UHF **Desktop app** Interfacing with a **Server**

Contents

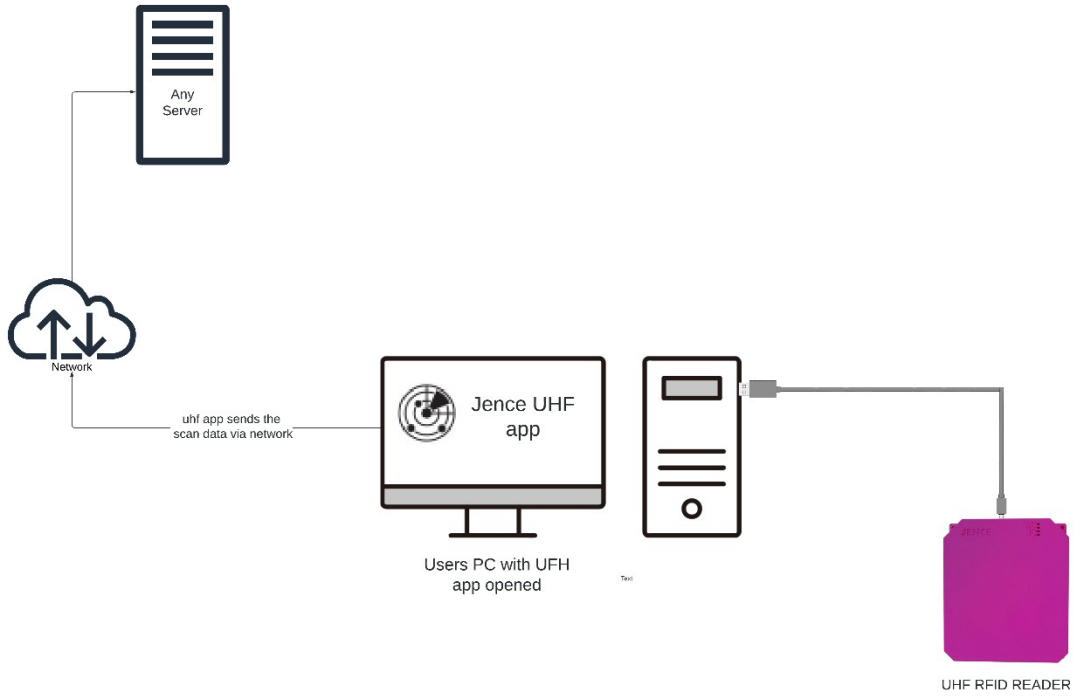
Getting Started

The UHF Multiplatform Desktop software of **Jence UHF RFID** reader has options for communication. It can be found on the messaging tab. A developer can use their own backend server to manage their database/Software without needing any custom solution. The desktop reader sends a JSON payload to specified network location once a scan occurs.

