Build Plan — Device Authorization Flow (V5 Parity)

Version: 5.0.0 • Date: 2025-10-01

Goal: Implement OAuth Device Authorization Grant using the V5 shell, styling, components, and logging. Reuse shared V5 pieces (layout, buttons, forms, status bars, toasts, logging bus) and only add what's unique to device code.

0) Concepts & Flow

1. Request device_code and user_code from device authorization endpoint.
2. Device displays user_code + verification URI (or QR).
3. User authorizes on secondary device.
4. Device polls token endpoint until success or error.

1) Config & Environment

- OIDC_ISSUER, OIDC_CLIENT_ID, OIDC_SCOPES
- DEVICE_POLL_INTERVAL_MS, DEVICE_POLL_MAX_SECONDS, DEVICE_GRANT_ENABLED

2) Routes

- /flows/device
- /flows/device/instructions
- /flows/device/poll
- /flows/device/summary

3) UI/UX (V5 Parity)

- Same layout, nav, status bar, tokens UI as Authorize V5.
- QR code optional if verification_uri_complete available.

4) Logging

Format: [YYYY-MM-DD HH:mm:ss] [■ OAUTH-DEVICE] [LEVEL] message

5) State Model

Store deviceCode, userCode, verificationUri, intervalMs, expiresAt, polling status, tokens.

6) Services

requestDeviceCode() → POST to /device_authorization

pollForToken() → POST to /token with device_code

7) Hooks & Screens

- useDeviceAuth() → state machine, logs, polling
- DeviceStart, DeviceInstructions, DevicePoll, DeviceSummary

8) Error UX

authorization_pending → continue

slow_down → backoff

access_denied → stop

expired_token → stop

9) Security

- Never log secrets
- Mask tokens in logs
- Enforce expiry
- Validate ID token

10) Tests

- Unit, integration, E2E with polling behavior, errors, success.

11) Acceptance Criteria

- Pixel parity with V5
- Tokens masked
- Unified logging
- Full test suite green

12) Cursor Tasks

1. Scaffold routes & config
2. Service layer
3. Hook/state machine
4. Screens
5. Security & validation
6. Tests
7. Docs

13) Nice-to-Haves

- “Open verification URL” button
- QR code rendering
- Progress countdown
- “Re-run” button