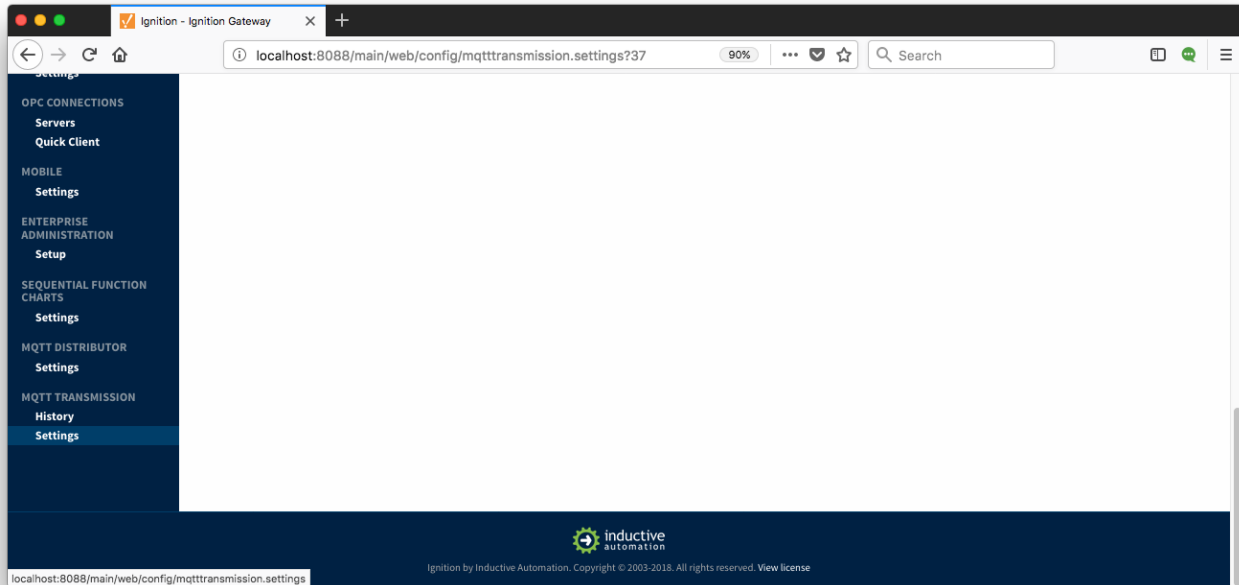


# MT: Configuration

MQTT Transmission provides a configuration section to the Ignition Gateway. These can be seen in the Configure section of the Ignition Gateway web UI. There are two configuration pages "History" and "Settings".



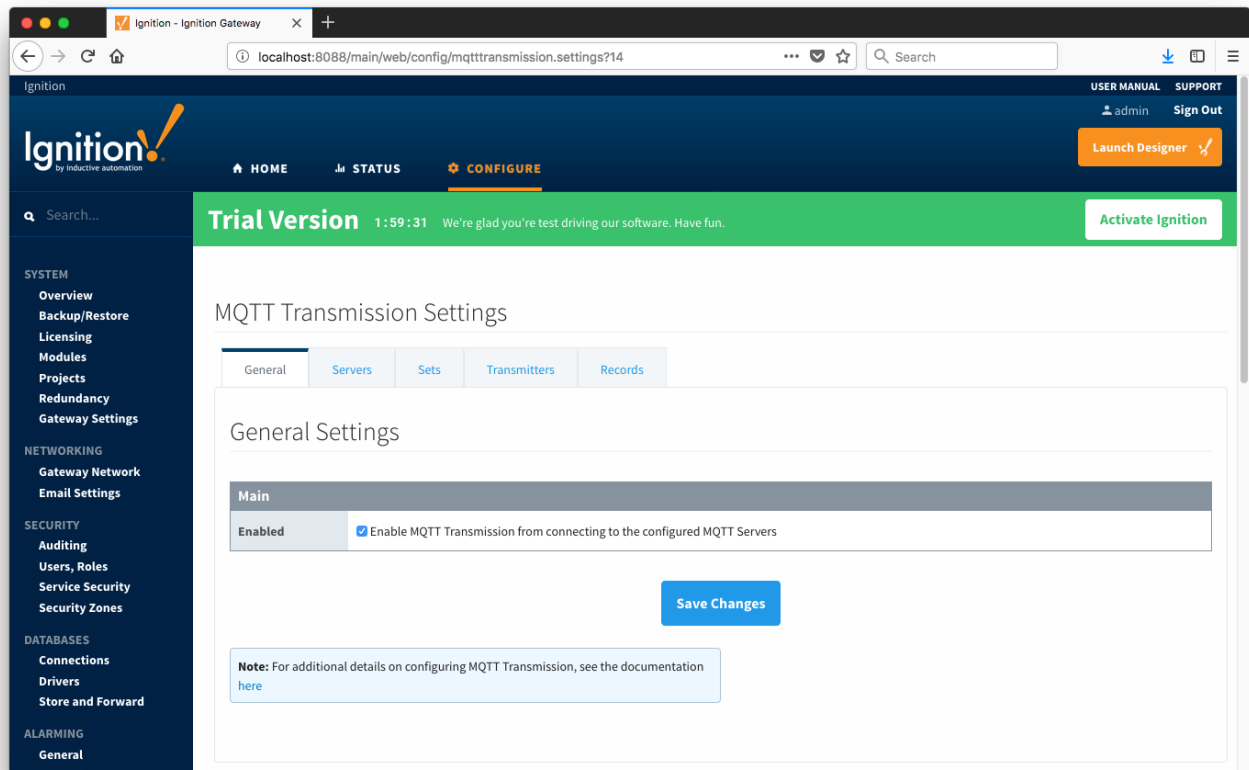
## Settings

Once in the "Settings" configuration section there are five tabs: General, Servers, Sets, Transmitters, and Records. Each of these tabs is described in detail in the following sections.

