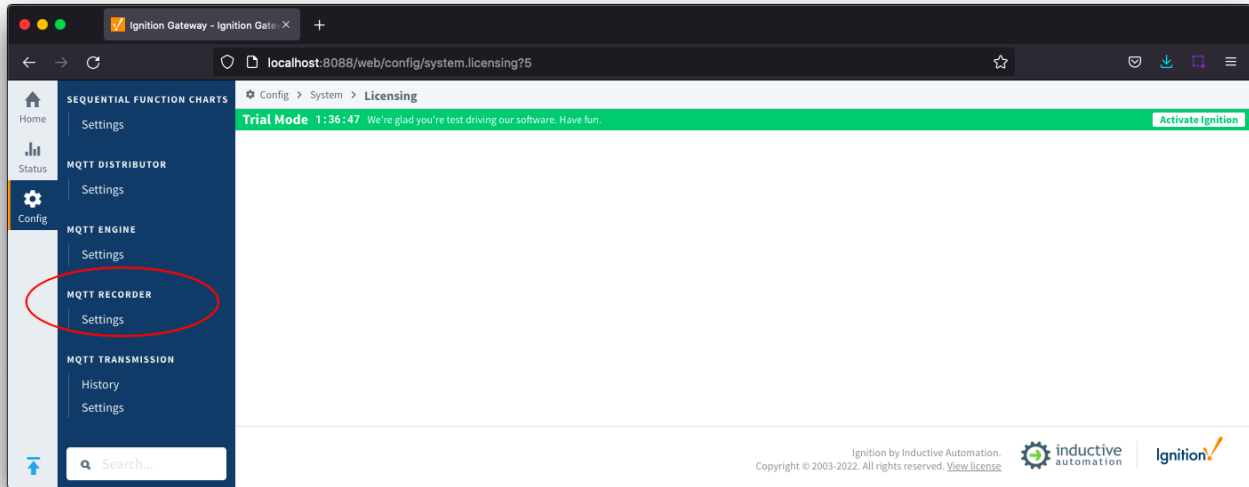# MR: Configuration

MQTT Recorder provides a configuration section to the Ignition Gateway and this can been seen in the left side bar of the Ignition Gateway web UI.

The configuration options for each of the tabs - General and Tables - are detailed below.
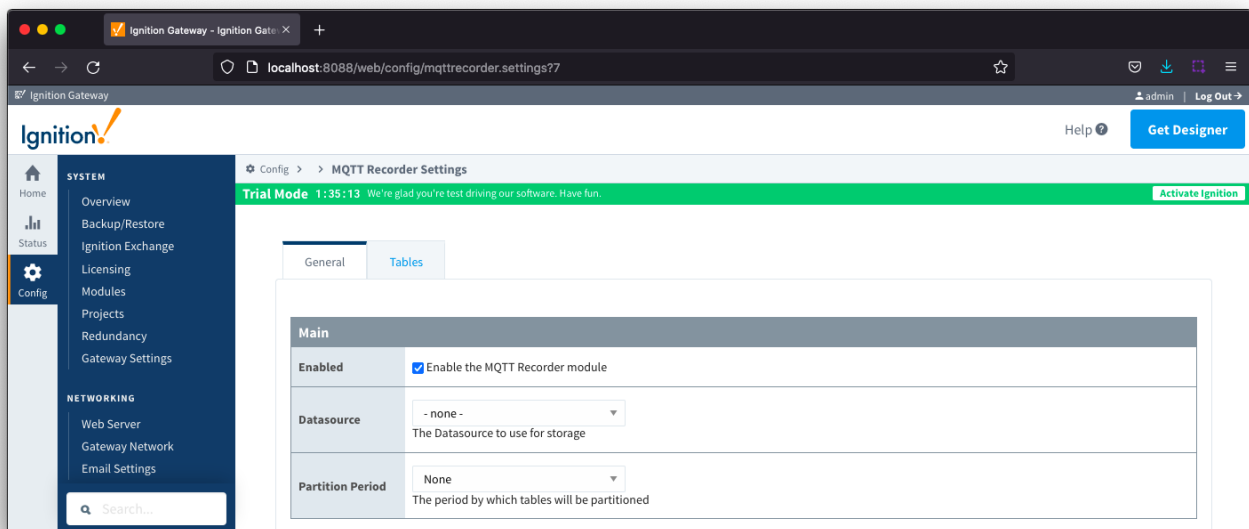
General

MQTT Recorder receives records from MQTT Engine and automatically generates database tables and populates the rows within the tables with the records as they are received.

