

Partie 7 Frameworks JavaScript

Chapitre 7-1 Positionnement des frameworks JavaScript

1. Présentation générale des frameworks JavaScript

De très nombreux frameworks JavaScript existent, avec des positionnements fonctionnels différents.

Il ne peut être question, dans le cadre de ce livre réservé à des débutants en JavaScript, d'en faire une revue exhaustive.

Ils ont tous les points communs suivants : masquer la complexité du langage JavaScript, apporter de la robustesse dans les développements et aussi permettre, pour certains d'entre eux, d'interagir avec des bases de données.

1.1 Frameworks « front-end »

Les plus populaires des frameworks dits « front-end », c'est-à-dire gérant le côté interface utilisateur des applications web ou mobiles (téléphones mobiles, tablettes...) sont :

- Angular, framework développé par Google (la première version était connue sous l'appellation AngularJS)
- React JS (ou React), framework développé par Facebook
- Vue.js
- Svelte

1.2 Frameworks « back-end »

Pour les interactions avec les systèmes de gestion de bases de données, des frameworks dits « back-end » existent. Ils sont souvent eux-mêmes basés sur Node.js, qui est un environnement d'exécution multiplateforme Open Source exécutant du code JavaScript en dehors d'un navigateur (dans un runtime).

Node.js permet de concevoir des services d'accès à des Bases De Données et à des ressources disponibles sur Internet. Il fonctionne parfaitement sur Windows, Linux ou encore macOS.

Nous verrons par exemple que l'accès aux données pour le framework Svelte peut être assuré par les frameworks Express ou Sapper, tous deux basés sur Node.js.

1.3 Solutions de développement « hybride »

Pour terminer ce rapide panorama concernant les frameworks JavaScript, n'oublions pas les solutions dédiées aux développements pour périphériques mobiles (smartphones et tablettes). Dans un précédent livre, publié en 2020 aux Éditions ENI (Java et Ionic - Développement mobile pour Android : natif vs hybride), le framework Ionic a été présenté. Ce framework est lui-même basé sur d'autres frameworks, dont le très réputé Angular (soutenu par Google) et Apache Cordova.

Un chapitre du présent livre sera aussi consacré au Framework React Native (assez proche de React qui, lui, est réservé aux applications web). React Native permet très facilement à des développeurs ayant déjà une réelle expérience en React de concevoir des applications pour mobiles. Ce framework extrêmement utilisé et supporté par Facebook est une alternative plus que crédible à Ionic. Les applications React Native sont facilement déployables sur les mobiles fonctionnant sous systèmes d'exploitation Android et iOS (iPhone, iPad).

2. Les frameworks Node.js, Svelte, React et React Native

Comme indiqué précédemment, un court chapitre (Installation de Node.js) sera consacré à l'installation du framework Node.js, socle sur lequel fonctionnent les frameworks Svelte, React et React Native.

Le framework Svelte, relativement récent, est un challenger crédible pour React (React JS) et Vue.js. Svelte, présenté dans le chapitre Framework Svelte, possède de nombreux atouts techniques. Il bénéficie par contre pour l'instant d'une communauté de développeurs plus restreinte que celles des deux acteurs principaux (React et Vue.js).

Dans le cadre de ce livre, un choix éditorial a été fait de ne pas évoquer Vue.js. Vous trouvez, toujours aux Éditions ENI, des ouvrages dédiés exclusivement à Vue.js, notamment le livre Vue.js - Développez des applications web modernes en JavaScript de Yoann GAUCHARD.

492 _____ Apprendre à développer

avec JavaScript

Le chapitre Framework React sera celui dédié à React. Comme dans le chapitre consacré à Svelte, après une rapide présentation des concepts de base, de nombreux exemples seront proposés et largement commentés.

Le livre se terminera au chapitre Framework React Native avec un exposé consacré à React Native, la version du framework React permettant le développement d'applications pour mobiles.

Chapitre 7-2

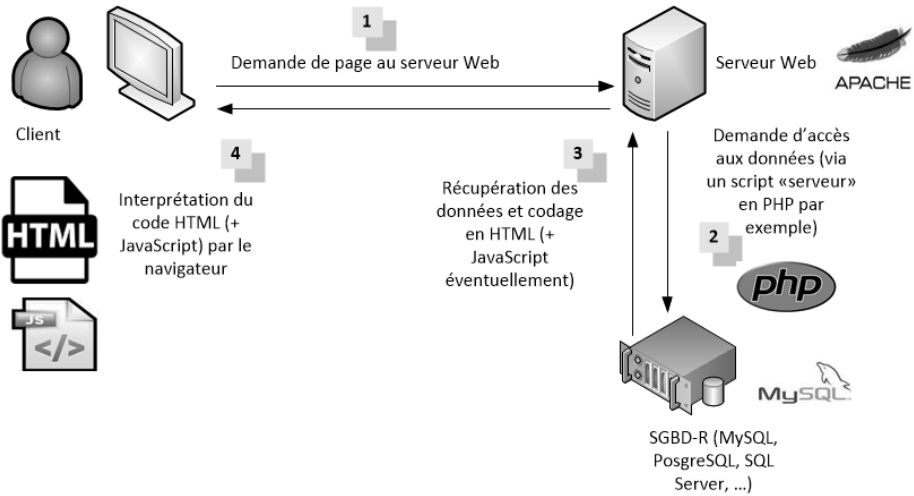
Installation de Node.js

1. Présentation du framework Node.js

JavaScript a longtemps été cantonné à une utilisation côté client. On utilisait JavaScript seulement pour ajouter de l'interactivité dans les pages web (animations, contrôles de saisie...).

Pour les accès aux Bases De Données distantes, comme MySQL, les applications web intégraient par ailleurs des scripts orientés serveur, là aussi souvent développés en langage PHP.

494 _____ Apprendre à développer avec JavaScript



Avec Node.js, il est bien sûr toujours possible d'utiliser JavaScript côté client pour manipuler les pages HTML. En plus, Node.js propose un environnement côté serveur qui permet aussi d'utiliser le langage JavaScript pour générer des pages web. En clair, il vient en remplacement de langages serveur comme PHP, Java EE, etc.