### General

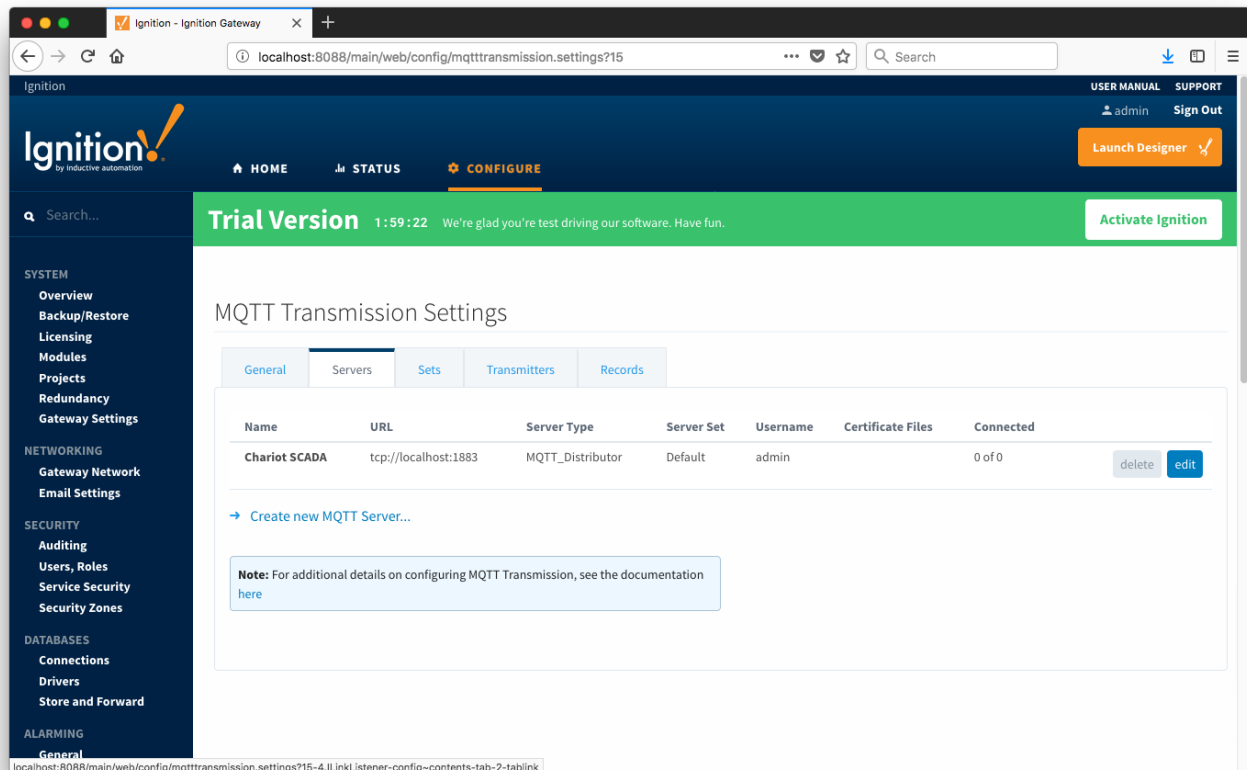
The General Settings tab contains advanced configuration options for the default MQTT Server settings created in the Servers tab. These are described below.

- **Enabled**
  - Whether or not to enable or disable MQTT Transmission from connecting to the configured MQTT Servers.
- **Primary Host ID**
  - The primary host ID to use for 'state' notifications. These notifications are used to notify MQTT Transmission if the primary backend application loses connection with the MQTT Server, meaning it has gone 'offline'. If MQTT Transmission is notified that the primary backend application has gone 'offline', it will close it's client connection with the MQTT server and walk to the next MQTT server defined in the set. If the primary host ID is not set, MQTT Transmission will not subscribe on the notification topics and not receive any 'state' notifications.



## Servers

The first tab is a list of MQTT Servers that MQTT Transmission should connect to. By default, MQTT Transmission is configured to connect to a local MQTT Distributor based MQTT Server. It is set up to connect to localhost, port 1883, using the default username/password pair of admin/changeme. Out of the box MQTT Transmission will work with MQTT Distributor and its default configuration.



Additional or alternative MQTT Servers can be configured in MQTT Transmission. Often times more than one will be configured to handle fail-over or in redundant or geographically distributed systems. The configuration options for servers are listed below.

## Main Properties

- **Name**
  - This is the friendly name of the MQTT Server used to easily identify it
- **URL**
  - This is the URL of the MQTT server. Its format is as follows: [protocol]://[location]:[port]. Each of these are shown below
    - protocol - Either tcp or ssl
    - location - The server location. e.g. localhost, [myserver.chariot.io](https://myserver.chariot.io), [mydomain.com](https://mydomain.com), etc
    - port - The port the MQTT Server is listening on. Generally this is 1883 if using TCP or 8883 if using SSL
- **Server Type**
  - This is the type of MQTT Server to connect to
    - Chariot - If connecting to a Cirrus Link Chariot on-premise or Chariot cloud based MQTT Server
    - MQTT\_Distributor - If connecting to a Ignition MQTT Distributor server
    - Third\_Party - If connecting to a third party 3.1.1 compliant MQTT Server
    - AWS\_Greengrass - If connecting to an AWS Greengrass Server.
      - Note: This server type should not be used when using MQTT Transmission in conjunction with MQTT Engine.
- **Server Set**
  - The Set that this server is a member of.
- **Username**
  - The MQTT username to use in the MQTT connect packet. This is required if the MQTT Server to connect to requires it.
- **Password**
  - The MQTT password to use in the MQTT connect packet. This is required if the MQTT Server to connect to requires it.

## TLS Properties

- **Certificates**
  - The server certificates to use if required. These are generally only required when connecting using TLS and the MQTT server does not have a genuine certificate issued by a trusted certificate authority.
  - CA certificate must always end in 'ca.pem'
  - If using client side certificates (i.e. a public/private keypair and CA cert), make sure the following naming conventions are followed.
    - Edge/Device certificate must end in 'cert.pem'
    - Edge/Device private key must end in 'private.key'
- **Password**
  - A password associated with the certificate's private key.

