

Étape 2 : Annuaire des employés

Objectif

Afficher la liste des employés avec leurs informations.

Concepts pratiqués

- Itération avec `map()`
 - Props obligatoire `key`
 - Rendu conditionnel
 - Passage de props entre composants
-

À créer

1. Composant `EmployeeCard.jsx`

Une carte affichant les informations d'un employé :

- Avatar
- Nom complet
- Rôle / Département
- Email et téléphone
- Indicateur de statut en ligne (point vert/gris)

2. Composant `EmployeeList.jsx`

La liste qui :

- Reçoit les données des employés
 - Affiche une `EmployeeCard` pour chaque employé
 - Gère le cas où la liste est vide
-

Données disponibles

```
// Importez depuis src/data/employees.js
import { employees } from '../../data/employees';

// Structure d'un employé :
{
  id: 1,
  firstName: "Victor",
  lastName: "Besson",
  email: "victor.besson@teamhub.com",
  department: "Développement",
  role: "Lead Developer",
  avatar: "https://i.pravatar.cc/150?img=1",
```

```
    phone: "01 23 45 67 89",
    isOnline: true,
    joinedAt: "2022-03-15"
}
```

Indices

- ▶💡 Itérer avec map() et key

```
{employees.map(employee => (
  <EmployeeCard
    key={employee.id} // ⚠️ OBLIGATOIRE et UNIQUE
    employee={employee}
  />
))}
```

Pourquoi key ?

- React utilise **key** pour identifier chaque élément
- Permet de savoir quels éléments ont changé
- Ne JAMAIS utiliser l'index comme key si la liste peut changer

- ▶💡 Rendu conditionnel pour le statut

```
// Méthode 1 : Opérateur ternaire

```

Points d'attention

- ⚠️ Erreur fréquente : Key manquante

```
Warning: Each child in a list should have a unique "key" prop.
```

- ⚠️ Solution : Ajouter **key={item.id}** sur l'élément racine du map.

- ⚠️ Erreur fréquente : Passer un objet entier vs déstructurer

```
// Approche 1 : Passer l'objet entier
<EmployeeCard employee={employee} />
// Dans EmployeeCard : function EmployeeCard({ employee }) { ... }

// Approche 2 : Destructurer (plus explicite)
<EmployeeCard
  name={employee.firstName}
  role={employee.role}
  // ...
/>
```

Les deux approches fonctionnent. L'approche 1 est plus simple pour beaucoup de props.

Critères de validation

- Toutes les cartes s'affichent correctement
- Pas de warning "key" dans la console
- Le statut en ligne est visible (indicateur vert/gris)
- Les informations sont bien formatées
- Design responsive