Tesla Fleet API — Owner Setup Checklist

EVFleetPulse • Updated Mar 12, 2026 • A pragmatic checklist to get a working owner integration (OAuth, PEM hosting, endpoints, wake strategy, and cost-safe telemetry).

Use this for:	Personal owner automation (SoC checks, charge rules, preconditioning), troubleshooting app registrati
Avoid:	High-frequency polling of vehicle/device data endpoints, waking vehicles just to show fresh UI
Best practice:	Cache last-known state + stream telemetry in windows (charging/driving)

1) OAuth setup (Authorization Code flow)

- ■ Create Tesla developer app and choose Authorization Code (owner grants access once).
- ■ Set redirect URI EXACTLY (scheme + host + path). No “close enough”.
- ■ Keep client_secret server-side only (never in browser).
- ■ Store refresh token encrypted at rest (KMS/envelope).
- ■ Implement refresh token rotation: always store/use the NEW refresh token returned.
- ■ Add a per-user single-flight lock to prevent concurrent refresh invalidation.

2) Token exchange + refresh (minimum validation)

Fast sanity checks:

- ■ Authorization URL loads and returns code to your redirect endpoint.
- ■ Token exchange returns access_token + refresh_token.
- ■ API call to /api/1/vehicles succeeds with Bearer access token.
- ■ Refresh works once, and a second refresh works using the NEW refresh token.

3) Regional base URL

- ■ Confirm you are using the correct regional Fleet API base URL for your account/vehicles (NA vs EU vs APAC).
- ■ Keep base URL configurable via environment variables so you can switch during debugging.

4) Public key / well-known PEM hosting (registration pitfall)

Most registration failures are one of: wrong path, redirects, or HTML instead of PEM.

- ■ Serve the public key at Tesla’s required .well-known path (exact path).
- ■ Return 200 OK with NO redirects (avoid 301/308).
- ■ Response is raw PEM (starts with -----BEGIN PUBLIC KEY-----). Not HTML, not your SPA index.html.
- ■ No auth required to fetch the PEM (public).
- ■ Verify externally with curl -I and curl | head from a clean network.

Verification commands:

curl -I https://YOUR_DOMAIN/.well-known/.../com.tesla.3p.public-key.pem

curl https://YOUR_DOMAIN/.well-known/.../com.tesla.3p.public-key.pem | head

5) Wake strategy (don’t kill your cost + battery)

- Do NOT wake vehicles just to display fresh data; show cached values with “last updated”.
- Wake only when executing a scheduled or user-requested command (and only if needed).
- Use backoff and stop after a few attempts (avoid wake loops).

6) Telemetry vs polling (cost-safe architecture)

- If you need ongoing signals, use Fleet Telemetry rather than polling.
- Stream in windows: during charging and (optionally) driving sessions; stop when inactive.
- Store deltas/aggregates rather than every raw sample forever.
- Alert on deltas with hysteresis + rate limits (avoid noisy alerts).

7) Fast troubleshooting

If something fails, check these in order:

- 401/403: token expired, refresh rotation bug, wrong scopes, wrong region base URL.
- Registration: redirect URI mismatch, PEM path wrong, redirect present, PEM returns HTML.
- Telemetry: server not reachable publicly, firmware prereqs, key pairing/proxy signing.
- Commands failing: vehicle offline/asleep, signature/key missing, retry/backoff needed.

If you don’t want to build all this yourself

EVFleetPulse is building the owner-focused layer on top of Fleet API: charge rules (weekday/weekend), smart preconditioning, and alerts. Use this checklist as a reference and grab the beta when you want the outcome without maintaining the plumbing.