

Meshtastic Web Interface and Remote Access

Many Meshtastic users don't realize a full web-based interface is available in addition to the mobile app. The web interface provides additional tools for configuration and administration that aren't available in the app.

The Meshtastic web app (client.meshtastic.org)

The official Meshtastic web application is hosted at client.meshtastic.org and runs in any modern browser (Chrome or Edge recommended for WebSerial support), connecting to your device via USB serial or Bluetooth.

Connecting via USB serial

1. Connect your Meshtastic device via USB cable
2. Open client.meshtastic.org in Chrome or Edge
3. Click "Connect" and select "Serial"
4. Choose your device from the port list
5. The device configuration loads automatically

Note: WebSerial requires Chrome or Edge. Firefox and Safari do not support WebSerial. If the device doesn't appear in the port list, install the appropriate driver (CH340 for Heltec, standard CDC for nRF52-based devices).

Connecting via Bluetooth

1. Open client.meshtastic.org in a Chromium-based browser
2. Click "Connect" and select "Bluetooth"

3. Pair with your device - the device should be in BLE advertising mode (typically shown by LED blink pattern)

What the web app offers beyond the mobile app

Feature	Mobile app	Web app
Node configuration	Yes	Yes (more detailed)
Channel management	Yes	Yes
Message view	Yes	Yes
Node map	Yes	Yes
Configuration export/import (YAML)	Channel/config sharing via URL/QR; recent apps also support config backup	Yes
Raw packet log	Limited	Full packet log
Serial debug console	No	Yes
Firmware update (OTA)	Yes (BLE OTA on supported devices)	Browser flashing via USB (Web Flasher); verify current per-platform support
Full module config access	Partial	Yes

Configuration backup and restore

The web app's configuration export feature is an easy way to back up and restore a Meshtastic node's settings (the exported file is YAML; the CLI uses the same format):

1. Connect to device via web app
2. Navigate to **Config → Export Config**
3. Save the YAML file to your computer

To restore: connect the device (after a reflash or factory reset), navigate to **Config → Import Config**, and upload the saved YAML file. This restores all radio settings, channel configurations, and module settings in one step. The equivalent CLI workflow is `meshtastic --export-config > config.yaml` to back up and `meshtastic --configure config.yaml` to restore (there is no `--import-config` flag).

Remote node management via WiFi

ESP32-based Meshtastic nodes (Heltec V3/V4, T-Beam, etc.) can be connected to a local WiFi network so they can be reached over the network:

```
meshtastic --set network.wifi_ssid "YourNetworkSSID"
meshtastic --set network.wifi_psk "YourPassword"
meshtastic --set network.wifi_enabled true
```

Once connected to WiFi, the device's local IP address is shown in the app or via `meshtastic --info`. That IP is used by the Meshtastic web client and API (TCP port 4403) rather than a self-hosted admin webpage; connect to it from client.meshtastic.org (choose the network/TCP connection option) to configure and monitor the node without USB or Bluetooth.

Use case: A deployed rooftop repeater with WiFi access can be configured and monitored remotely from any device on the same network, eliminating the need for physical access for routine configuration changes.

Security warning: The node's network interface has no built-in authentication. Anyone who can reach the node's IP on the network can connect to it, reconfigure it, change channels, or read its configuration (including channel PSKs via export). Only expose a node on a trusted, isolated network - never on open or shared WiFi. For untrusted environments, enable managed mode (`security.is_managed`) and use PKC admin keys for remote administration.

Serial debug console

The web app's serial console shows raw debug output from the device firmware. This is invaluable for diagnosing unusual behavior:

- View all received packets with raw headers and signal data
- See routing decisions in real time
- Monitor power-related messages (battery voltage, charge state)
- Debug GPS acquisition

The serial console is accessed via **Tools → Serial Console** in the web app. Output can be copied and shared when reporting bugs.