

# Guide d'étapes clés

Projet “Implémentez le front-end d'une application bancaire avec React”

## Comment utiliser ce document ?

Sur cette page, vous trouverez un exemple de découpage en étapes pour faire votre projet. Vous y trouverez :

- des recommandations pour réussir chaque étape ;
- les problèmes récurrents et points de vigilance à garder en tête ;
- une estimation de votre avancement sur l'ensemble du projet.

Ce découpage est simplement une suggestion pour vous organiser. Vous n'êtes pas obligé de compléter les étapes dans l'ordre.

**Gardez en tête que votre progression sur les étapes n'est qu'une estimation, et sera différente selon votre vitesse de progression.**

## Étape 1 : Initialisez le projet

### 5 % de progression



#### Avant de démarrer cette étape, je dois avoir :

- Suivi le cours [Débutez avec React](#).
- Suivi la partie 1 du cours [Créez une application React complète](#).
- Avoir pris connaissance de la demande relative au projet Kasa.

#### Une fois cette étape terminée, je devrais avoir :

- L'application React installée et le repo GitHub initialisé.
- Ajouté le fichier JSON qui contient les données.
- L'application de démonstration qui se lance dans le navigateur lorsque je tape ``npm start`` dans le terminal.

#### Recommandations :

Après vous être assuré d'avoir bien compris tous les éléments demandés pour le projet, initialisez votre projet avec Create React App.

**Ressources :**

- Découvrez [la documentation de Create React App \(CRA\)](#) pour des informations toujours actualisées sur comment initialiser votre application.

## Étape 2 : Ajoutez React Router

### 10 % de progression

---

**Avant de démarrer cette étape, je dois avoir :**

- Initialisé mon application.

**Une fois cette étape terminée, je devrais avoir :**

- React Router et ses dépendances installés sur mon application.

**Recommandations :**

Les bibliothèques JavaScript sont en constante évolution. Si vous avez des difficultés à installer React Router avec les instructions du cours, n'hésitez pas à vous tourner vers la documentation de React Router et à vérifier la version que vous installez.

**Ressources :**

- Lisez [la documentation de React Router](#) pour des informations à jour sur l'installation de la bibliothèque.

## Étape 3 : Créez le routeur et les différentes pages de l'application

### 25 % de progression

---

**Avant de démarrer cette étape, je dois avoir :**

- Installé React Router à l'application.

**Une fois cette étape terminée, je devrais avoir :**

- Le routage de mon application fonctionnel.

**Recommandations :**

- Lorsque vous créez la logique de routage, profitez-en pour créer les différentes pages de l'application. Posez-vous la question : Que se passe-t-il si j'écris mal l'URL d'une route ou si j'entre une route qui n'existe pas ?

- Pour la navigation dans l'application, vous pourrez soit faire un header et un footer, soit faire un layout dans lequel vous mettrez le reste de l'application via `props.children`. Pensez à créer ces composants et à y intégrer le composant du router.
- C'est le moment de commencer à faire la mise en page de votre application pour le layout principal. Vous avez pour cela différentes options (CSS, SASS, modules CSS).

**Points de vigilance :**

Créer un composant à part pour le routeur.

**Ressources :**

- Recommandations de CRA pour l'utilisation du CSS :
  - [Ajout de feuilles de style](#)
  - [Ajout de modules CSS \(recommandé par CRA\)](#)
  - [Ajout de SASS](#)

## Étape 4 : Créez la page d'accueil

### 40 % de progression

---

**Avant de démarrer cette étape, je dois avoir :**

- La logique de routage fonctionnelle.

**Une fois cette étape terminée, je devrais avoir :**

- Les composants banner et card créés.
- Une liste des propriétés disponibles.

**Recommandations :**

- Analysez bien les différents composants et voyez si ceux-ci sont réutilisés dans l'application. Évaluez quels éléments de l'interface nécessitent un composant qui leur est propre.

**Points de vigilance :**

- Attention à ce que la création des cards ne génère pas de warning dans la console.

