https://wiki.teltonika-gps.com/view/Help_with_Server_FAQ

Help with Server FAQ

This page is dedicated to quickly find most commonly faced issues or questions when trying to create an IoT data platform which supports Teltonika Telematics devices.

☐

Contents

Pre-Requisites for server implementation

What are the pre-requisites for deploying devices on my server?

To develop the platform, below are the requirements:

Requirement	**Comment**
Hardware	A high CPU count promotes better cloud tiering throughout because although object storage can be slow in I/O operations per thread, object storage can support many threads. Any standard x86 64-bit servers can be ideally used.
Memory Requirements	Cloud services demand a large amount of memory, which is why the minimum recommended memory size is 8GB.
Hard disk	For a medium-sized (up to a 1000 devices) server, 300GB RAID1 disks can be recommended.
Database	A database is required to store the records/messages incoming from multiple devices. Further, this data must be assigned to its appropriate device (recognized by IMEI). MySQL can be used for the database.
Security	Teltonika devices support TLS Encryption which can be implemented. How to generate TLS certificates (Windows)?
Programming Skills	Knowledge about programming language is a must. Teltonika Telematics is a hardware production company, we do not provide software programming services and cannot help with writing code for the server. Only consult on the logic and algorithms of device features and possible issues in data interpretation from the server side.
Hardware Knowledge	It can be found on our Teltonika WIKI: https://wiki.teltonika-gps.com/view/Main_Page To understand the devices, their use cases and how they send data.
Protocol Implementation	Teltonika Telematics device protocols need to be implemented for the server to understand and parse the received data correctly. Teltonika Data Sending Protocols
Socket Programming	Sockets and the socket API are used to send messages across a network. For Teltonika devices to make connection to your server, the server needs to have an application socket programmed to accept these connection request. **Note:** Each socket should be dedicated to one device, this will allow the server to differentiate the devices when more than 1 device is sending data at the same time.
Network Considerations	Cloud storage supports storing cold data on a public cloud object storage service,such as Amazon S3. When using a public cloud, the connection is over a WAN.
Bandwidth Requirements	Bandwidth speed requirements are based on the amount of data transfer per month. It can be calculated using some online tools such as: https://www.calculator.net/bandwidth-calculator.html.
IP Address Requirements	Static IP or a dedicated port and domain is required otherwise device configuration would need to be updated constantly with a new IP or a new Port.

How to open ports in my computer for virtual testing?

The step by step instructions to open a TCP/UDP port can be found here:
https://wiki.teltonika-gps.com/view/Opening_TCP/UDP_port

What ports should i use to keep connection with the server?

Any Non-reserved ports on the server side, which are not being blocked and used by server's firewall and services respectively.

Documentation

Does Teltonika offer any homologation or server implementation documents which we can use to see how the data is sent, received, and parsed?

Regarding the documents/sources, here's what we offer:
1. The wiki link on data sending protocols:
https://wiki.teltonika-gps.com/view/Teltonika_Data_Sending_Protocols
2. The parsing toolkit containing the TCP/UDP Listener, source code and other related documents. It can be downloaded from here:
https://wiki.teltonika-gps.com/view/Universal_Device_Test_Guide#Protocols_implementation

What are the supported Network Protocols for Teltonika devices which I need to implement on my server

Currently, the Teltonika devices works with 03 different protocols for Data Sending; TCP, UDP, and MQTT. Please keep a note that MQTT is supported only via AWS server or a custom server which should be implemented based on the AWS protocols. More information on MQTT (based on AWS) can be found here: https://wiki.teltonika-gps.com/view/Getting_Started_with_AWS_IoT_Core.

Is there any ID/Value available corresponding to the paramters that Teltonika devices offer?

Yes, there are several parameters that you can get from our devices and the AVL IDs corresponding to each one of them can be found here:
https://wiki.teltonika-gps.com/view/FMM130_Teltonika_Data_Sending_Parameters_ID.