## Advanced Properties

- **Client ID**
  - Optional MQTT client ID to use. If specified this will be used in the MQTT Engine connect packet when connecting to the server. If left blank, a random client ID will be create of the form 'IgnitionTarget-xxxxxxx-xxxx-xxxx'
- **Keep Alive**
  - The MQTT client keep alive time (in seconds).

Clicking on the 'Create new MQTT Server...' link will bring up the following form to add a new Server.

The screenshot shows the Ignition Gateway web interface. The left sidebar contains a navigation menu with categories: NETWORKING (Gateway Network, Email Settings), SECURITY (Auditing, Users, Roles, Service Security, Security Zones), DATABASES (Connections, Drivers, Store and Forward), ALARMING (General, Journal, Notification, On-Call Rosters, Schedules), TAGS (History, Realtime), OPC-UA SERVER (Certificates, Devices, Settings), OPC CONNECTIONS (Servers, Quick Client), MOBILE (Settings), ENTERPRISE ADMINISTRATION (Setup), SEQUENTIAL FUNCTION CHARTS (Settings), MQTT DISTRIBUTOR (Settings), and MQTT TRANSMISSION (History, Settings). The main content area is titled 'localhost:8088/main/web/config/mqtttransmission.settings?25'. It features a 'Main' section with fields for Name (New Mqtt Server), URL, Server Type (MQTT\_Distributor), Server Set (Default), Username, Password, and a re-type password field. Below this is a 'TLS' section with fields for Certificate Files, Password, and a re-type password field. A checkbox labeled 'Show advanced properties' is checked, revealing an 'Advanced' section with fields for Client ID and Keep Alive (30). A blue button at the bottom right says 'Create New MQTT Server'.

Main	
Name	New Mqtt Server The friendly name of this MQTT Server
URL	<input type="text"/> The URL of this MQTT Server. Should be of the form tcp://mydomain.com:1883 or ssl://mydomain.com:8883
Server Type	MQTT_Distributor The type of MQTT Server to connect to (default: MQTT_Distributor)
Server Set	Default The Server Set this MQTT Server is associated with
Username	<input type="text"/> The username for this MQTT connection if required by the MQTT Server (optional)
Password	<input type="password"/> The password for this MQTT connection if required by the MQTT Server (optional)
Password	<input type="password"/> Re-type password for verification.

TLS	
Certificate Files	<input type="button" value="Browse..."/> No file selected. Files: <input type="text"/>
Password	<input type="password"/> The password associated with the certificate's private key (optional)
Password	<input type="password"/> Re-type password for verification.

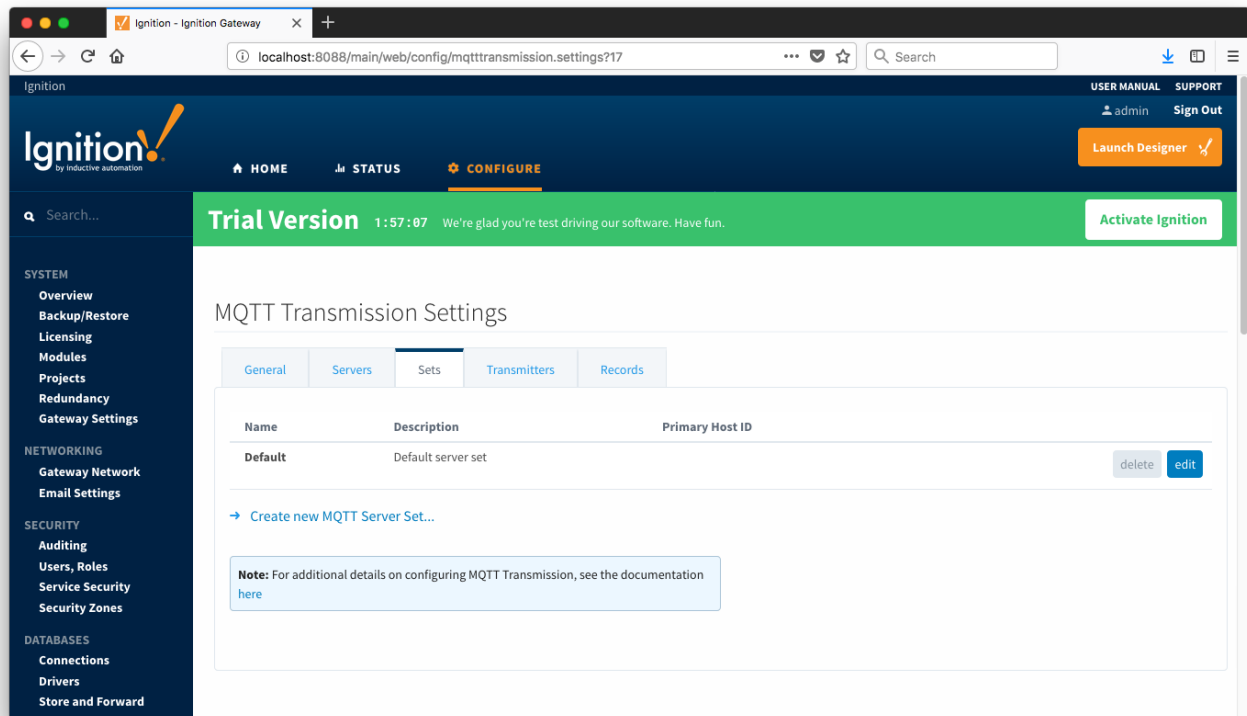
☒ Show advanced properties

Advanced	
Client ID	<input type="text"/> The MQTT Client ID for this MQTT connection. If left blank one will be auto-generated (optional)
Keep Alive	30 The MQTT Client keep alive time (in seconds) (default: 30)

