DESCRIPTION
user_id		INT	Identifier for username
Contact_Type		TEXT	Can be either 'email' or 'sms'
Contact_Info		TEXT	The pertinent contact info

Once these tables are created in your database, fill them in with the appropriate values and create a database connection from the Ignition gateway to this database.

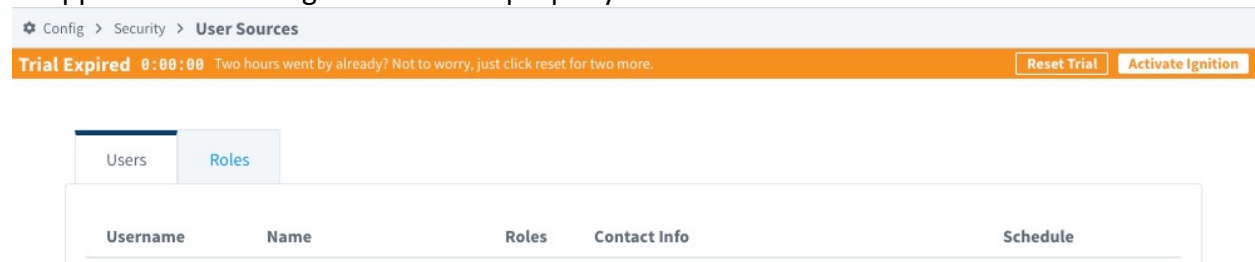
Once there is a connection to the database with these tables created and populated, we can create the User Source. In my case, I have created a user source with a Failover Source of default, which is the default internal Ignition user source. I specified the Failover Mode for Soft. The reason I did this is to designation the database source as the list of users for non-administrative roles, so that a list of those users can be imported into my database. I want this database to be maintained separately from any administrative accounts.

(It should be noted that if you try to import a CSV file into MySQL, the import of a CSV doesn't work if it is encoded with Byte Order Mark (BOM). Unfortunately, Excel saves a CSV as UTF-8 with BOM. I use Notepad++, which allows opening the CSV file and using the Encode menu to select UTF-* (with no BOM). This allows import into a MySQL table.)

With this arrangement, the users that login will be authenticated through the list of users I have in my database tables. When an administrator logs in, this account will not be found in the list of users but since there is a soft failover the authentication will look to the default source and find the authentication there.

Now we get to the complicated part, which is not well documented. The Manual Mode configuration in the gateway requires specifying SQL to return the right information. While there are example queries given, it can be tricky to get it right.

The field names that are returned in the SQL need to match what the gateway expects in order to appear in the Manage Users tables properly.



It may not match exactly the field names you have used in the tables. The field names that are needed are:

List of USERS in Manage Users table:

FIELD
username
firstname
lastname
schedule
contact_type
contact_info_

For the list of roles in the Manage Users table, it is not clear where this comes from but the User's Roles Query must be correct or else the User list will not be correct. An SQL error in this one will result in having no users listed. What has been working for me is:

```
select role from Database.Roles_Table where ID=?
```

Now it needs to be explained that these are SQL queries and can be used as any other SQL to cast a field as a different name or connect through JOIN. I have found this essential. For example, in the List Users Query I have a table of users with field names 'Name_First' and 'Name_Last', but the gateway needs 'firstname' and 'lastname'. Therefore, I cast them in the SQL as follows:

```
SELECT username, Name_First as firstname, Name_Last as lastname, schedule FROM Database.User_Table
```

For the contact information, since it is in a different table from the user information the Contact Info Query looks like this, which is very different from the example script given:

```
SELECT Database.Contact_Table.Contact_Type, Database.Contact_Table.Contact_Info FROM Database.Contact_Table inner join Database.User_Table on Database.User_Table.ID_Number = Database.Contact_Table.user_id WHERE Database.User_Table.Username=?
```

Going through each of the SQL scripts for Manual Mode, what has been working for me is:

SCRIPT	SQL	EXPLANATION
Authentication Query	SELECT * FROM Database.User_Table WHERE username = ? AND password = MD5(?)	The * wildcard seems to work, it doesn't appear that you have to explicitly list each field name to return as shown in the example SQL. MD5 allows for using encrypted passwords in the table
Badge Authentication Query		I don't think I will be using badge authentication in my application. This is an optional configuration
List Roles Query		This doesn't appear to do anything. I have a list of roles in the Manager Users table for Roles with this SQL blank
User's Roles Query	SELECT role from Database.Roles_Table where ID=?	I am not sure how this works, but if you mess this up no users will appear under Manage Users
List Users Query	SELECT username, Name_First as firstname, Name_Last as lastname, schedule FROM Database.User_Table	This returns some of the fields that are seen under Manage Users. I type-cast the first and last names to match the field names that the gateway is looking for

Contact Info Query	SELECT Database.Contact_Table.Contact_Type, Database.Contact_Table.Contact_Info FROM Database.Contact_Table inner join Database.User_Table on Database.User_Table.ID_Number = Database.Contact_Table.user_id WHERE Database.User_Table.Username=?	Unless you pack the contact info fields into the user table and only allow one type of contact info to be used for each user, I see no way to do this without INNER JOIN. This SQL allows a separate table of contact info with multiple types of contacts for each user. The SQL example given on the gateway configuration page is insufficient.
Schedule Adjustment Query		The Manage Users table just looks at a text field in the user table called 'schedule'. It is not obvious how this SQL works. I would like to only allow user login during scheduled times, so I will probably have to figure out how this works.
Extra Properties Query		This is optional

With all of that, I have an application that allows me to login as a user listed in the table or an administrator configured internally in the gateway. I have not yet tested roles and suspect I will have to adjust some of these queries.