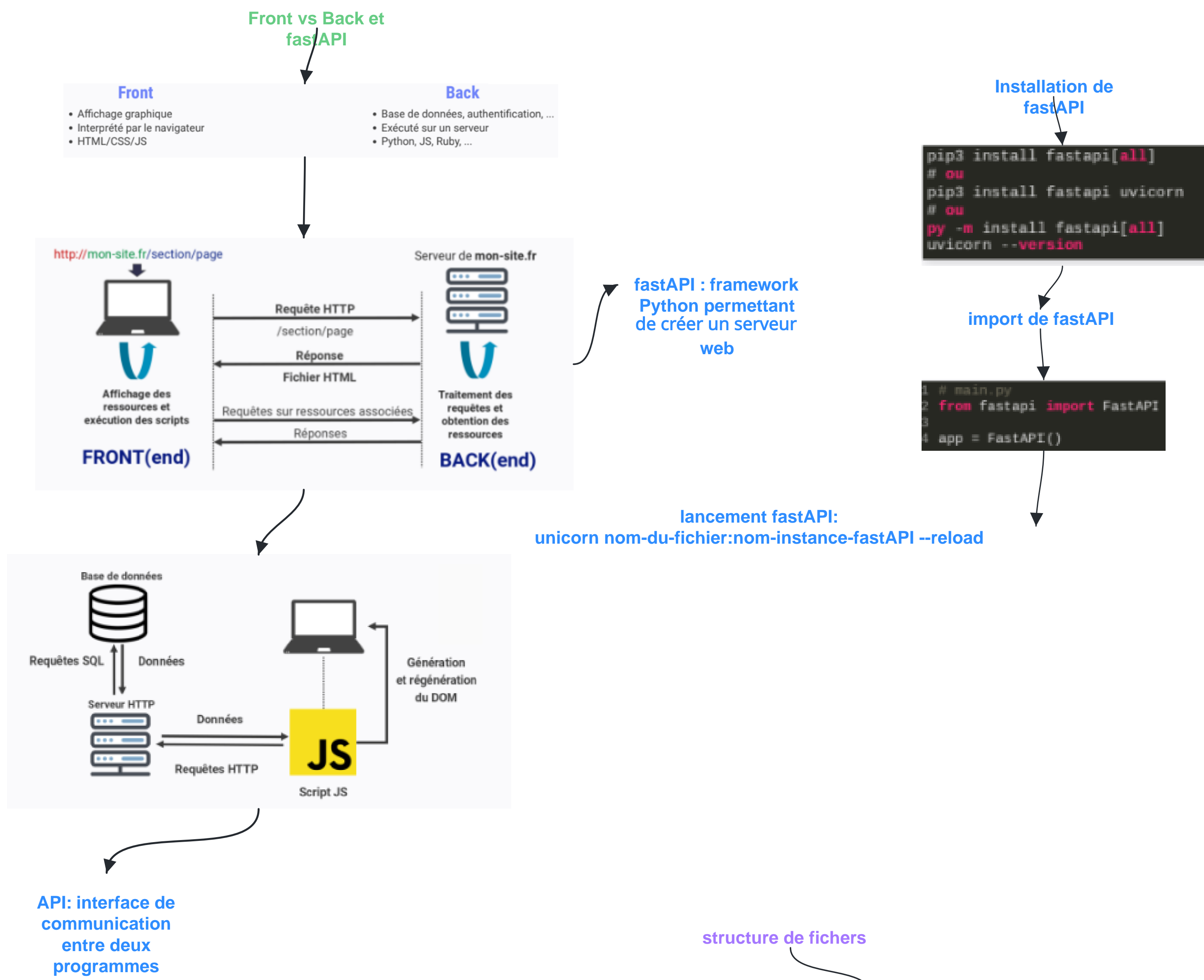


- Les API (Application Programming Interfaces) sont des interfaces de programmation qui permettent à différentes applications de communiquer entre elles. Dans une architecture 2 tiers ou 3 tiers, les API sont utilisées pour faciliter la communication entre les différents niveaux de l'architecture.
- Dans une architecture 2 tiers, les API sont utilisées pour permettre la communication entre le client et le serveur. Par exemple, si vous utilisez une application mobile pour accéder à un service en ligne, l'API est utilisée pour permettre à l'application mobile de communiquer avec le serveur.
- Dans une architecture 3 tiers, les API sont utilisées pour permettre la communication entre les différents niveaux de l'architecture. Par exemple, si vous utilisez une application Web pour accéder à un service en ligne, l'API est utilisée pour permettre à l'application Web de communiquer avec le serveur d'application, qui communique ensuite avec la base de données.
- L'utilisation des API dans une architecture 2 tiers ou 3 tiers présente plusieurs avantages. Tout d'abord, cela permet de séparer les différents niveaux de l'architecture, ce qui facilite la maintenance et l'évolutivité de l'architecture. De plus, cela permet de réduire la complexité de l'architecture en fournissant une interface standardisée pour la communication entre les différents niveaux. Enfin, cela permet de faciliter l'intégration de nouveaux services ou applications dans l'architecture.



Les verbes GET, POST, PUT et DELETE sont des verbes HTTP qui sont utilisés pour effectuer des opérations CRUD sur les ressources d'un serveur

- GET : requêtes de récupération des données (exécuté par un navigateur pour afficher une page)

- POST: requêtes de création des données

- PUT modifier une valeur

- DELETE supprimer des données

- La différence principale entre les requêtes **POST** et **PUT** est que **PUT** est idempotent, c'est-à-dire que l'appeler une ou plusieurs fois successivement a le même effet (c'est-à-dire pas d'effet secondaire), tandis que les demandes **POST** identiques successives peuvent avoir des effets supplémentaires, comme si vous passiez une commande plusieurs fois.

- Par exemple, si vous utilisez une requête **PUT** pour modifier un objet, vous pouvez appeler cette requête plusieurs fois sans effet secondaire. En revanche, si vous utilisez une requête **POST** pour ajouter un objet, chaque appel de cette requête ajoutera un nouvel objet.

- En général, vous pouvez utiliser la méthode **PUT** lorsque vous devez modifier une seule ressource, tandis que la méthode **POST** est utilisée lorsque vous devez ajouter une ressource enfant

- Par exemple, si vous utilisez une API pour poser des questions, vous pouvez utiliser la méthode **POST** pour ajouter une nouvelle question à une liste de questions existante. En revanche, si vous voulez mettre à jour une question existante, vous pouvez utiliser la méthode **PUT**.