The communication has 2 methods,

1. Raw Socket with json.
2. http or https server.

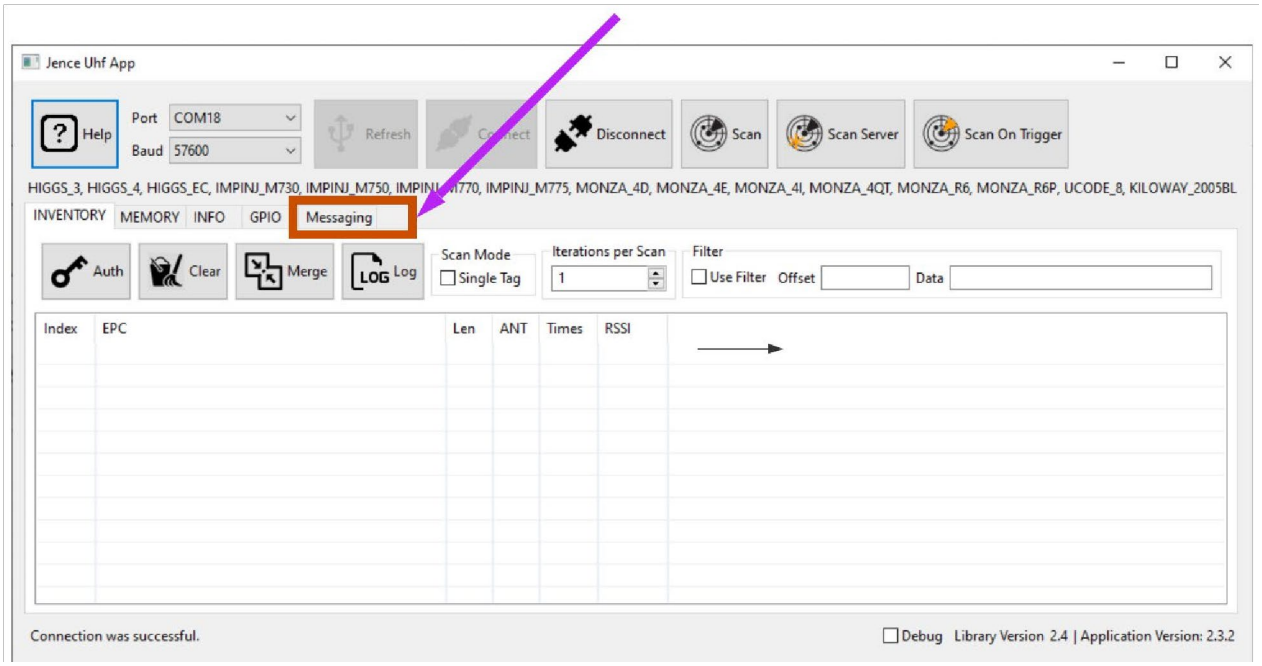
The basic idea of the setup is to take the data from RFID reader via desktop software then send it to any sort of server in a JSON format. The diagram below clarifies the concept further



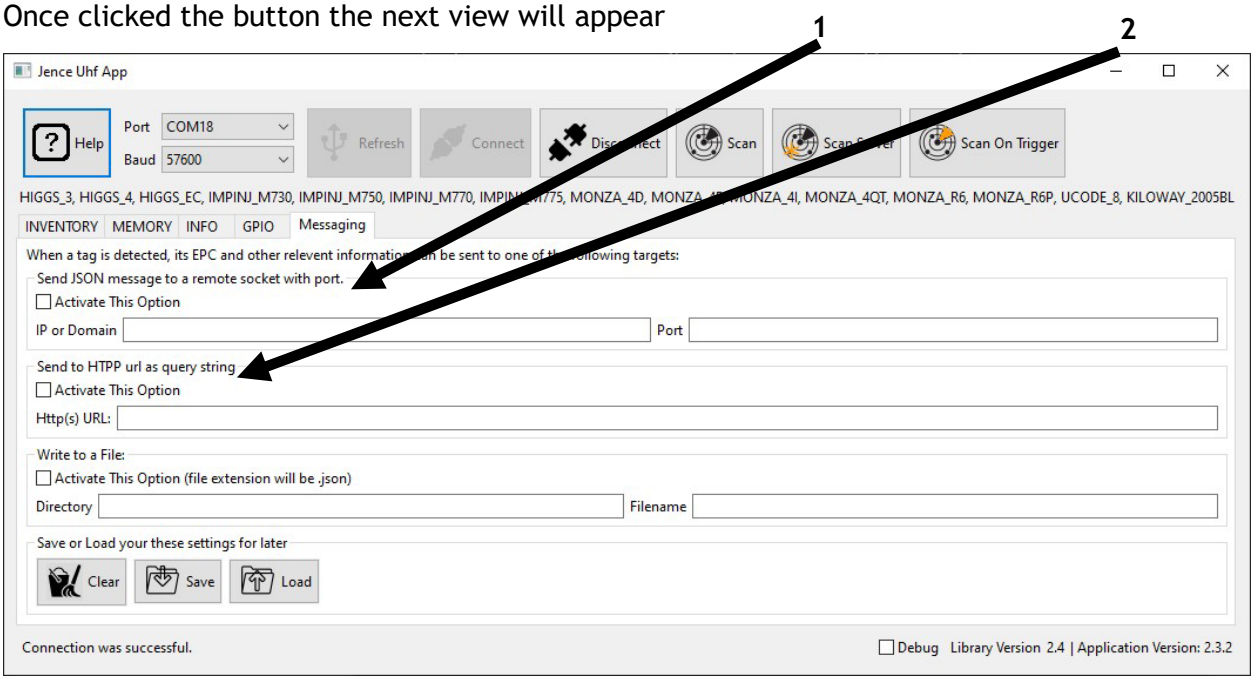
Software Familiarization:

Jence multiplatform **UHF app** has many features that is crucial to use the **UHF Reader**. And, this guide will focus on the messaging aspect of the software.

The messenger tab can be found from below picture



Once clicked the button the next view will appear



In this view 2 network options are visible. One option sends a JSON string through the given **IP** and **PORT**. Another option sends a http or https request in the given link with and the JSON data will be in the parameter of the request.

NOTE: To open the Messaging tab the connection with a **RFID reader** is mandatory. Without connection the tab won't appear.

Format of the Sent Data:

The Scanned data that will be send through the messaging option will be in a JSON format and the format is demonstrated below,

```
{
  Ant: 1,
  RSSI: -116,
  Count: 2,
  EPCLength: 12,
  EPC: '3110AFEC2B0BEBC201000000',
  Timestamp: '2024-03-04T15:57:57.5063634+06:00'
}
```

This format will be for each tag. So, for each tag the app will send a json payload or http request.

The parameters are self-explanatory, they are

Ant	The antenna that was used for the Scan
RSSI	Signal strength of the Scanned UHF Tag
Count	The Number of time tag responded
EPCLength	Length of the EPC
EPC	The EPC value of the tag
Timestamp	The time of scan

Example NodeJS server Setup:

Any server with json parsing ability will work perfectly. This example will use a NodeJS server but the idea is same for every language server.

Step-1: Install the NodeJS from official website or from your package manager

Step-3: open in the codes in a code editor to read it or you can just simply open a terminal in the **examples** directory.

Step-4:

Run the command

```
npm install
```

It will install the required dependencies.

Step-5:

There are 3 Examples one uses ExpressJS to create a http & https server. Another is normal http server. Last one is a socket server. You can run each of those by entering,

```
node http-server.js
```

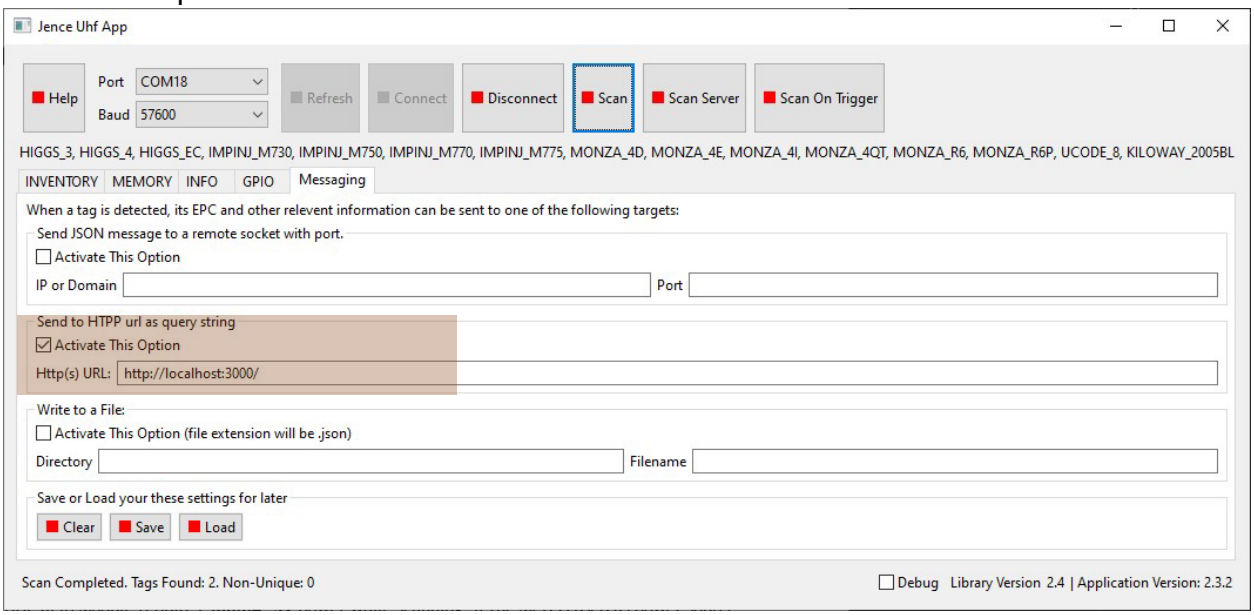
Or other examples. By running other **js** files by changing the file name in the place of **httpserver.js.**