Create New MQTT Server

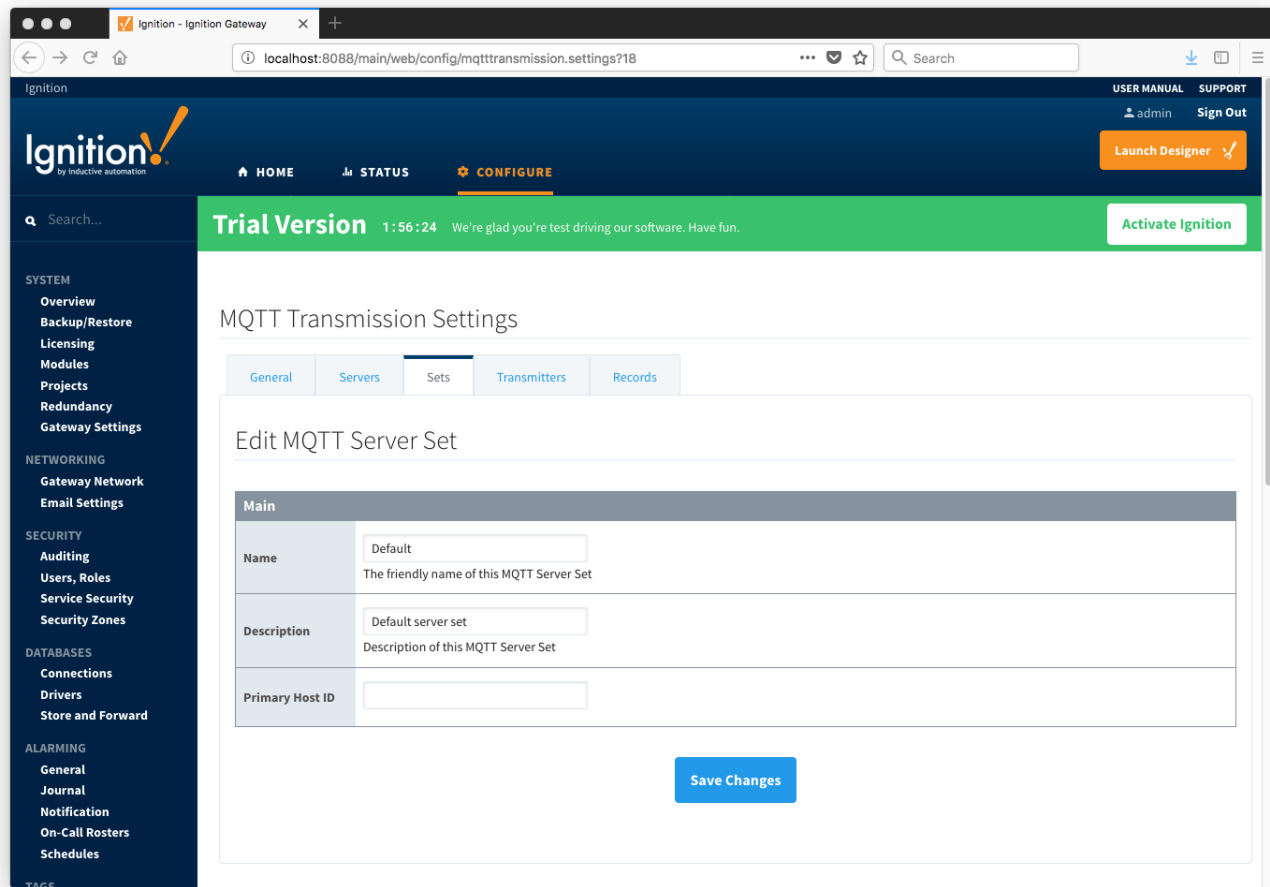
## Sets

The Sets tab contains a list of server sets. Each set represents a logical grouping of MQTT servers. When a set is referenced by a Transmitter configuration a single connection to one of the servers in the set will be maintained. The other servers will act as failover in the case that a connection with the first is lost. Server sets are disjoint, meaning that a single MQTT server cannot be in more than one set.



Additional sets can be configured in MQTT Transmission for situations where multiple Custom Settings will need to point to different sets (or groupings) of MQTT servers. The configuration options for sets are listed below.

- **Name**
  - This is the friendly name of the set used to easily identify it.
- **Description**
  - This is a friendly description of the set.
- **Primary Host ID**
  - The primary host ID to use for 'state' notifications. These notifications are used to notify MQTT Transmission if the primary backend application loses connection with the MQTT Server, meaning it has gone 'offline'. If MQTT Transmission is notified that the primary backend application has gone 'offline', it will close it's client connection with the MQTT server and walk to the next MQTT server defined in the set. If the primary host ID is not set, MQTT Transmission will not subscribe on the notification topics and not receive any 'state' notifications.



## Transmitters

Transmitters are the agents within MQTT Transmission that monitor tags, convert them to Sparkplug Messages, and publish them to an MQTT Server. Each transmitter is configured with a server Set and will attempt to maintain an MQTT client connection with one server in that Set at all times.

There are two types of Transmitters: Default and Custom. These types are explained in more detail below.

### Default Transmitter

MQTT Transmission provides a single Default Transmitter. The Tags that this Transmitter monitors must be placed in the "Edge Nodes" folder within the configured Tag Provider. Their Tag paths must conform to the strict structure of:

```
<Provider>/Edge Nodes/<GroupID>/<EdgeNodeID>/<DeviceID>
```

The individual elements of the above tag path (Group ID, Edge Node ID, and Device ID) are used when building up the MQTT message payload that will represent the Tags.

The configuration options for the Default Transmitter are listed below.:

- **Tag Provider Name**
  - The name of the tag provider that Transmission will monitor. By default this is the Ignition 'default' provider.
- **Tag Pacing Period**
  - The buffer period for outgoing Transmission messages in milliseconds. The default is 1000ms. This means when a tag change event is detected 1000ms will elapse before an MQTT message is sent. This allows additional tag change events to be buffered and put into the message and in turn reduce the number of generated MQTT messages.
- **Set**
  - The Set that the default MQTT Transmission client will connect to.
- **Discovery Delay**
  - An optional startup delay, in milliseconds, to wait before scanning for Tags to monitor. This is useful when Tags are dynamically created on initial startup, as is the case when using the MQTT Engine module.
- **History Store**
  - The MQTT Transmission History Store to use with the Default Transmitter.
- **Aliased Tags**

- Whether to use aliases for tag names when published data messages as tag values change. This is used to optimize payload size when publishing data.
- **Compression**
  - The algorithm to use to compress payloads before they are published to the MQTT Server. If 'NONE' is selected then compression is disabled.
- **Block Commands**
  - Whether to block commands and tag writes received as messages from the MQTT server.
- **Convert UDTs**
  - Whether to convert UDT members to normal Tags before publishing. If enabled the Tags representing the UDT member will retain their member path prefixed by the UDT Instance name.

