

# How to become a feeder?

- [\[PL\] Zostań feederem ADS-B.Pro](#)
- [\[EN\] Become ADS-B.Pro Feeder!](#)

# [PL] Zosta? feederem ADS-B.Pro

## Konfiguracja ADS-B.Pro feeder

### 1. Co b?dzie potrzebne

- Raspberry Pi / Debian / Ubuntu z dost?pem do internetu.
- Odbiornik RTL-SDR lub inny odbiornik ADS-B.
- Dzia?aj?cy `dump1090-fa`, `dump1090-mutability`, `dump1090` albo `readsb`.
- Konto w ADS-B.Pro.
- Tw?j prywatny feeder token z konta u?ytkownika.

Token feeder jest unikalny. W panelu konta jest pokazywany jako **Your Feeder Token** i ma format:

```
ADS-xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

czyli `ADS-` plus 32 znaki hex. Skrypt wymaga ma?ych liter `a-f`.

### 2. Sk?d wzi?? token

1. Zaloguj si? w ADS-B.Pro.
2. Wejdź w **My Account**.
3. Skopiuj warto? z pola **Your Feeder Token**.
4. Nie publikuj tego tokena w logach, na GitHubie ani w screenach.

Frontend u?ywa tego tokena tylko jako warto?ci przypisanej do konta. Sam feeder wysyła go do backendu przez osobny skrypt `/opt/radarview.py`.

### 3. Najprostsza instalacja feedera

Rekomenduj? wariant `radarview_setup-nogit.sh`, bo nie wymaga klonowania repozytorium i sam pobiera aktualny `radarview.py`.

```
sudo apt update
sudo apt install -y curl wget ca-certificates

curl -fsSL
https://raw.githubusercontent.com/br3jski/radarview/refs/heads/main/radarview_setup-nogit.sh -
o radarview_setup-nogit.sh
chmod +x radarview_setup-nogit.sh

sudo ./radarview_setup-nogit.sh
```

Po uruchomieniu skrypt poprosi o token:

```
Please enter your RadarView token (format: ADS-XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX):
```

Wklej token z **My Account**.

## 4. Co robi skrypt

Skrypt:

1. Sprawdza, czy jest uruchomiony jako `root`.
2. Waliduje token użytkownika.
3. Sprawdza, czy istnieje lokalne źródło ADS-B:
  - `dump1090-fa`
  - `dump1090-mutability`
  - `dump1090`
  - `readsb`
4. Jeśli nie ma `dump1090/readsb`, pyta, czy zainstalować `dump1090-fa`.
5. Pobiera `radarview.py` do:

```
/opt/radarview.py
```

6. Wstawia token do zmiennej:

```
USER_TOKEN = 'ADS-...'
```

7. Tworzy usługę systemd:

```
/etc/systemd/system/radarview.service
```

8. Uruchamia i włącza usługę na start systemu:

```
systemctl start radarview  
systemctl enable radarview
```

## 5. Jak działa wysyłanie danych

`radarview.py` łączy się lokalnie z odbiornikiem ADS-B na porcie SBS:

```
127.0.0.1:30003
```

Następnie wysyła dane do ADS-B.Pro:

```
feed.ads-b.pro:48581
```

Przed każdą linią danych skrypt wysyła token:

```
TOKEN:ADS-...
```

Dlatego kluczowe są dwie rzeczy:

- lokalny `dump1090` / `readsb` musi wystawiać dane SBS na `127.0.0.1:30003`,
- urządzenie musi mieć możliwość połączenia wychodzącego TCP do `feed.ads-b.pro:48581`.

## 6. Sprawdzenie po instalacji

Status usługi feedera:

```
sudo systemctl status radarview
```

Logi na żywo:

```
sudo journalctl -u radarview -f
```

Poprawne logi powinny zawierać coś w tym stylu:

```
Connected to source 127.0.0.1:30003  
Connected to destination feed.ads-b.pro:48581  
Data sent with token
```

Sprawdzenie, czy lokalny ADS-B działa na porcie `30003`:

```
nc -vz 127.0.0.1 30003
```

Sprawdzenie połączenia do serwera ADS-B.Pro:

```
nc -vz feed.ads-b.pro 48581
```

## 7. Jeśli używasz readsb zamiast dump1090-fa

Upewnij się, że `readsb` ma włączony port SBS `30003`. Feeder z podanego skryptu nie czyta JSON-a ani Beast, tylko tekstowy strumień SBS/BaseStation z portu `30003`.

Przykładowo konfiguracja `readsb` powinna zawierać odpowiednik:

```
--net  
--net-sbs-port 30003
```

Po zmianach zrestartuj `readsb`.

## 8. Zmiana tokena po instalacji

Jeśli token został odnowiony albo wklejony błędnie, najprościej uruchomić skrypt instalacyjny ponownie.

Można też ręcznie poprawić `/opt/radarview.py`, znaleźć linię:

```
USER_TOKEN = 'ADS-...'
```

i po zmianie zrestartować usługę:

