

CertifyQuiz — Migrazione Vite → Next.js

Stato e prossimi passi • 27 September 2025

Contesto

- Frontend: Next.js 15 (App Router) su Vercel, repo: certifyquiz-next.
- Backend: Express/MySQL su Railway.
- Obiettivo: migrare il frontend da Vite a Next, con proxy interno e ISR.

Cose fatte (■)

- Proxy interno `**/api/backend/[...path]**` con inoltro header/cookie e compatibilità Next 15 (`await su context.params`).
- Pagine di debug: `**/debug/api-test**` e `**/debug/usa-api**` per validare il proxy (200 OK).
- Pagina dinamica `**/certificazioni/[slug]**` con `**ISR (revalidate=86400)**` e `**generateStaticParams/Metadata**`; URL verificato: ``/certificazioni/comptia-itf-plus``.
- Dati server-side in ``src/lib/data.ts``: uso di `**API_BASE_URL**` con `**type guards**` (niente `any`) e fallback mock.
- Endpoint Next per `**on-demand revalidation**`: ``src/app/api/revalidate/route.ts``; variabile `**REVALIDATE_SECRET**` configurata su Vercel.
- Test revalidation: `POST → `{ revalidated: true }`` (OK).
- Backend: helper ``utils/revalidate.js`` + endpoint admin `**POST /api/admin/revalidate-cert/:slug**` (e `by-topic`); env `**REVALIDATE_SECRET**` e `**NEXT_REVALIDATE_URL**` su Railway.
- Verifica endpoint backend: `**/api/ping-availability**` → ``{ ok: true, where: 'quizRoutes' }``; revalidate admin → ``{ ok: true, slug: 'comptia-itf-plus' }``.
- ESLint: flat config attiva; file aggiornati per evitare ``no-explicit-any``.

Da fare (■)

- UI & Layout: ``Header`/`Footer`` in ``src/components`` e montaggio in ``src/app/layout.tsx``.
- SEO: ``src/app/sitemap.ts`` e ``src/app/robots.ts``; impostare ``NEXT_PUBLIC_SITE_URL``.
- Pagina elenco `**/certificazioni**` e `**Home**` con link alle singole certificazioni.
- Auth: pagina `**/debug/login-test**` per provare ``POST /api/backend/auth/...`` e cookie `first■party`; poi schermate reali.
- Redirect 301 (se servono): ``/it/certificazioni/:slug`` → ``/certificazioni/:slug`` in ``next.config.js``.
- Pulizia: rimuovere pagine di debug e allineare regole ESLint definitive.
- Backend: aggiungere ``/health`` semplice e collegare ``revalidateCert(slug)`` nei flussi di UPDATE/PUBLISH (automatico, non solo admin).
- Cutover: puntare dominio prod su Vercel, aggiornare ``NEXT_REVALIDATE_URL`` a ``https://www.certifyquiz.com/api/revalidate``, smoke test finale.
- QA: test funzionali, link, SEO (metadata/OG), performance e error pages (404/500).

Variabili d'ambiente (prod)

- Vercel (frontend): ``API_BASE_URL``, ``REVALIDATE_SECRET``, ``NEXT_PUBLIC_SITE_URL``.
- Railway (backend): ``REVALIDATE_SECRET``, ``NEXT_REVALIDATE_URL`` → ``https://certifyquiz-next.vercel.app/api/revalidate`` (poi dominio prod).

Percorsi/commit rilevanti

- Frontend: `src/app/api/backend/[...path]/route.ts`, `src/app/certificazioni/[slug]/page.tsx`, `src/lib/data.ts`, `src/app/api/revalidate/route.ts`.
- Backend: `quiz_project/utils/revalidate.js`, `quiz_project/routes/publicRoutes.js` (o `quizRoutes.js`) per endpoint admin.

URL utili (deploy attuale)

- Pagina esempio: `/certificazioni/comptia-itf-plus`.
- Proxy test: `/debug/api-test`, `/debug/usa-api`.
- Revalidate (FE): `/api/revalidate` (POST, con secret).
- Revalidate (BE): `/api/admin/revalidate-cert/:slug`.