

Brief Technique – AEBL

05/10/2025

Brief Technique – AEBL (African Elite Basketball League)

Ce document (1–2 pages) synthétise les choix techniques, défis, optimisations et pistes d'amélioration du projet.

1. Contexte et objectifs

- Construire un site moderne pour une ligue de basketball avec Next.js (App Router), TypeScript strict et Tailwind.
- Priorités: UX/Design, qualité de code, perfs (SSG/ISR), SEO, animations, responsive.

2. Architecture générale

- Framework: `Next.js 15 (App Router)`.
- Langage: `TypeScript` (mode strict, `strict: true`).
- Styles: `Tailwind CSS` + composants `shadcn/ui`.
- Structure:
 - `app/` pages et layouts (SSR/SSG/ISR, metadata par page).
 - `lib/data/` données statiques typées (équipes, joueurs, matches).
 - `lib/` hooks/utilitaires (ex: notifications, recherche).
 - `components/` UI réutilisables.
 - `public/` assets (logos, photos, icônes).

3. Modèle de données (extrait)

- `Team`: { id, name, city, country, conference, wins, losses, logo }
- `Player`: { id, teamId, name, number, position, photo, stats: { ppg, rpg, apg } }
- `Match`: { id, date, time, venue, status, homeTeamId, awayTeamId, homeScore?, awayScore? }
- Relations: `Team 1—N Player`, `Team 1—N Match (home/away)`.

4. Stratégies de rendu

- Liste d'équipes et détails: SSG + `generateStaticParams` pour générer `/teams/[id]`.
- Données: lecture locale (mock réaliste). Possibilité d'exposer des routes API internes si nécessaire (ex: `/api/teams`).
- ISR (optionnel): paramètre `revalidate` pour certaines pages de liste afin de simuler un rafraîchissement.
- `generateMetadata`: métadonnées par page (title/description dynamiques).

5. UX/UI & animations

- Navigation responsive (header, sheet mobile).
- Mode sombre/clair via `next-themes`.
- Animations légères (classes utilitaires Tailwind + transitions).
- Accessibilité: libellés, `sr-only`, contrastes, tailles clicables.

6. SEO & Perf

- Images optimisées ``next/image`` (lazy, sizing) et ``images.unoptimized`` ajusté pour l'environnement statique.
- Balises méta par page, titres descriptifs (ex: ``Team Detail`` !' ``${team.name} - AEBL``).
- SSG par défaut pour temps de réponse réduit et meilleur Core Web Vitals.
- Tri/filtrage côté serveur lorsque pertinent pour limiter le JS côté client.

7. Qualité de code

- TS strict, typage des props (pages dynamiques ajustées à Next 15: ``params`` async).
- ESLint + organisation des composants UI.
- Découpage des responsabilités: données, hooks, composants, pages.

8. Défis rencontrés et solutions

- "Skipping validation" pendant le build: dû à ``ignoreDuringBuilds/ignoreBuildErrors`` dans ``next.config``; supprimés pour réactiver lint/TS.
- Avertissement racine Next (lockfiles multiples): défini ``outputFileTracingRoot`` pour ancrer le projet.
- Changement de typing Next 15 pour pages dynamiques: ``params`` peut être ``Promise``; pages mises à jour en ``async`` et ``await params``.
- Import NotificationToggle: collision entre ``lib/notifications.ts`` (hook) et ``ts`` (composant). Solution: composant isolé dans ``lib/notification-toggle.ts`` et import explicite.

9. Optimisations clés

- Pré-calcul SSG pour listes/détails !' TTFB et SEO.
- ``generateStaticParams`` pour pages ``/teams/[id]``.
- Découpage UI et réutilisation (cards, badges, boutons).
- Données typées pour éviter erreurs à l'exécution.

10. Pistes d'amélioration (futures)

- API interne (routes ``/api/...``) exposant ``lib/data/*`` pour démontrer SSR/ISR plus riche et filtres côté serveur.
- Data viz (classements/évolution de stats) via ``recharts``.
- Mode live simulé (polling/SSE) + notifications locales.
- Export avancé (PDF multi-pages avec sommaire, CSV du calendrier).
- Tests E2E (Playwright) et unitaires (Vitest/RTL).
- CI (GitHub Actions) et règles de protection (status checks) si requis par l'équipe.

11. Déploiement

- Cible: Vercel (Next.js). Build: ``npm run build``. Dossier sortie: ``.next``.
- SSG prêt à l'emploi; aucune dépendance externe obligatoire.

Ce brief illustre la cohérence des choix techniques avec les critères d'évaluation: design/UX, qualité du code, initiative, et performance.