

# RUT361 Data to Server

[Main Page](#) > [RUT Routers](#) > [RUT361](#) > [RUT361 Manual](#) > [RUT361 WebUI](#) > [RUT361 Services section](#) > **RUT361 Data to Server**

The information in this page is updated in accordance with firmware version [RUT361\\_R\\_00.07.08](#).

□

## Contents

- [1 Summary](#)
- [2 Data Senders](#)
  - [2.1 Sender Settings](#)
    - [2.1.1 General](#)
    - [2.1.2 Collection settings](#)
    - [2.1.3 Server configuration](#)

## Summary

The **Data to Server** feature provides you with the possibility to set up data senders that collect data from various sources and periodically send it to remote servers.

**Note:** On RUT361, Data to Server is additional software that can be installed from the **System** → [Package Manager](#) page.

If you're having trouble finding this page or some of the parameters described here on your device's WebUI, you should **turn on "Advanced WebUI" mode**. You can do that by clicking the "Advanced" button, located at the top of the WebUI.

✘

## Data Senders

A **Data Sender** is an instance that gathers and periodically sends collected data to a specified server. The Data Senders list is empty by default so, in order to begin configuration you must add a new data sender first. To add a new data sender, click the 'Add' button.

✘

After this you should be redirected to the newly added data sender's configuration page.

## Sender Settings

---

Refer to the table below for descriptions on data sender configuration fields.

**Note:** these tables have coloring schemes to indicate which fields can be seen with different configuration.



## General

Field	Value	Description
Name	string; default: <b>none</b>	Name of the data sender. Used for easier data senders management purposes only (optional).
Type	Base   GSM   Lua script   <b>Mobile usage</b>   MNF info   <b>Modbus*</b>   Modbus Alarms   <b>Wifi scanner*</b>   <b>DNP3*</b>   <b>MQTT</b>   <b>OPC UA</b>   <b>DLMS*</b> ; default: <b>Base</b>	Source of the data to be sent to server.
Format type	Json   <b>Custom</b> ; default: <b>Json</b>	Arranges the format of the sent JSON segment.
<b>Format string</b>	string; default: <b>none</b>	Specifies custom format string.
<b>Empty value</b>	string; default: <b>N/A</b>	A string which will be placed if any value cannot be received.
<b>Delimiter</b>	string (Maximum length of value is 1 bytes); default: <b>N/A</b>	Specifies delimiters for multiple data segments.
Segment count	integer [1..64]; default: <b>1</b>	Max segment count in one JSON string sent to server.
Send as object	off   on; default: <b>off</b>	When turned on, sends JSON segment as object and not as an array element.
Data filtering	All   Server ID   Alarm ID   Register number; default: <b>All</b>	If Data type: <b>Modbus alarms data</b> . Choose which data this sender will send to server.
<b>Data filtering</b>	All   <b>Server IP address</b>   <b>Server ID</b>   <b>Request name</b> ; default: <b>All</b>	If Data type: <b>Modbus data</b> . Choose which data this sender will send to server.
<b>Data filtering</b>	All   <b>Name</b>   <b>MAC address</b>   <b>Signal strength</b> ; default: <b>All</b>	If Data type: <b>Wifi scanner</b> . Choose which data this sender will send to server.
<b>Data period</b>	<b>Day</b>   <b>Week</b>   <b>Month</b> ; default: <b>Day</b>	If Data type: <b>Mobile usage</b> . Choose for which time period to send info from.
<b>Current</b>	off   on; default: <b>off</b>	If Data type: <b>Mobile usage</b> .
<b>Data filtering</b>	All   <b>Name</b> ; default: <b>All</b>	If Data type: <b>DLMS</b> . Choose which data this sender will send to server.
<b>Invert file</b>	off   on; default: <b>off</b>	If Data type: <b>DLMS</b> . Inverts filter condition.
<b>Data filtering</b>	All   <b>Address</b>   <b>IP</b> ; default: <b>All</b>	If Data type: <b>DNP3</b> . Choose which data this sender will send to server.
<b>Database</b>	RAM   Flash; default: <b>RAM</b>	Database location
<b>Data filtering</b>	All   <b>Name</b> ; default: <b>All</b>	If Data type: <b>OPC UA</b> . Choose which data this sender will send to server.

