

🏠 PRÉSENTATION

Bien démarrer

Tutoriel

Learning Path

[React](#)

[Base de données & Model de données](#)

[Intégrations](#)

[Tâches en arrière plan & tâches Cron](#)

[Tester](#)

[Architecture Logicielle](#)

[Compétences de communication](#)

[Conception de produits](#)

What is Next.js?

Why Blitz instead of Next.js?

Migrating to Blitz

Tradeoffs

Free Stickers

👥 COMMUNAUTÉ

Comment la Communauté opère

Manifesto

Historique

How to Contribute

Être un Mainteneur

Code de Conduite

Traduction des Documents

📁 BASES

File Structure

[<App>](#)

[<Image>](#)

[<Script>](#)

CSS

Font Optimization

Static Files

Environment Variables

Error Handling

[<ErrorBoundary>](#)

Blitz pas à pas 📖

Vous souhaitez perfectionner vos compétences de développement avec Blitz mais vous ne savez pas par où commencer ? Vous êtes arrivés sur la bonne page. Les liens évoqués sur cette page sont en anglais.

Inspiré par la **Frontend Developer Roadmap**, ce document aborde de multiples thèmes qui touchent aux concepts & idées, outils & bibliothèques et histoire & anecdotes afin de vous améliorer au développemnt applicatif. Cette page a été écrite pour l'usage de Blitz mais la plus part de ces enseignements s'appliquent universellement.

🕒 Info

Les liens marqué par sont destinés aux développeurs chevronnés. Si vous démarrez dans le monde du développement web, vous pouvez très bien les ignorer.

React

- [React for Beginners – A React.js Handbook for Front End Developers](#). Une grande introduction à React, y compris les composants fonctionels ainsi que les Hooks de React.
- [freeCodeCamp's Full React Course 2020](#). Une vidéo de 10 heures d'introduction à React.
- [The Beginner's Guide to React](#). Une Courte introduction à React par Kent C. Dodds, comprenant aussi les composants fonctionels.

Base de données & Model de données

- [Prisma's Data Guide: Data Modeling](#). Une introduction à la conception d'un bon schéma de base de données pour votre application.
- [What is Prisma?](#). Un panorama global de Prisma et ses fonctionalités.
- [Zero-Downtime Deployments with Data Migrations](#). Comment changer votre schéma de base de données sans arrêt de service.
- Vous pouvez contribuer à cette partie en ajoutant vos moyens d'apprentissage favoris via [Pull Request](#) ! Voici quelques idées de contenu qu'il nous manque encore :
 - un guide à propos de la création d'applications [multi-tenant](#)

Intégrations

- Vous pouvez contribuer à cette partie en ajoutant vos moyens d'apprentissage favoris via [Pull Request](#) ! Voici quelques idées de contenu qu'il nous manque encore :
 - comment envoyer des emails
 - comment intégrer Stripe
 - comment intégrer Paddle
 - quel fournisseur de paiement choisir

Tâches en arrière plan & tâches Cron

- Vous pouvez contribuer à cette partie en ajoutant vos moyens d'apprentissage favoris via [Pull Request](#) ! Voici quelques idées de contenu qu'il nous manque encore :
 - comment les configurations Cron marchent
 - cas d'usages des tâches en arrière plan
 - comment implémenter une tâche Cron

Tester

Les tests automatiques sont une bonne façon de vérifier que votre application marche ... et qui ne voudrait pas s'assurer que son application marche ?

- [Testing Javascript](#). Un guide détaillé sur tout ce qui est relatif aux tests par Kent C. Dodds.
- Vous pouvez contribuer à cette partie en ajoutant vos moyens d'apprentissage favoris via [Pull Request](#) ! Voici quelques idées de contenu qu'il nous manque encore :
 - une introduction à la pyramide de testing
 - comment implémenter des tests unitaires avec Jest
 - comment écrire du code facilement testable
 - un guide sur Cypress

Architecture Logicielle

Au fur et à mesure que votre application grandit et se complexifie, une structure peut devenir nécessaire. Il y a bien entendu un million de façon de structurer votre project, voici quelques guides pour bien démarrer:

- Vous pouvez contribuer à cette partie en ajoutant vos moyens d'apprentissage favoris via [Pull Request](#) ! Voici quelques idées de contenu qu'il nous manque encore :
 - [TODO](#): un bon guide sur l'onion et l'architecture claire

Compétences de communication

La création applicative est souvent un effort de group et requiert de la communication en équipe.

- [How to Make Your Code Reviewer Fall in Love with You](#). Les relectures de code sont un bon outil quand elles sont bien faites mais vraiment horribles lorsqu'elles ne le sont pas. Lisez cet article pour améliorer grandement vos PRs !
- Vous pouvez contribuer à cette partie en ajoutant vos moyens d'apprentissage favoris via [Pull Request](#) !

Conception de produits

La conception d'un produit est bien plus que la simple conception d'une application. Trouver le bon modèle de monétisation, évaluer les idées, recevoir des critiques constructives, construire un panel d'utilisateurs, tout ceci est très différent de la conception applicative. Si vous conceptualisez un produit (une bonne partie des utilisateurs Blitz sont entrepreneurs !), cette section est pour vous:

- [The Mom Test](#). Un livre sur l'honnêteté lors de critiques d'utilisateurs.
- Vous pouvez contribuer à cette partie en ajoutant vos moyens d'apprentissage favoris via [Pull Request](#) !

[Idea for improving this page? Edit it on GitHub.](#)

[← Tutoriel](#)

[What is Next.js? →](#)

Abonnez vous à notre newsletter pour vous tenir au courant des dernières nouveautés de Blitz !

[→](#)

Docs

- [Documentation](#)
- [Démarez avec Blitz](#)
- [Comment contribuer](#)

Community

- [Discord](#)
- [Discussions sur forum](#)
- [Twitter](#)
- [Vitrine](#)

Autre

- [Déploiement avec Flightcontrol](#)
- [GitHub](#)
- [Wiki](#)
- [Goodies](#)