**Ressources :**

- Voir la documentation sur les [listes et clés de React](#) pour voir comment gérer la répétition d'éléments sans warning.

## Étape 5 : Créez la page À propos

50 % de progression

---

### Avant de démarrer cette étape, je dois avoir :

- La logique de routage fonctionnelle.

### Une fois cette étape terminée, je devrais avoir :

- Le composant Collapse créé.

### Recommandations :

- Gardez en tête que le Collapse doit avoir deux états différents : il peut être ouvert ou fermé.
- Pensez à bien utiliser les propriétés des composants pour passer les données voulues.
- Souvenez-vous que votre composant peut prendre différentes formes en fonction de son utilisation.

### Ressources :

- Voir la documentation sur comment [gérer l'état dans un composant fonctionnel](#).

## Étape 6 : Mettez en page la page d'erreur

60 % de progression

---

### Avant de démarrer cette étape, je dois avoir :

- La logique de routage fonctionnelle.

### Une fois cette étape terminée, je devrais avoir :

- La page d'erreur fonctionnelle.

### Recommandations :

- Suivez bien le design de la maquette.
- Assurez-vous que toute route inexistante mène bien à la page d'erreur.

### Ressources :

- Lisez [la documentation de React Router sur la gestion des erreurs](#) pour intégrer une page d'erreur 404.

## Étape 7 : Créez le carrousel de la page de logement

75 % de progression

---

### Avant de démarrer cette étape, je dois avoir :

- La logique de routage fonctionnelle.
- Le clic sur les cartes de propriété de la page Home fonctionnel.

### Une fois cette étape terminée, je devrais avoir :

- Le composant Slideshow créé.

### Recommandations :

- Commencez par ajouter les flèches et la numérotation des slides de façon dynamique. Si une propriété n'a qu'une seule image, on ne devrait voir ni flèche ni numérotation.
- Ajoutez les event listeners pour qu'au clic le changement d'image se fasse correctement.

### Points de vigilance :

- Attention à ne pas avoir d'erreur lorsque vous vous trouvez aux extrémités du carrousel : le clic sur la flèche gauche à la première diapositive doit vous mener à la dernière image, et vice versa.

### Ressources :

- Voir cette documentation pour [gérer l'état dans un composant fonctionnel](#).

## Étape 8 : Finalisez la page de logement

85 % de progression

---

### Avant de démarrer cette étape, je dois avoir :

- Le carrousel fonctionnel sur la page.
- Le clic sur les propriétés dans la page d'accueil.

### Une fois cette étape terminée, je devrais avoir :

- La page de propriété fonctionnelle.

### Recommandations :

- Suivez bien le design de la maquette pour intégrer et mettre en page chaque élément.
- Pensez à réutiliser le composant Collapse créé auparavant.

## Étape 9 : Gérez l'erreur sur l'id d'une propriété

90 % de progression

---

### Avant de démarrer cette étape, je dois avoir :

- La page de propriété fonctionnelle.

### Une fois cette étape terminée, je devrais avoir :

- Une redirection vers la page d'erreur en cas d'id incorrect.

### Recommandations :

Utilisez les fonctionnalités de React Router pour gérer l'erreur et rediriger vers la page d'erreur.

### Ressources :

- Gérez la redirection avec le [composant Navigate](#) ou le [hook useNavigate](#).

## Étape 10 : Vérifiez l'ensemble de votre site

100 % de progression

---

### Avant de démarrer cette étape, je dois avoir :

- L'ensemble du projet réalisé.

### Une fois cette étape terminée, je devrais avoir :

- Le projet finalisé et vérifié.

### Recommandations :

- Faites-vous une checklist de tout ce qui était demandé sur le projet.
- Vérifiez que le style correspond à la maquette, y compris en responsive.
- Vérifiez que toutes les erreurs sont bien gérées, en particulier dans la navigation.
- Vérifiez que les composants changeant d'état fonctionnent de manière appropriée (changement des images dans le carrousel, et Collapses qui s'ouvrent et se ferment).

## Projet terminé !