

Entretien Technique

Théo - Développeur Full Stack TS

Appuyer sur espace pour commencer →

Déroulé de l'entretien

Discussion

Questions rapides

Code Review

Mise en situation

Discussion

Vision et veille

Comment fais-tu ta veille technique ?

Vision et veille

C'est quoi du bon code ?

JavaScript / TypeScript

C'est quoi TypeScript ?

JavaScript / TypeScript

Asynchrone vs Synchrone

React

Quelles sont les différences principales entre composants fonctionnels et composants classes en React ?

React - Optimisation

Comment éviter des rendus inutiles dans un composant ?

React - State management

Comment gérer l'état local dans un composant fonctionnel ?

React - State management

Quelles sont les différences entre props, state et context, et dans quels cas utiliser chacun ?

React

Quelles sont les libraries que tu aimes utiliser avec React ?

React - Testing

Comment t'assures tu que ton app React soit bien testée ?

React - Next.Js

As-tu déjà travaillé avec Next.js ?

Backend

Comment conçois-tu une API REST propre et maintenable ?

Containerisation

En Docker, quelle est la différence entre une image et un container ?

Containerisation

Explique-moi ton workflow Docker pour le développement

Questions Rapides

Développeur

MacOS , Windows ou Linux ?

JavaScript / TypeScript

Différence entre map() et forEach() ?

JavaScript / TypeScript

C'est quoi une closure ?

JavaScript / TypeScript

npm, yarn, pnpm, bun, deno ?

Frontend

Ta façon préférée de gérer le CSS (outil , méthode)

Frontend

Explique moi une bonne pratique de performance web côté front.

Frontend

Ca te parle le strictMode sur React ?

Testing - Cypress

Que fait cy.visit() ?

Développeur - GIT

Git, en ligne de commande ou avec un outil graphique ?

Développeur - GIT

Sur Git, quelle est la différence entre un merge et un rebase ?

Sécurité

Différence entre HTTP et HTTPS

Sécurité

C'est quoi les CORS ?

Sécurité

C'est quoi un token JWT ?

Code Reviews

React

```
function UserList({ users }) {
  return (
    <ul>
      {users.map((user, index) => (
        <li key={index}>{user.name}</li>
      )));
    </ul>
  );
}
```

React

```
useEffect(() => {
  const interval = setInterval(() => console.log('Ping'), 1000);
}, []);
```

React

```
useEffect(() => {
  fetch('/api/data')
    .then(res => res.json())
    .then(data => setData(data));
}, []);
```

React

```
import React, { useState, useMemo, useCallback } from 'react';

function JetCounter() {
  const [count, setCount] = useState(0);

  const doubled = useMemo(() => count * 2, [() => count]);

  const logCount = useCallback(() => {
    console.log('Count:', count);
  }, []);

  return (
    <>
      <div>{doubled}</div>
      <button onClick={() => setCount(count + 1)}>Increment</button>
      <button onClick={logCount}>Log Count</button>
    </>
  );
}
```

TypeScript

```
const apiResponse = {  
  id: 123,  
  name: "Book",  
  price: 19.99,  
  tags: ["fiction", "bestseller"]  
};
```

TypeScript

```
get<T>(key: string): T | undefined;
get<T>(key: string, defaultValue: T): T;
get<T>(key: string, defaultValue?: T): T | undefined {
    try {
        const rawData = this.#storage.getItem(` ${STORAGE_PREFIX}${key}` );
        if (!rawData) {
            return defaultValue;
        }

        return JSON.parse(rawData);
    } catch (error) {
        if (error instanceof SyntaxError) {
            console.error('Failed to parse cache data:', error);
            this.remove(key);
        }
    }

    return defaultValue;
}
}
```

Mise en situation

Tu as trois demandes urgentes en même temps :

- une correction de bug
- une fonctionnalité à livrer rapidement
- et une revue de code mandatée par le client.

Comment organises-tu ton travail ?

Tu es en réunion avec un collègue qui ne partage pas ton avis un choix technique critique.

Comment gères-tu cet échange ?

Fin