Records are published in a Sparkplug-like format on a topic such as spBv1.0/g*roup*/NRECORD/*edgenode* or spBv1.0/g*roup*/DRECORD/e*dgenode*/d*evice* and a payload in JSON format containing the metric name as the Record Type and associated tags such as:

{"timestamp":1671052287150,"metrics":[{"name":"Event","timestamp":1671052286643,"dataType":"String","properties":{"Description":{"type":"PropertySet","value":{"fieldValue":{"type":"String","value":"MyEvent"}}},"RunNumber":{"type":"PropertySet","value":{"fieldValue":{"type":"Int32","value":1}}},"DataSource":{"type":"PropertySet","value":{"fieldValue":{"type":"String","value":"true"}}}}}],"seq":4}

These are the general MQTT Recorder configuration settings and there is a single Main section available.
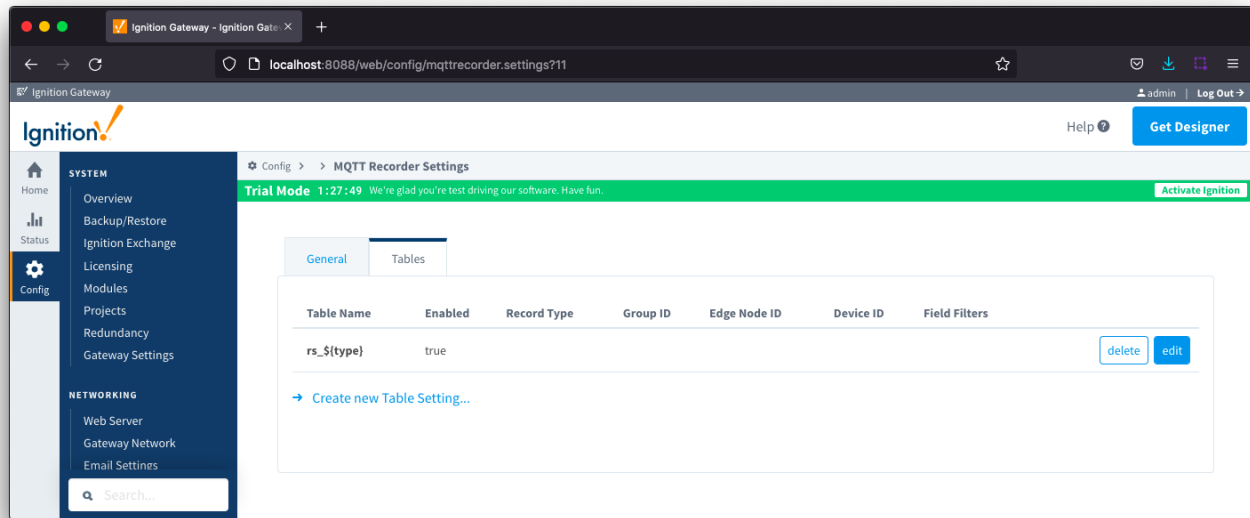
General - Main