MQTT Transmission Settings

General Servers Sets Transmitters Records

Default Custom

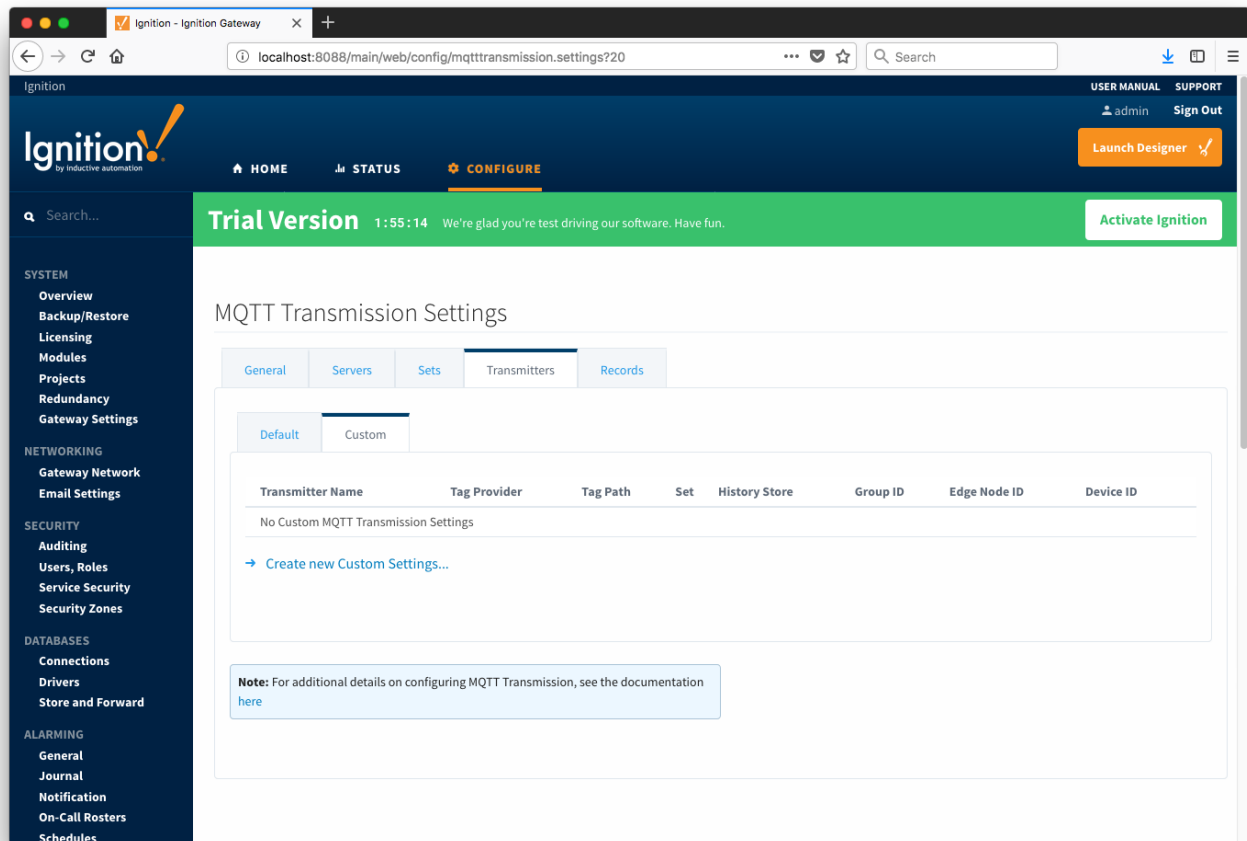
**Configuration**

Tag Provider Name	default The Name of the tag provider
Tag Pacing Period	1000 The waiting period in milliseconds after an initial tag change event before publishing all changed tags
Set	Default The MQTT Server Set to use with the Default Transmitter
Discovery Delay	0 The Transmitter Discovery Delay in milliseconds. This is useful when using MQTT Engine as the tag provider (default: 0)
History Store	- none - The MQTT Transmission History Store to use with the Default Transmitter
In-Order History	<input type="checkbox"/> Flush history in-order (synchronously) before live data resumes (default: false)
Aliased Tags	<input type="checkbox"/> Use aliases for tag names to optimize payload sizes when publishing data
Compression	NONE The algorithm to use for compressing payloads before publishing
Block Commands	<input type="checkbox"/> Block incoming commands (writes) to Edge Node and Device Tags
Convert UDTs	<input type="checkbox"/> Converts UDT members to normal Tags before publishing

Save Changes

## Custom Transmitters

Custom Transmitters behave in much the same way as the Default Transmitter. However, unlike the Default Transmitter, Custom Transmitters are not bound to a specific "Edge Nodes" folder and do not have any required folder path structure. The Group ID, Edge Node ID and Device ID are explicitly defined in a Custom Transmitter configuration instead of inferred from the tag path, as in the Default Transmitter. This allows Custom Transmitters to point at any folder in any provider and monitor the contained tags and/or tag folders.



Each Custom Setting has the following fields

## Tag Settings

- **Transmitter Name**
  - The name of the Transmitter Setting.
- **Tag Provider Name**
  - The name of the tag provider that Transmission will monitor.
- **Tag Path**
  - An optional folder path under the tag provider where the root folder of the tags can be found.
- **Tag Pacing Period**
  - The buffer period for outgoing Transmission messages in milliseconds.
- **Set**
  - The Set that the Custom Settings client will connect to.
- **Discovery Delay**
  - An optional startup delay, in milliseconds, to wait before scanning for Tags to monitor. This is useful when Tags are dynamically created on initial startup, as is the case when using the MQTT Engine module.
- **History Store**
  - The MQTT Transmission History Store to use with the Default Transmitter.
- **Aliased Tags**
  - Whether to use aliases for tag names when published data messages as tag values change. This is used to optimize payload size when publishing data.
- **Compression**
  - The algorithm to use to compress payloads before they are published to the MQTT Server. If 'NONE' is selected then compression is disabled.
- **Block Commands**
  - Whether to block commands and tag writes received as messages from the MQTT server.
- **Convert UDTs**
  - Whether to convert UDT members to normal Tags before publishing. If enabled the Tags representing the UDT member will retain their member path prefixed by the UDT Instance name.