Communication with server

How does device communicate with the server?

First, when module connects to server, module sends its IMEI. First comes short identifying number of bytes written and then goes IMEI as text (bytes).
For example, IMEI 356307042441013 would be sent as 000F333536333037303432343431303133.
First two bytes denote IMEI length. In this case 0x000F means, that IMEI is 15 bytes long.
After receiving IMEI, server should determine if it would accept data from this module. If yes, server will reply to module 01, if not - 00. Note that confirmation should be sent as binary packet. I.e. 1 byte 0x01 or 0x00. Then module starts to send first AVL data packet. After server receives packet and parses it, server must report to module number of data received as integer (four bytes). If sent data number and reported by server doesn't match module resends sent data.

How to setup records sorting from protocols Codec?

Records can be sorted by using timestamp as a record sorting method, as it allows us to know what record was received early.

How to setup the answer to the Codec protocols in C#, Java, C++ from the server side, while counting records?

You can write script in your preferred language (C#, JAVA or Python) which can extract the number of records received from device and send it as a ACK packet to the device.

Troubleshooting Data

How to know if the device is sending the data to server or not? And how to know if the server is accepting the data from device or not?

1. We can always check the Device Status->GSM info page on the Teltonika Configurator. If device' GPRS status is "Activated" and Sent Records count is above zero and the Sent Records count is increasing that means device is sending data to the server. Thereafter, the server has to accept the data and send back response.
2. If the server has the Codec 8 protocol integerated properly, it will send the response in HEX as number of records received by server. On Teltonika Configurator under Status->GSM info, we can check the "Received Data", Socket and Last server Response time to confirm when was the last connection happened with server and its response hour.

How to make multiple session separation from the server side from different devices?

Multiple devices can be separated by recognizing the IMEIs of the devices on the server side. The very first step by the device is to send the IMEI on the server and server receives IMEI and sends ACK packet back to the module. All the modules have different IMEI which can be used as a point of differenciation.

How can we know if the acknowledgment is required on the server or not? Can we change this setting?

Acknowledgment is made depending upon the "Record Settings":
1. If ACK Type, TCP/IP is selected then Server Acknowledgement is not required
2. If ACK Type, AVL is selected then the Server must respond with the Acknowledgement.

Common Mistakes

I am getting the error on the server side saying"Protocol Mismatch". What could possible be wrong in this case?

This is because of mismatch in the selection of Network Protocol on the device and server side. It is possible that on the Device, TCP is selected and on the server side its UDP or vice-versa. Since the structure for both the protocols is different, this mismatch can restrict establishing a connection on the server.

Communication over GPRS

How can my device communicate over GPRS messages?

We have few CODEC protocols supporting GPRS communication: Codec 12,13,14,15. However the functionality basics remains the same and is explained below for Codec 12.
1. First, the Teltonika device opens the GPRS session and sends AVL data to the server (refer to device protocols).
2. Once all records are sent and the correct sent data array acknowledgment is received by the device then GPRS commands in Hex can be sent to the device.
3. The ACK (acknowledgment of IMEI from server) is a one-byte constant 0x01. The acknowledgment of each data array sent from the device is four bytes integer â€“ a number of received records.
Detailed information on the protocol can be found here:
https://wiki.teltonika-gps.com/view/Teltonika_Data_Sending_Protocols#Codec_12

RAW Data Example

I need to parse the data received on my end. Do you have any piece of code or examples that you can provide that I can use to develop my parser?

Here's an example of the Data Received in the HEX Format and how to parse it:
Received data in the hexadecimal stream:
000000000000003608010000016B40D8EA30010000000000000000000000000000000105021503010101425E0F01F10000601A014E0000000000000000010000C7CF.
The detailed parsing can be found in the wiki page, under the category Communication with server:
https://wiki.teltonika-gps.com/view/Teltonika_Data_Sending_Protocols#Codec_8