- **Enabled**
  - Checkbox to enable/disable the MQTT Recorder module. Selected by default.
- **Datasource**
  - A dropdown list to select the database connection to use for record storage. This list is populated with the database connections set up within the Ignition Gateway.
- **Partition Period**
  - The partitioning period to use for record tables within database. Each table name will be appended with a date indicating the partition time period. For example, if table were created for "alarm" records on December 10, 2022 at 2:00 AM, the table name for each partition type would be:
    - None: *tablename*
    - Hourly: *tablename_2022_12_10_02*
    - Daily: *tablename_2022_12_10*
    - Weekly: *tablename_2022_49*
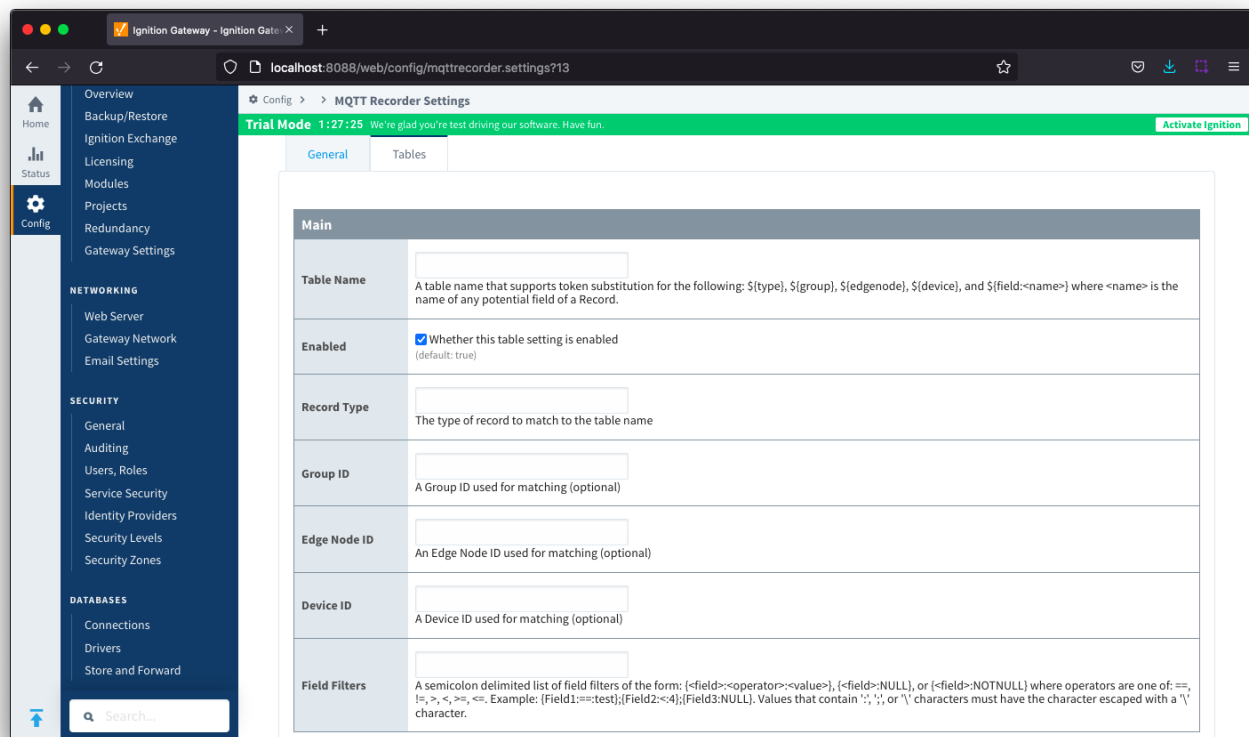   - Monthly: *tablename_2022_12*
    - Yearly: *tablename_2022*

Tables

These are the settings for the database tables that MQTT Recorder will insert the records into and there is a single Main section available.

By default, a single table will be defined for MQTT Recorder with a table name of rs_${type}. With this single table definition, a separate table will be created for each unique record type however you can create custom table names using the additional filters described below.

Tables - Main

- **Table Name**
  - The name of the table where records will be inserted. When creating a dynamic table name using tokens, the values must contain only valid characters allowed in the database table name.
  - This name can be statically defined or be dynamically build up using the following tokens
    - ${type} - The Record Type of the record being inserted
    - ${group} - The Sparkplug Group ID of the record being inserted
    - ${edgenode} - The Sparkplug Edge Node ID of the record being inserted
    - ${device} - The Sparkplug Device ID of the record being inserted
    - ${field:<name>} - The value of any field of the record being inserted where <name> is the name of the field
  - For example AAA_${group}_BBB_${edgenode}_${type}
- **Enabled**
  - Checkbox to enable/disable the table setting. Selected by default.
- **Record Type (Optional)**
  - Record Type to match to the table name.
  - If specified, only records with this Record Type will be placed in the table.
- **Group ID (Optional)**
  - Sparkplug Group ID to filter on.
  - If specified, only records with this Group ID will be placed in the table.
- **Edge Node ID (Optional)**
  - Sparkplug Edge Node ID to filter on.
  - If specified, only records with this Edge Node ID will be placed in the table.
- **Device ID (Optional)**
  - Sparkplug Device ID to filter on.
  - If specified, only records with this Device ID will be placed in the table.
- **Field Filters (Optional)**
  - Semicolon delimited list of field filters.
  - Each field filter is declared using one of the following forms: {<field>:<operator>:<value>}, {<field>:NULL}, or {<field>:NOTNULL}
    - <field> - the name of the field to filter on.
    - <operator> - the filter operator. Must be one of the following: ==, !=, >, <, >=, or <=
    - <value> - the value of the field.
  - Values that contain ':', ';', or '\' characters must have the character escaped with a leading '\' character.
  - Examples:
    - To only insert records that match a "serial_number" field equal to "0123456" the filter would be: {serial_number:==:0123456}
    - To only insert records that contain a "temp1" field the filter would be: {temp1:NOTNULL}
    - To exclude records that match a "unit_id" field equal to "112:008:32455" the filter would be: {unit_id:!=:112\:008\:32455}