Server address	Default: <b>empty</b>	Hostname or ip address of the broker to connect to.
Port	integer [0..65535]; default: <b>1883</b>	Port number for connecting to MQTT.
Keepalive	integer [1..640]; default: <b>60</b>	MQTT Keepalive period in seconds.
Topic	string; default: <b>none</b>	MQTT topic to be used for publishing the data.
Client ID	string; default: <b>none</b>	Client ID to send with the data. If empty, a random client ID will be generated
QoS	integer [0..2]; default: <b>0</b>	MQTT Quality of Service. Allowed values: <ul style="list-style-type: none"> <li>• 0 - when we prefer that the message will not arrive at all rather than arrive twice,</li> <li>• 1 - when we want the message to arrive at least once but don't care if it arrives twice (or more),</li> <li>• 2 - when we want the message to arrive exactly once. A higher QoS value means a slower transfer.</li> </ul>
Enable secure connection	off   <b>on</b> ; default: <b>off</b>	Enables the use of TLS certificates.
On: TLS type	Certificate based   Pre-shared key based; default: <b>Certificate based</b>	Select type of TLS.
Certificate based: Allow insecure connection	off   <b>on</b> ; default: <b>off</b>	Allow not verifying server authentication.
Certificate based: Certificate files from device	off   <b>on</b> ; default: <b>off</b>	Specify where the certificates will be used from.
Certificate based: CA File	.ca file; default: <b>none</b>	<b>Certificate authority</b> is an entity that issues digital certificates. A digital certificate certifies the ownership of a public key by the named subject of the certificate. Certificate file is a type of digital certificate that is used by client systems to make authenticated requests to a remote server. If client certificate is not needed, leave both client certificate and client key fields empty.
Certificate based: Client certificate	.crt file; default: <b>none</b>	File containing private key for this client. This file needs to be not encrypted.
Certificate based: Client private Key	.key file; default: <b>none</b>	The pre-shared-key in hex format with no leading "0x".
Pre-shared key based: Pre-Shared-Key	string; default: <b>none</b>	The identity of this client. May be used as the username depending on the server settings.
Pre-shared key based: Identity	string; default: <b>none</b>	Username used in authentication.
Username	string; default: <b>none</b>	Enables password for authentication.
Require password	on   off; default: <b>off</b>	Password used in authentication.
Password	string; default: <b>none</b>	

\* This is additional software that can be installed from the **System** → [Package Manager](#) page.

## Collection settings

---

Field	Value	Description
Enabled	off   on; default: <b>on</b>	Enables data to server collection instance.
Format type	JSON   <b>custom</b> ; default: <b>JSON</b>	Data collection objects formatting.
<b>Format string</b>	Default: <b>Instance name</b>	Specifies custom format string
<b>Empty value</b>	Default: <b>N/A</b>	A string which will be placed if any value cannot be received
Period	Default: <b>60</b>	Interval in seconds for collecting/sending data to destination.
Retry	off   <b>on</b> ; default: <b>off</b>	In case of a failed attempt, retry to send the same data to destination later.
<b>Retry count</b>	Default: <b>10</b>	Retry to send the same data N times
<b>Timeout</b>	Default: <b>1</b>	Timeout in second between retry attempts

## Server configuration

---

Field	Value	Description
Type	HTTP   <b>MQTT</b> ; default: <b>HTTP</b>	Interval in seconds for collecting/sending data to destination.
Server address	Default: <b>empty</b>	Hostname or IP address of the broker to connect to.
HTTP headers	Default: <b>empty</b>	Allows to add custom headers to the HTTP requests.
Enable secure connection	on   off; default: <b>off</b>	Enables the use of TLS certificates.
<b>Port</b>	integer [0..65535]; default: <b>1883</b>	Port number for connecting to MQTT.
<b>Keepalive</b>	integer [1..640]; default: <b>60</b>	MQTT Keepalive period in seconds.
<b>Topic</b>	string; default: <b>none</b>	MQTT topic to be used for publishing the data.
<b>Client ID</b>	string; default: <b>none</b>	Client ID to send with the data. If empty, a random client ID will be generated
<b>QoS</b>	integer [0..2]; default: <b>0</b>	MQTT Quality of Service. Allowed values: <ul style="list-style-type: none"><li>• 0 - when we prefer that the message will not arrive at all rather than arrive twice,</li><li>• 1 - when we want the message to arrive at least once but don't care if it arrives twice (or more),</li><li>• 2 - when we want the message to arrive exactly once.</li></ul> A higher QoS value means a slower transfer.
Enable secure connection	off   <b>on</b> ; default: <b>off</b>	Enables the use of TLS certificates.

Allow insecure connection	off   on; default: <b>off</b>	Allow not verifying server authentication.
Certificate files from device	off   on; default: <b>off</b>	Specify where the certificates will be used from.
Certificate authority file	.ca file; default: <b>none</b>	<b>Certificate authority</b> is an entity that issues digital certificates. A digital certificate certifies the ownership of a public key by the named subject of the certificate. Certificate file is a type of digital certificate that is used by client systems to make authenticated requests to a remote server. If client certificate is not needed, leave both client certificate and client key fields empty.
Client Certificate	.crt file; default: <b>none</b>	
Client Private Keyfile	.key file; default: <b>none</b>	File containing private key for this client. This file needs to be not encrypted.
Use credentials	off   <b>on</b> ; default: <b>off</b>	Enables use of username and password for authentication.
<b>On:</b> Username	string; default: <b>none</b>	Username used in authentication.
<b>On:</b> Password	string; default: <b>none</b>	Password used in authentication.