

Architecture n-tiers

Rapport : Choix Technologiques

Notre projet nTiers consiste à réaliser une application web permettant de partager des photos et de les visualiser de manière synchronisée.

La première réalisation fut de créer un serveur web permettant d'héberger notre application afin qu'elle puisse être disponible par différents clients. Pour cela nous nous sommes orientés vers une machine virtuelle linux sur laquelle nous avons installé Apache, MySQL et PHP. Ce choix est justifié de par l'habitude d'utiliser des LAMP pour nos différents projets.

Ensuite, nous avons réalisé l'application en elle-même. Pour une architecture web, nous avons, à l'aide du bloc INFO6, appris à maîtriser le côté back-end par springboot (Java) et le côté front-end avec Angular. De plus, ces technologies sont très adaptées à ce genre de projet qui requiert une base de données communicant avec une page web à l'aide d'appels API. C'est donc pour cela que nous avons choisi ces technologies.

FRONT-END

En rentrant plus dans les détails, Angular possède une approche de type composants très pratique dans notre cas (une navbar, un bloc diaporama, une liste de photos...) puisque tous ces composants sont traités indépendamment. Ces composants sont regroupés dans différentes pages web de notre application, gérées via le routage d'angular (gérant les différents chemins avec les composants à afficher). Nous utilisons des librairies de mise en forme (bootstrap, mdb, ng-material...) afin de rendre l'application plus attrayante et agréable. Enfin, la gestion des photos, pour le côté front-end, se fait par appel API pour récupérer les informations de la BDD mais aussi pour en envoyer. L'affichage de ces dernières se fait dans une liste et aussi via une librairie gérant les diaporamas.

BACK-END

Côté back-end, nous avons choisi une architecture Java avec le framework spring permettant de d'exposer une api consommée par un front end ainsi que la gestion de l'application et de ses données. Spring boot permet donc d'incorporer des annotations à notre code gérant l'ajout de photos, la modification de commentaires... afin de retranscrire côté API les actions à mener (push,patch,delete...) ,qui elles, sont en langage HTML.

BDD

Côté données nous avons choisi d'utiliser mySQL, car une base de données de type sql est adaptée à notre projet comportant un jeu de données structurées (photos et commentaires). Enfin notre base de données comporte une table contenant deux éléments : un commentaire (chaîne de caractères) associé à une image (lien vers le répertoire des images sur le gestionnaire de fichiers du serveur).

Pour résumer, le serveur sera hosté sur une machine virtuelle linux LAMP (avec un serveur web apache) afin de pouvoir être accessible pour n'importe quel client qui s'y connecte. Il contiendra donc notre back-end, notre base de données et un répertoire contenant les images qui y seront uploadées. Côté client, l'accès se fait par navigateur web via l'adresse ip du serveur (ou par lien web si l'on configure un DNS).