

Supabase Environment Safety Interview Script

Hook (0:00-0:10)

"I own the Supabase environment resolution layer, which keeps preview deployments safe while preventing production outages."

Config Logic (0:10-0:35)

"In `src/integrations/supabase/config.ts` we read `import.meta.env` directly, normalize boolean flags, and detect preview environments via Vercel hostnames or an explicit `VITE_SUPABASE_FORCE_PREVIEW`. If a build ends up on `preview-*.vercel.app`, we automatically switch to preview URLs; otherwise we stick to the standard cluster."

Fail-fast & Fallbacks (0:35-0:55)

"We treat production as sacred: missing URLs or publishable keys throw immediately so we never ship a broken prod build. In development we log a warning and fall back to local test credentials to keep DX smooth."

Client & Storage (0:55-1:15)

"`client.ts` consumes that config and instantiates `createClient<Database>` with browser-safe storage. `localStorage` is only touched when `window` exists, which keeps SSR layouts from exploding. We also centralize type definitions in `src/integrations/supabase/types.ts` so the client is fully typed end-to-end."

Security Tests (1:15-1:30)

"We locked this down with vitest suites: `client.environment.test.ts` stubs `import.meta.env` and `window.location` to validate the preview routing, and `environment.security.test.ts` enforces that only vetted `VITE_` variables ever reach the browser."

Wrap-up (1:30-1:35)

"Net result: deployments pick the right Supabase cluster automatically, and we catch misconfigurations before users do."