## Sparkplug Settings

- **Group ID**
  - An ID representing a logical grouping of MQTT Edge Of Network (EoN) Nodes and Devices into the infrastructure.
- **Edge Node ID**



- An ID that uniquely identifies the MQTT Edge Of Network (EoN) Node within the infrastructure.
- **Device ID**
  - An optional ID that uniquely identifies a Device within the infrastructure.

Note that if a 'Device ID' is not specified, any folder within the folder specified by the Tag Path will be considered a device folder and any Tags within it will be published as device Tags.

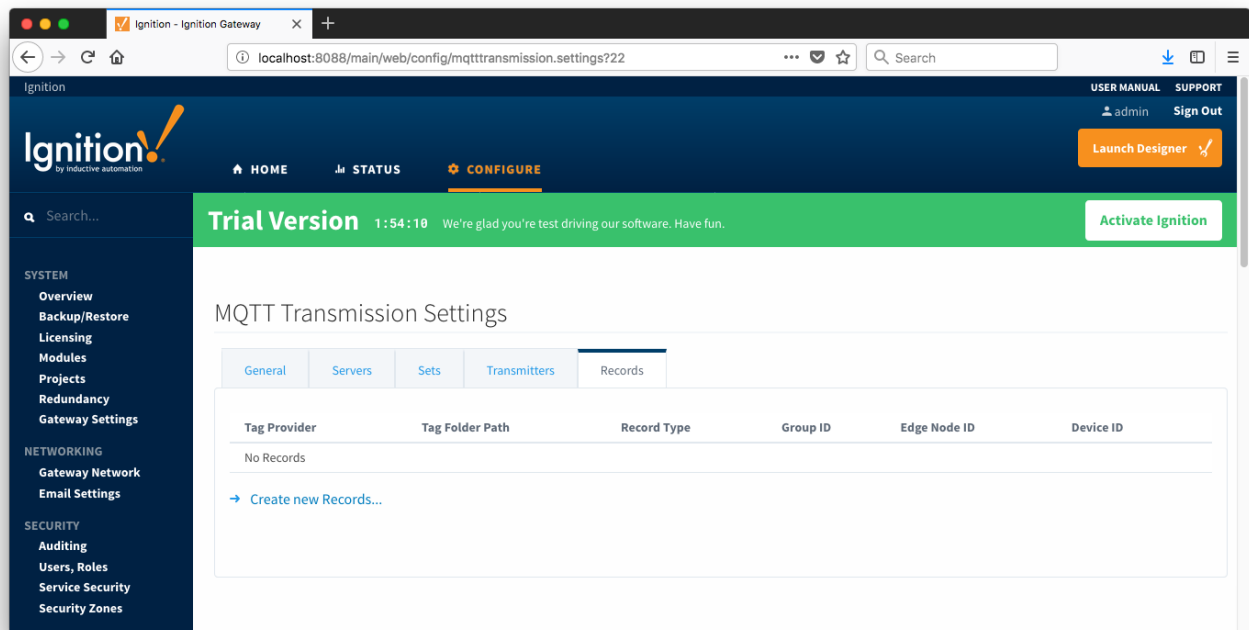
Clicking on the 'Create new Custom Settings..' link will bring up the following form to add a new Server.

The screenshot shows the Ignition Gateway web interface. The browser address bar indicates the URL is `localhost:8088/main/web/config/mqtttransmission.settings?21`. The interface has a dark blue header with the Ignition logo and navigation links: HOME, STATUS, and CONFIGURE. A green banner at the top right indicates a 'Trial Version' with a timer at '1:54:35' and a button to 'Activate Ignition'. On the left, a sidebar lists various system categories like SYSTEM, NETWORKING, SECURITY, DATABASES, ALARMING, TAGS, OPC-UA SERVER, and OPC CONNECTIONS. The main content area is titled 'MQTT Transmission Settings' and has tabs for General, Servers, Sets, Transmitters, and Records. The 'Transmitters' tab is active, showing a 'New Custom Settings' form. This form has two sub-tabs: 'Default' and 'Custom'. The 'Custom' sub-tab is selected, displaying a 'Tag Settings' table with the following fields:

