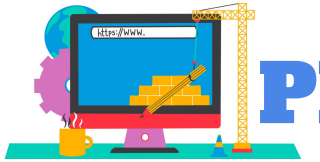
Website rendering types

Où mettre en œuvre la logique et le rendu dans une application web ?



Plan

Website rendering types

I. Type de rendu

1) ‘Server-Side Rendering’
2) ‘Client-Side Rendering’
3) ‘Server-Side’ **VS** ‘Client-Side’
4) ‘Static Rendering’
5) ‘Server-Side’ **VS** ‘Static’
6) ‘Prerendering’
7) ‘Static’ **VS** ‘Prerendering’
8) ‘Universal Rendering’

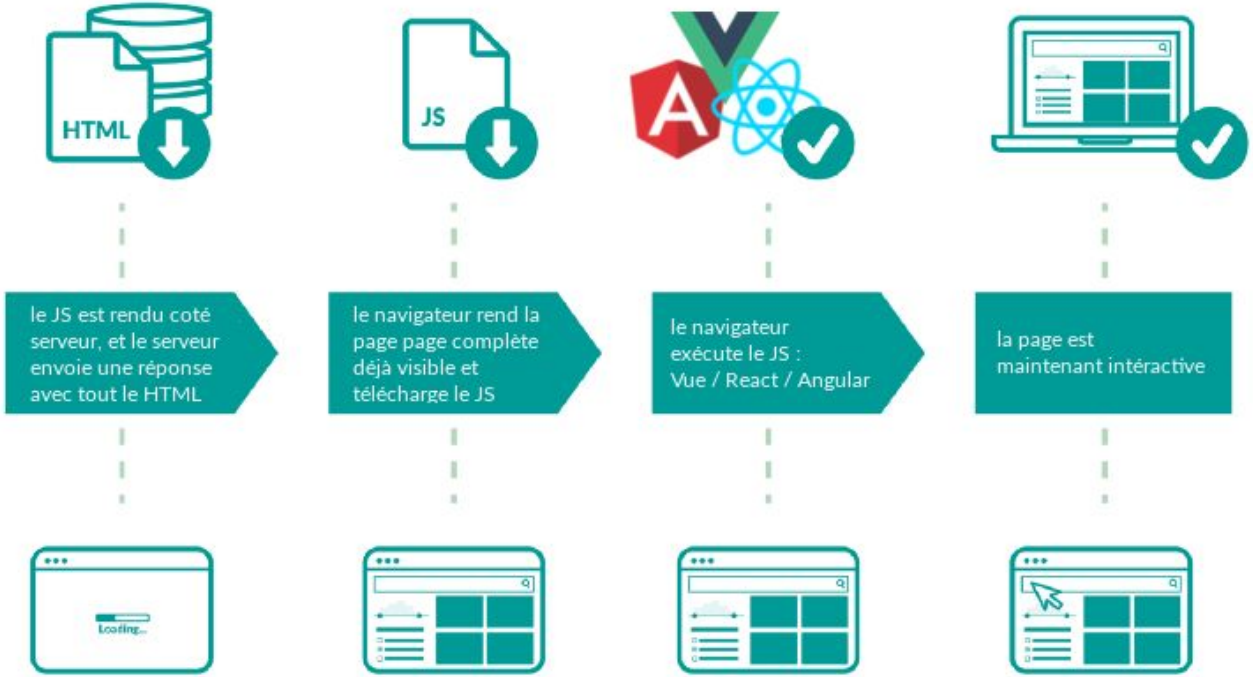
II. Performance et expériences utilisateurs

1) ‘Time to First Byte’
2) ‘First Paint’
3) ‘First Contentful Paint’
4) ‘Time To Interactive’

