

Fiche Révision - Examen React + Zustand

PARTIE A : Gestion des propriétés (15 pts)

1) Récupération des propriétés depuis JSON Server :

- GET `http://localhost:3001/properties`
- `useEffect()` + `fetch()` + `useState`

2) Filtrage dynamique par prix :

- 2 champs input pour min / max
- `.filter()` en temps réel

3) Bouton "Réserver la propriété" :

- Visible uniquement si `available === true`

4) Lors du clic sur "Réserver" :

- PATCH pour `views +1`
- `navigate("/reserve", { state: { property } })`

5) Formulaire de réservation :

- Champs : nom, date début, date fin
- Calcul durée : `Math.ceil((end - start) / (1000 * 60 * 60 * 24))`
- Total = durée * price
- PATCH `{ available: false }`
- Message personnalisé affiché
- Redirection après 5s vers /

PARTIE B : Évaluations avec Zustand (5 pts)

1) Création du store Zustand (`useEvaluationStore.js`)

- Stocke les évaluations dans un objet `{ propertyId: [notes...] }`
- Fonction `addEvaluation(propertyId, note)`

2) Dans Property.jsx :

- Bouton "Ajouter une évaluation"
- Affichage d'une liste 1 à 5 (select)
- Au clic :
 - la note est ajoutée via Zustand
 - console.log de la liste des notes
 - calcul de la moyenne
 - si moyenne > 4.5 => alert("Propriété excellente !")

Snippets React utiles :

- GET données :

```
useEffect(() => {  
  fetch("http://localhost:3001/properties")  
    .then(res => res.json())  
    .then(data => setProperties(data));  
}, []);
```

- PATCH données :

```
fetch("http://localhost:3001/properties/" + id, {  
  method: "PATCH",  
  headers: { "Content-Type": "application/json" },  
  body: JSON.stringify({ views: views + 1 })  
});
```

- Redirection :

```
navigate("/reserve", { state: { property } });
```

- Zustand :

```
const { addEvaluation } = useEvaluationStore();  
addEvaluation(property.id, note);
```