in the terminal. It will start listening to port 3000. And you can use the uhf desktop app to send message to that server with IP and PORT.

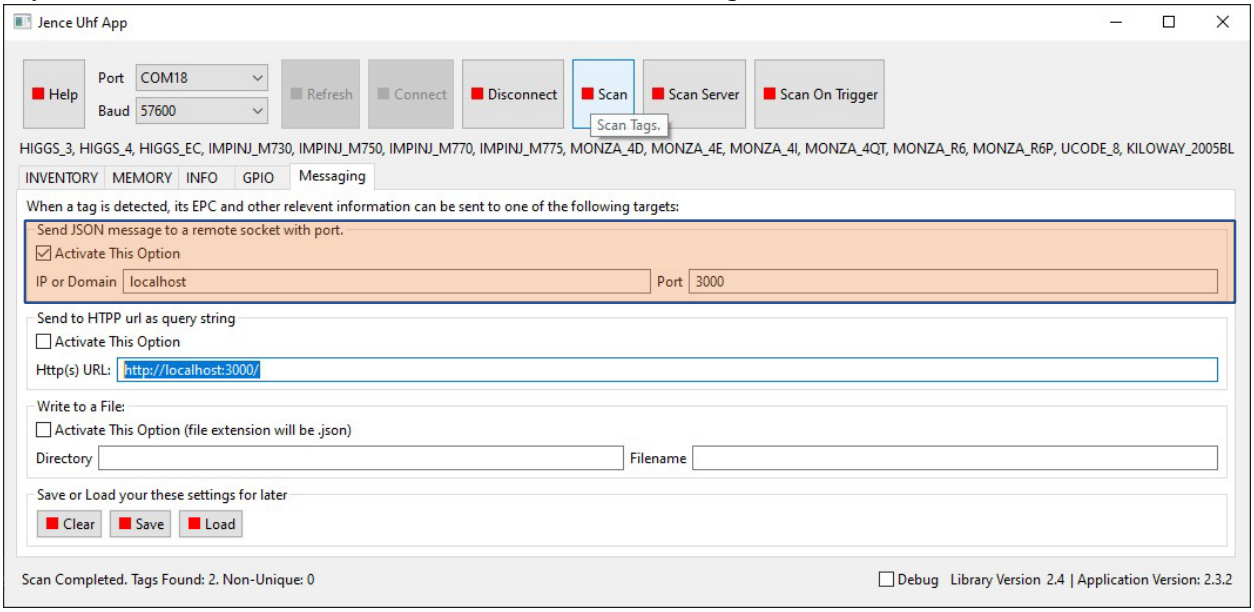
Example usage

1. Connect the RFID reader with PC via the **USB cable**.
2. Download the software if you haven't already downloaded it. The software can be found on github repository of jence . You can download the latest release from the link or you can just clone the repository.
3. Refresh and connect to the appeared COM port of the RFID reader.
4. Once connected keep a tag nearby and press Scan button if the tag shows up then you the connection was successful. Similar check can be done by going to info tab and it check the same without scanning. Info tag will give the settings of the reader.
5. If the connection wasn't successful, you can press the help button on top left for more information. Furthermore, the installation guide can also be found in the **doc** folder of the repository.
6. Once the connection between reader and pc ha been established, we can start the messaging process.
7. From previous step node js server is up and running. If not then follow the steps to create a server.
8. If you have run the http server example of the node js server then check mark the http option from the messaging section and enter the server **URL/IP** with corresponding **PORT**.
   Then every scan will send the json data to the server.
9. If you have selected socket example then a socket server will be opened and you have to choose the 1st option of sending data in right IP and Port.