III. Demonstration de l’application next.js

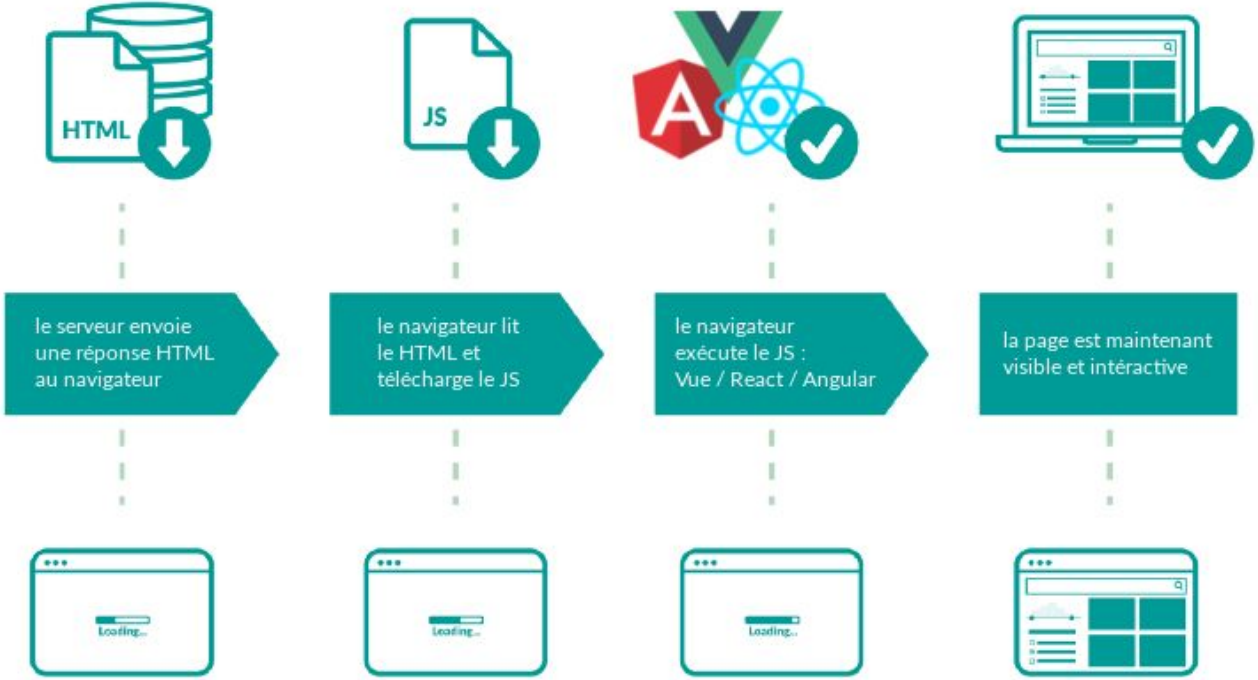
I. Rendering types

1) ‘Server-Side Rendering’ (CSR)



I. Rendering types

2) ‘Client-Side Rendering’ (CSR)



I. Rendering types

3) ‘Server-Side Rendering’ VS ‘Client-Side Rendering’

Les limites du rendu côté client :

- Pas d’HTML donc mauvais résultats SEO (moteurs de recherches)
- Temps de chargement initial élevé

Le rendu côté serveur permet quand à lui d’optimiser la performance et le SEO.

**SEO = ‘Search Engine Optimisation’*

I. Rendering types

4) ‘Static Rendering’

Le rendu statique signifie la production d’un fichier HTML distinct pour chaque URL à l’avance. Les réponses HTML étant générées à l’avance, les rendus statiques peuvent être déployés sur plusieurs réseaux de diffusion de contenu (CDN) pour tirer parti de la mise en cache périphérique.

Premier rendu rapide

car le HTML d’une page n’a pas besoin d’être générée à la volée.

**CDN = ‘Content Delivery Network’*

I. Rendering types

5) ‘Server Side Rendering’ VS ‘Static rendering’

L'avantage du rendu du serveur est la possibilité d'extraire plus de données «en direct» et de répondre à un ensemble de requêtes plus complet que ce qui est possible avec le rendu statique.

Mais ce n'est pas parce que le rendu du serveur peut faire apparaître quelque chose plus tôt que vous avez moins de travail à faire.

Le rendu du serveur n'est pas une solution miracle. Sa nature dynamique peut entraîner des frais généraux de calcul importants.

I. Rendering types

6) ‘Prerendering’

Le pré-rendu est un **processus de pré chargement** des éléments de la page en vue d'un robot d'exploration Web pour le voir.

Un service de pré-rendu intercepte la demande de page pour **voir si l'agent utilisateur qui visualise votre site est un bot.**

- Si c'est le cas, alors le middleware de pré-rendu enverra une version en cache de votre site à afficher avec tous les éléments rendus statiquement. (JavaScript, images, etc)
- Sinon, alors tout est chargé normalement.

I. Rendering types

7) ‘Static Rendering’ **VS** ‘Prerendering’

Les pages de rendu statique sont interactives sans qu'il soit nécessaire d'exécuter beaucoup de JS côté client.

Tandis que le prérendu améliore le premier rendu d'une application à page unique qui doit être démarrée sur le client pour que les pages soient vraiment interactives.

I. Rendering types

8) ‘Universal Rendering’

Un compromis entre le rendu côté client et le rendu serveur en faisant les deux.

Inconvénients :

- Impact négatif significatif sur les performances 'Time To Interactive', même s'il améliore les performances 'First Paint'.
- Les pages côtés serveur semblent souvent chargées et interactives de manière trompeuse, mais ne peuvent pas réellement répondre aux entrées tant que le JS côté client n'est pas exécuté et que les gestionnaires d'événements ont été attachés. Cela peut prendre quelques secondes, voire quelques minutes sur mobile.

II. Performance

‘Time to First Byte’

- Le temps entre le clic sur un lien et le premier bit de contenu entrant.

‘First Paint’

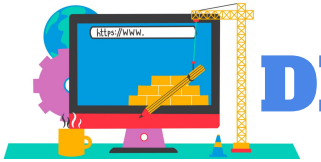
- La première fois qu'un pixel devient visible pour l'utilisateur.

‘First Contentful Paint’

- Le moment auquel le contenu demandé (corps de l'article, etc.) devient visible.

‘Time To Interactive’

- Le moment auquel une page devient interactive (événements câblés, etc.).



DEMONSTRATION

Next.js