```
sudo systemctl restart radarview
```

## 9. Typowe problemy

### Invalid token format

Token nie pasuje do formatu `ADS-` plus 32 znaki hex. Skopiuj token dokładnie z **My Account**.

### **Error: User token not set**

W `/opt/radarview.py` token jest pusty. Uruchom setup ponownie albo popraw `USER_TOKEN`.

### **Socket error przy `127.0.0.1:30003`**

`dump1090` / `readsb` nie działa albo nie wystawia SBS na porcie `30003`.

### **Socket error przy `feed.ads-b.pro:48581`**

Problem z DNS, internetem, firewallem albo blokadą połączeń wychodzących TCP na port `48581`.

### **Konto nadal nie ma statusu Feeder**

Feeder musi faktycznie wysyłać stabilne dane. Po stronie frontendlu status `Feeder` pojawia się z konta użytkownika/backendu; czasem trzeba się wylogować i zalogować ponownie albo odświeżyć token sesji po zatwierdzeniu feedera.

# [EN] Become ADS-B.Pro Feeder!

## ADS-B.Pro Feeder Setup Guide

### 1. Requirements

You will need:

- Raspberry Pi / Debian / Ubuntu with internet access.
- RTL-SDR or another ADS-B receiver.
- Working `dump1090-fa`, `dump1090-mutability`, `dump1090`, or `readsb`.
- ADS-B.Pro account.
- Your private feeder token.

The feeder token is unique per account. In the account panel it is shown as **Your Feeder Token** and has this format:

```
ADS-xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

That means `ADS-` plus 32 hexadecimal characters. The setup script expects lowercase `a-f`.

### 2. Where To Find Your Token

1. Log in to ADS-B.Pro.
2. Open **My Account**.
3. Copy the value from **Your Feeder Token**.
4. Do not publish this token in logs, screenshots, GitHub issues, or config examples.

The frontend displays this token from your account data. The actual feeder sends it to the backend through `/opt/radarview.py`.

### 3. Recommended Installation

Use `radarview_setup-nogit.sh`. It does not require cloning the repository and downloads the current `radarview.py` automatically.

```
sudo apt update
sudo apt install -y curl wget ca-certificates

curl -fsSL
https://raw.githubusercontent.com/br3jski/radarview/refs/heads/main/radarview_setup-nogit.sh -
o radarview_setup-nogit.sh
chmod +x radarview_setup-nogit.sh

sudo ./radarview_setup-nogit.sh
```

The script will ask for your token:

```
Please enter your RadarView token (format: ADS-XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX):
```

Paste the token from **My Account**.

## 4. What The Script Does

The script:

1. Checks that it is running as `root`.
2. Validates the user token.
3. Checks whether a local ADS-B source exists:
  - `dump1090-fa`
  - `dump1090-mutability`
  - `dump1090`
  - `readsb`
4. If no `dump1090/readsb` installation is found, it asks whether to install `dump1090-fa`.
5. Downloads `radarview.py` to:

```
/opt/radarview.py
```

6. Inserts your token into:

```
USER_TOKEN = 'ADS-...'
```

7. Creates a systemd service:

```
/etc/systemd/system/radarview.service
```

8. Starts and enables the service:

```
systemctl start radarview  
systemctl enable radarview
```

## 5. How Data Forwarding Works

`radarview.py` connects to your local ADS-B receiver on the SBS/BaseStation port:

```
127.0.0.1:30003
```

Then it forwards the data to ADS-B.Pro:

```
feed.ads-b.pro:48581
```

Before every aircraft data line, the script sends your token:

```
TOKEN:ADS-...
```

So two things are required:

- local `dump1090` / `readsb` must expose SBS data on `127.0.0.1:30003`,
- the feeder machine must be able to make an outbound TCP connection to `feed.ads-b.pro:48581`.

## 6. Verify The Installation

Check service status:

```
sudo systemctl status radarview
```

Watch live logs:

```
sudo journalctl -u radarview -f
```

Healthy logs should look similar to:

```
Connected to source 127.0.0.1:30003  
Connected to destination feed.ads-b.pro:48581  
Data sent with token
```

Check local ADS-B source:

```
nc -vz 127.0.0.1 30003
```

Check ADS-B.Pro connectivity:

```
nc -vz feed.ads-b.pro 48581
```

## 7. If You Use readsb

Make sure `readsb` exposes SBS/BaseStation output on port `30003`. The feeder script does not read JSON or Beast directly. It expects text SBS data from port `30003`.

Your `readsb` configuration should include the equivalent of:

```
--net  
--net-sbs-port 30003
```

Restart `readsb` after changing its configuration.

## 8. Changing The Token Later

If the token was pasted incorrectly or regenerated, the easiest option is to run the setup script again.

You can also edit `/opt/radarview.py` manually, update:

```
USER_TOKEN = 'ADS-...'
```

and restart the service:

```
sudo systemctl restart radarview
```

## 9. Common Problems

### Invalid token format

The token does not match `ADS-` plus 32 hex characters. Copy it again from **My Account**.

**Error: User token not set**

`USER_TOKEN` in `/opt/radarview.py` is empty. Run the setup again or edit the file manually.

**Socket error for `127.0.0.1:30003`**

`dump1090` / `readsb` is not running or SBS output is not enabled on port `30003`.

**Socket error for `feed.ads-b.pro:48581`**

Likely DNS, internet, firewall, or outbound TCP port `48581` connectivity issue.

**Account does not show Feeder status yet**

The feeder needs to send stable data first. The frontend account status comes from the backend account data, so after approval you may need to log out and back in, or refresh your session token.