10. For our example servers the URL is **localhost** and **PORT** is **3000**



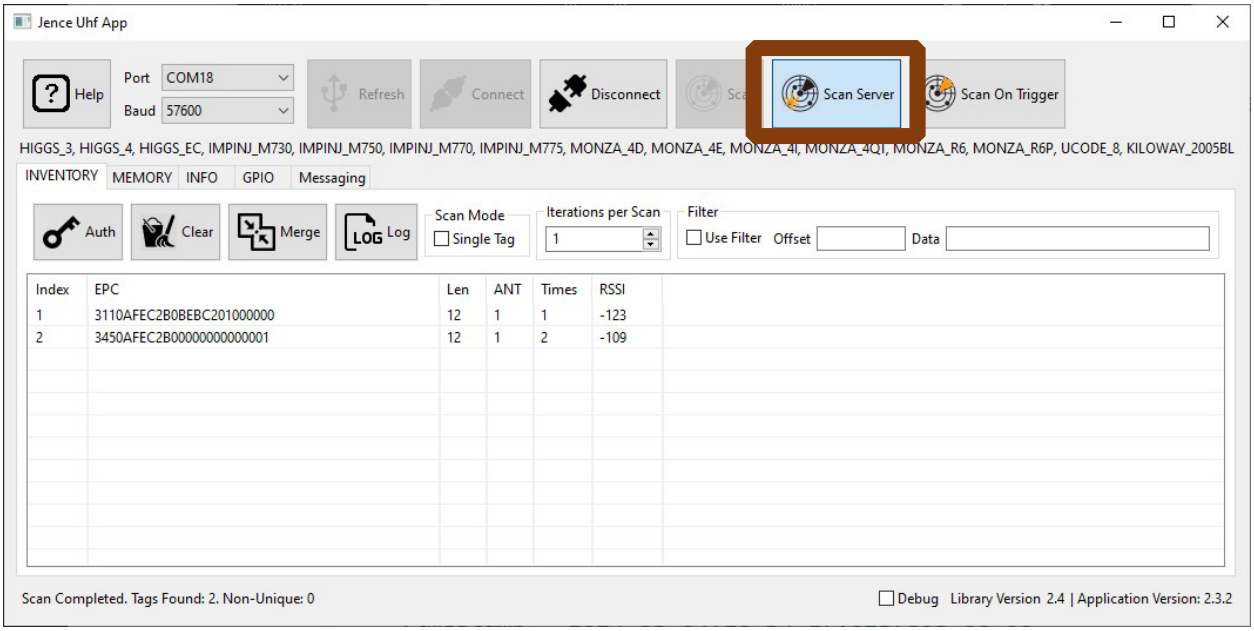
11. If you have selected the Socket server then a the configuration will look like below



12. After each scan the EPC value will be send to the server. After each scan the node js example servers console will look like the below picture

```
Client connected
Received JSON: {
  Ant: 1,
  RSSI: -118,
  Count: 1,
  EPCLength: 12,
  EPC: '3110AFEC2B0BEBC201000000',
  Timestamp: '2024-03-04T16:54:17.6039093+06:00'
}
Received JSON: {
  Ant: 1,
  RSSI: -73,
  Count: 1,
  EPCLength: 12,
  EPC: '3450AFEC2B00000000000001',
  Timestamp: '2024-03-04T16:54:17.6237893+06:00'
}
Client disconnected
```

13. After turning on the correct messaging configuration. You can press Scan Server to continuously scan and send the data to your server.



14. You can use the demo anyhow you may like with a database or with any website. Depending on your need.

FAQ

1. Can I do the server and RFID readers connection standalone without the Desktop app?

Ans: Yes. It can be done using the Shared library of the reader driver. The documentations and examples can be found in the repository. In the repository see the python example from **platform** > **Python** folder. And API guide from **doc** folder.