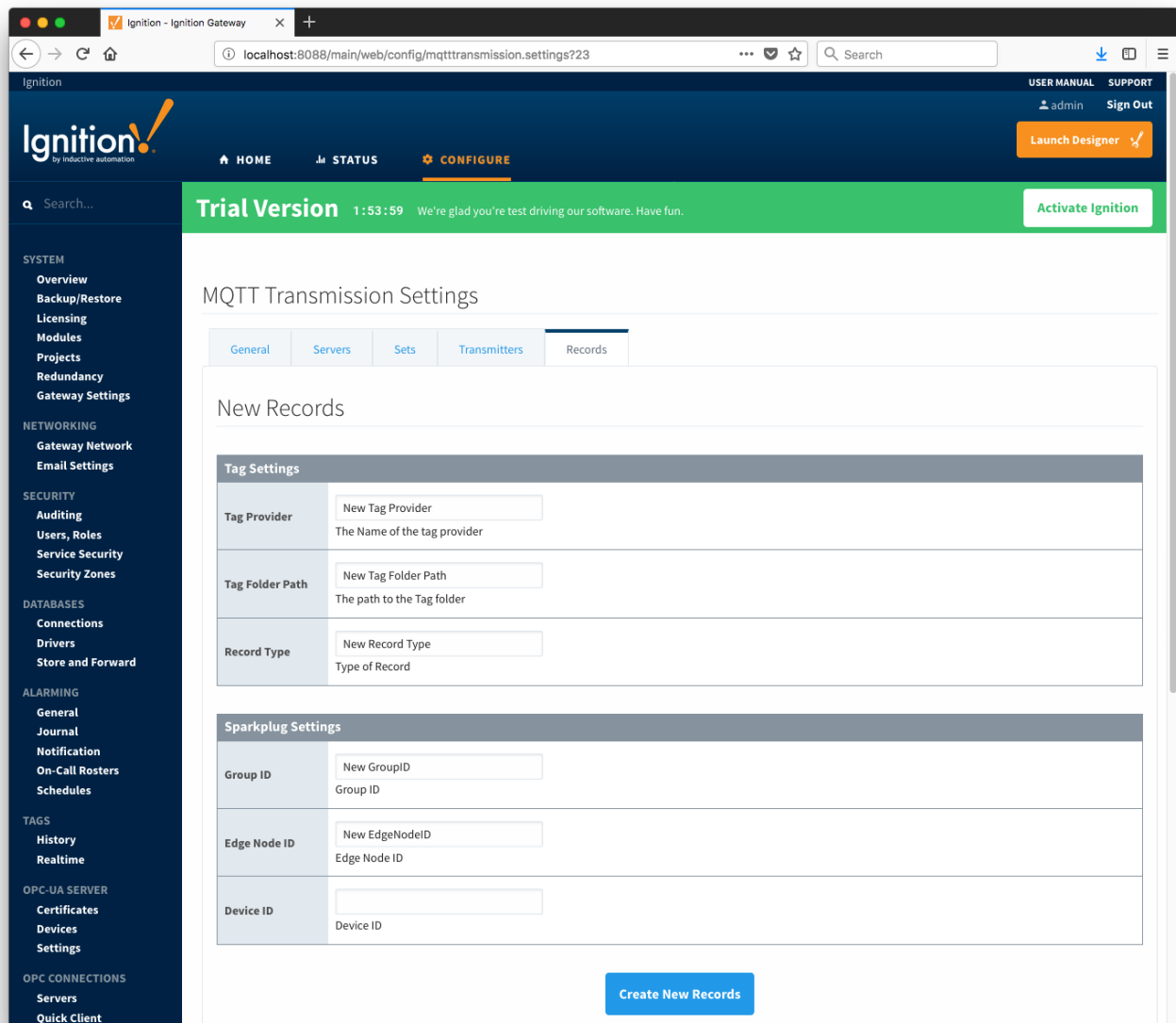
Tag Settings	
Transmitter Name	<input type="text"/> A unique name for the Custom Transmitter
Tag Provider	<input type="text"/> The Name of the tag provider
Tag Path	<input type="text"/> A path to the root folder where the tag tree starts (optional)
Tag Pacing Period	<input type="text" value="1000"/> The waiting period in milliseconds after an initial tag change event before publishing all changed tags (default: 1,000)
Set	<input type="text" value="Default"/> The MQTT Server Set to use with this Transmitter
Discovery Delay	<input type="text" value="0"/> The Transmitter Discovery Delay in milliseconds. This is useful when using MQTT Engine as the tag provider (default: 0)
History Store	<input type="text" value="- none -"/>

## Records

The 'Records' tab allows user to create a table of custom records. Each record defines a folder (under specified tag provider) in which user can create a number of tags that will be published as a record by clicking the 'Publish' checkbox. The 'Publish' tag will be created automatically in the folder specified.



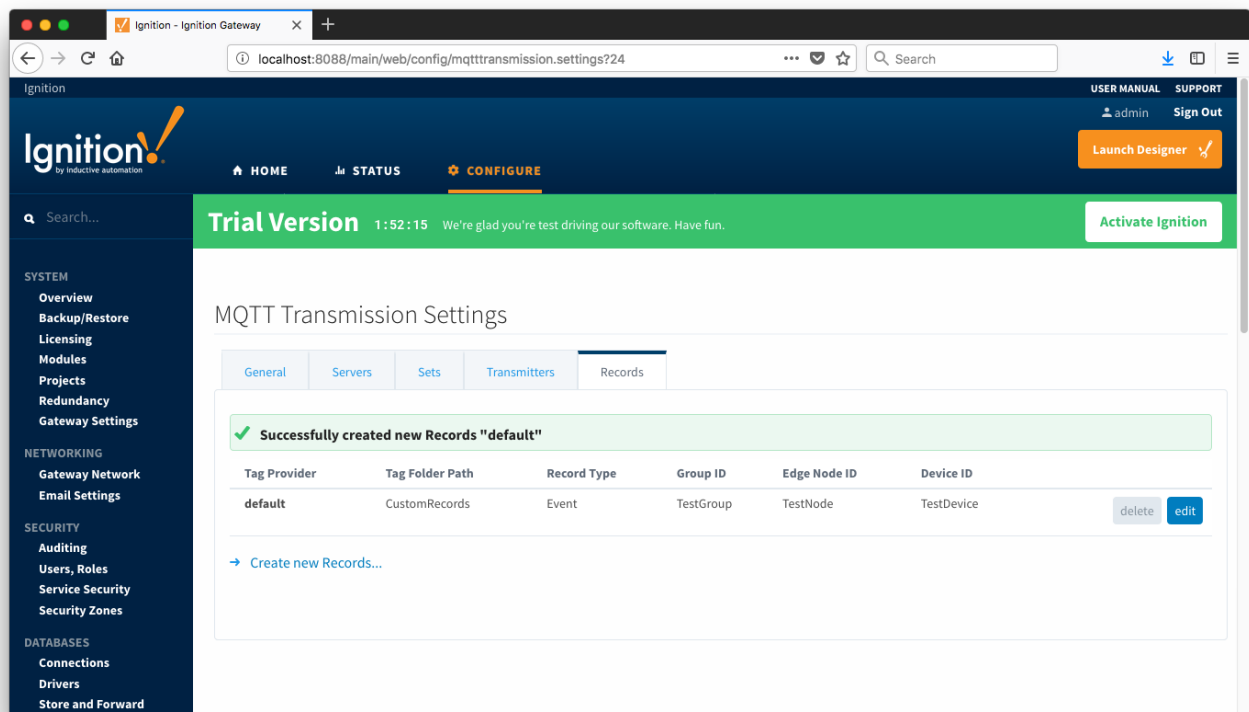
Clicking on the 'Create new Records ...' link will bring up the following form to add a new Record:



Each custom record has the following fields under two categories: 'Tag Settings', and 'Sparkplug Settings'

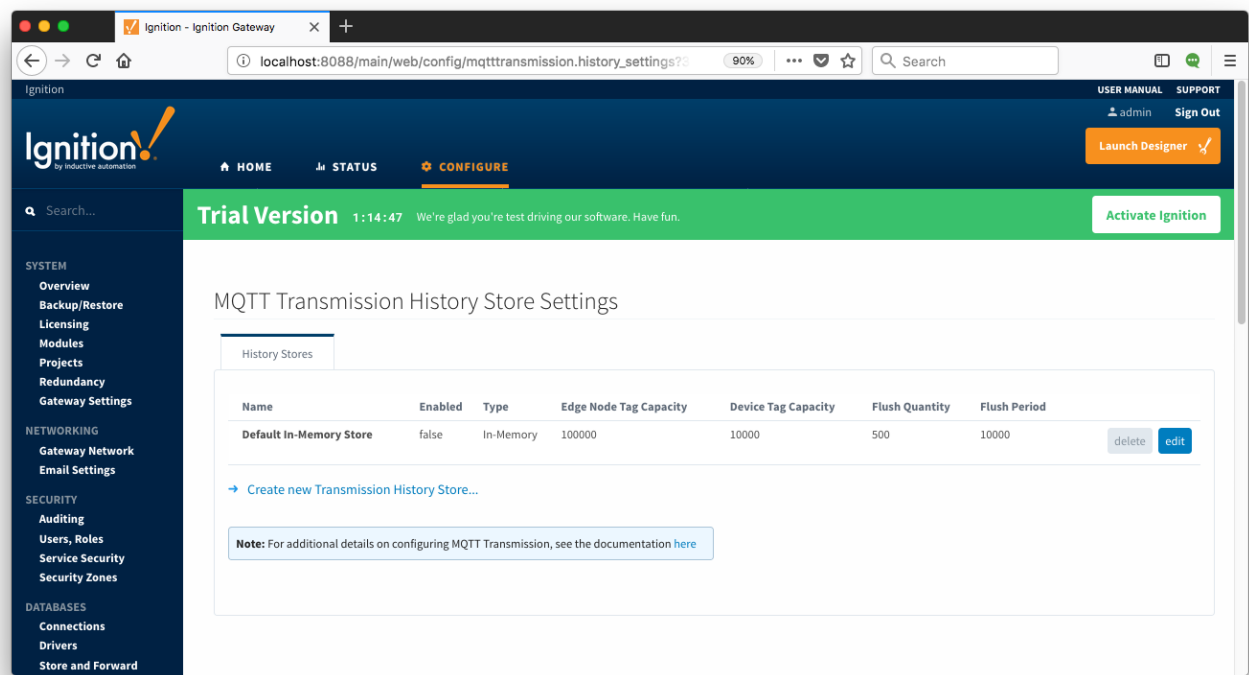
- Tag Settings
  - Tag Provider
    - The name of the tag provider (i.e. default)
  - Tag Folder Path
    - The path to the tag folder under specified tag provider
  - Record Type
    - Type of Record
- Sparkplug Settings
  - Group ID
  - Edge Node ID
  - Device ID

Below is an example of a submitted record:



## History

The "History" page allows for the configuration of MQTT Transmission History Stores. In the event that a Transmitter loses its connection with the MQTT Server and is unable to reconnect, a History Store (if enabled) will store all messages corresponding to change events on the monitored tags. Once a connection with an MQTT server is reestablished the History Store will publish the stored messages with a flag set to indicate that the messages are "historical" to prevent confusion with live data values.



Each History Store has the following fields

- **Name**
  - The name of the History Store.
- **Enabled**
  - Whether the History Store is enabled.
- **Type**
  - The type of History Store.
- **Edge Node Tag Capacity**
  - The maximum number of edge node tags to store (per edge node) before dropping the oldest currently stored tag value.
- **Device Tag Capacity**
  - The maximum number of device tags to store (per device) before dropping the oldest currently stored tag value.
- **Flush Quantity**
  - The maximum number of tags tags to publish in a single message upon reestablishing communication.
- **Flush Period**
  - The period to wait in milliseconds between publishes when flushing messages upon reestablishing communication.

The screenshot shows the Ignition Gateway configuration interface in a web browser. The browser address bar shows the URL: `localhost:8088/main/web/config/mqtttransmission.history_settings?2`. The page title is "MQTT Transmission History Store Settings". The interface includes a sidebar with navigation links for SYSTEM, NETWORKING, SECURITY, DATABASES, ALARMING, TAGS, OPC-UA SERVER, OPC CONNECTIONS, MOBILE, and ENTERPRISE ADMINISTRATION. The main content area is titled "MQTT Transmission History Store Settings" and contains a "History Stores" tab. Below the tab is a "New Transmission History Store" form. The form has a "Main" section with the following fields:

Main	
Name	<input type="text" value="New History Store"/> Name of this History Store
Enabled	<input checked="" type="checkbox"/> Enable this History Store
Type	<input type="text" value="Choose One"/> The Type of this History Store
Edge Node Tag Capacity	<input type="text"/> Maximum number of edge node tags to store (per edge node) before dropping oldest historical edge node tags
Device Tag Capacity	<input type="text"/> Maximum number of device tags to store (per device) before dropping oldest historical device tags
Flush Quantity	<input type="text"/> Upon reestablishing communication, the maximum number of tags tags to publish at a time
Flush Period	<input type="text"/> Upon reestablishing communication, the period to wait in milliseconds between publishes in flushing messages

Below the form is a blue button labeled "Create New Transmission History Store". At the bottom, there is a note: "Note: For additional details on configuring MQTT Transmission, see the documentation [here](#)".