

Cadriciel Web côté serveur

Cours 1 - Intro Cadriciel (côté serveur)

Steve Lévesque, Tous droits réservés © où applicables

Table des matières

- 1 Côté Client et Serveur - Différences
- 2 Vue d'ensemble de l'architecture Web et l'emplacement du Cadriciel Web côté serveur dans celui-ci
- 3 Limite réseau (network boundary)
- 4 Différences principales entre la partie Client et Serveur
- 5 Exercice

Côté Client et Serveur - Différences

Pour comprendre le fonctionnement des composants serveur et client, il est utile de se familiariser avec deux concepts Web fondamentaux :

- Les environnements dans lesquels votre code d'application peut être exécuté : le serveur et le client.
- Limite réseau (network boundary) qui sépare le code du serveur et du client.

Côté Client et Serveur - Différences

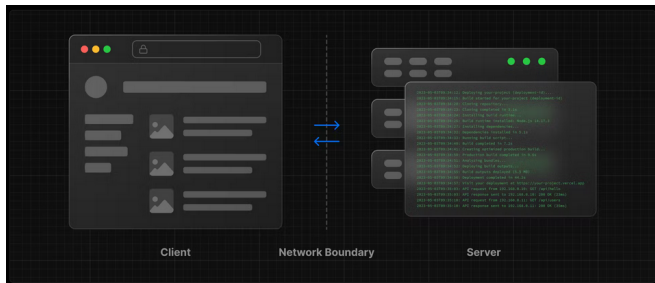


Figure: <https://nextjs.org/learn/react-foundations/server-and-client-components>

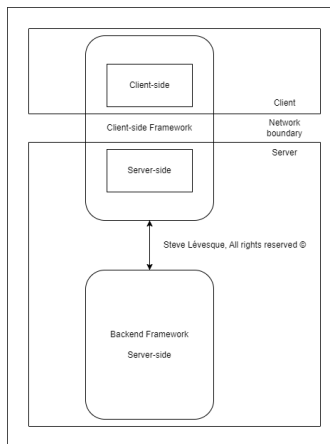
Vue d'ensemble de l'architecture Web et l'emplacement du Cadriciel Web côté serveur dans celui-ci

Avec le côté serveur du Cadriciel Web, il est possible d'exécuter des tâches sans que l'utilisateur y soit impliqué.

Cette partie reste comme une composante Frontend, et le Backend est complètement en arrière et abstrait tout de même.

On peut considérer que la partie du Cadriciel côté Serveur est entre le Cadriciel Web côté client et le Backend.

Vue d'ensemble de l'architecture Web et l'emplacement du Cadriciel Web côté client dans celui-ci



Limite réseau (network boundary)

Maintenant que nous avons une idée des deux types de Components, comment sont-ils mutuellement reliés ?

Par le fait que les Components serveur peuvent importer des Components client et faire leur rendu final de cette manière.

Limite réseau (network boundary)

Par exemple, il est possible de créer un Component Nav qui est rendu sur le serveur et partagé entre les pages, mais pour afficher un état actif pour les liens, il faut afficher la liste des liens sur le client puisqu'il y a les variables réactives de Hooks et Pathnames (client uniquement).

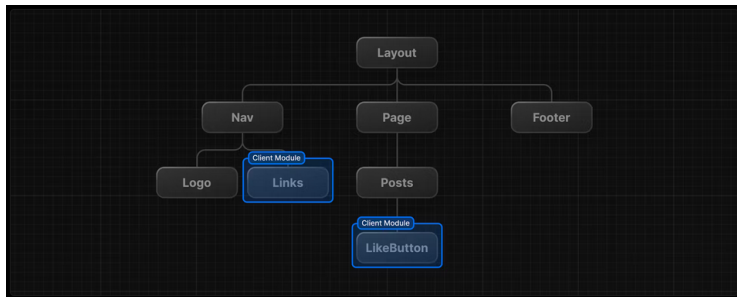


Figure: <https://nextjs.org/learn/react-foundations/server-and-client-components>

Différences principales entre la partie Client et Serveur du Cadriciel - Atouts

Cadriciel Client : Utilisation des variables réactives (Hooks, pathname, etc.) pour changer le DOM du client.

Cadriciel Serveur : Rendu sécurisé et côté serveur du réseau, donc invisible aux yeux de l'utilisateur.

Différences principales entre la partie Client et Serveur du Cadriciel - Défauts

Cadriciel Client : Les informations sont public à l'utilisateur puisque ceci est le côté navigateur Web (i.e. variables d'environnements complètement visible) et les opérations sont plus lentes puisqu'il en incombe à l'ordinateur de l'utilisateur d'exécuter le tout.

Cadriciel Serveur : Impossible d'accéder à des informations du client et d'utiliser des variables réactives (Hooks, pathname, etc.) puisque le côté du réseau est strictement caché du navigateur Web.

Exercice

Suivre les étapes d'un exercice introductif sur l'intégration de Components client dans un Component serveur.

Les étapes sont ici : <https://nextjs.org/learn/react-foundations/server-and-client-components>

Bibliographie

- `https://nextjs.org/learn/react-foundations/server-and-client-components`