



Ingénieur Back-End

▼ Qu'est ce que le Back-end ?

Le back end désigne **les parties du code d'une application ou d'un logiciel permettant son fonctionnement et inaccessibles à l'utilisateur**. Il fait référence à la partie d'un système informatique qui n'est pas directement visible par les utilisateurs finaux. Il englobe les serveurs, les bases de données, l'application elle-même, et tout ce qui se passe côté serveur. En d'autres termes, le back-end gère la logique métier, le traitement des données, et la gestion des requêtes provenant du front-end (la partie visible par les utilisateurs)..

▼ Qu'est ce que l'ingénieur Back-end ?

Le développeur back-end a pour mission de **concevoir tous les éléments techniques indispensables au fonctionnement d'un site web ou d'une application**.

Parce qu'il est responsable de l'évolution de l'infrastructure du site web, le développeur back-end doit aussi **être en mesure de développer la plateforme de gestion des contenus et de développer les applications web**. Si des problèmes persistent, le développeur back-end préconise des solutions pour le client et assure un support technique du site, ainsi qu'une maintenance réactive. Il a aussi le devoir d'aider le client si ce dernier lui livre un site. Mise à part la programmation et les opérations de paramétrage de retouche, il doit davantage se focaliser sur le côté analytique du site.

En outre, toutes les phases telles que la programmation, le paramétrage, le contrôle qualité et la documentation technique sont à la charge du développeur back-end. C'est pourquoi ses bagages informatiques doivent être complets pour le métier.

▼ Quel est le rôle d'ingénieur Back-end?

Le développement back end peut donc englober la mise en place, la configuration et le maintien de plusieurs composantes essentielles d'un projet web :

C'est quoi le Back-end (intro) : Le backend est la partie d'un logiciel que les utilisateurs ne peuvent pas voir ou avec laquelle ils ne peuvent pas interagir et qui contient toutes les fonctionnalités. Selon le contexte, le développeur back-end peut avoir un rôle plus ou moins large: Il peut aussi bien être un exécutant (avec un rôle « simple » de codeur), qu'un décisionnaire dans les choix techniques.

Roadmap :

Un développeur back-end est responsable du développement des composants côté serveur d'une application Web, c'est-à-dire travailler avec des bases de données, gérer les requêtes, créer des API côté serveur qui peuvent être utilisées par les développeurs front-end pour récupérer et manipuler des données, assurer l'évolutivité des systèmes, c'est-à-dire en s'assurant que le backend peut gérer un volume de trafic élevé et est performant, en intégrant des services externes tels que des passerelles de paiement, des files d'attente de messages, des services cloud, etc.

1. Langages de Programmation :

- Maîtrise d'au moins un langage de programmation (Java, Python, Node.js, etc.).
- Compréhension des frameworks associés.

2. Bases de Données :

- Conception de bases de données relationnelles (MySQL, PostgreSQL).
- Expérience avec des bases de données NoSQL (MongoDB, Redis).

3. APIs et Services Web :

- Conception et développement d'APIs RESTful.
- Expérience avec des technologies telles que GraphQL.

4. Systèmes d'exploitation et Réseaux :

- Connaissance des systèmes d'exploitation (Linux/Unix).
- Compréhension des concepts réseau.

5. Frameworks Backend :

- Utilisation de frameworks Backend (Spring Boot, Django, Express.js, etc.).
- Architecture de microservices.

6. Sécurité des Applications :

- Pratiques de sécurité des applications.
- Gestion des autorisations et des accès.

7. Tests et Débogage :

- Écriture de tests unitaires et d'intégration.
- Débogage des applications.

8. Outils de Développement :

- Utilisation d'IDE (Environnement de Développement Intégré) et d'outils de gestion de versions.

9. Performance et Évolutivité :

- Optimisation des performances des applications.
- Conception pour l'évolutivité.

10. Collaboration et Communication :

- Travail efficace avec les équipes front-end, DevOps et autres.
- Utilisation d'outils de collaboration comme Jira.

▼ Comment devenir Ingénieur Back-end ?

Pour devenir ingénieur back-end, vous pouvez suivre ces étapes générales :

1. **Formation académique** : La plupart des ingénieurs back-end ont une formation en informatique, en génie logiciel, en sciences informatiques ou dans un domaine connexe.
2. **Apprentissage des langages de programmation** : Maîtrisez au moins un langage de programmation couramment utilisé pour le développement back-end, comme Python, Java, Ruby, PHP, ou Node.js. Familiarisez-vous également avec les concepts de bases de données et les langages de requêtes associés (SQL, par exemple).
3. **Apprentissage des Frameworks** : Explorez les frameworks back-end populaires tels que Django (Python), Ruby on Rails (Ruby), Spring (Java), Laravel (PHP), ou Express.js (Node.js). Ces frameworks simplifient le développement en fournissant des structures et des fonctionnalités prêtes à l'emploi.
4. **Connaissance des bases de données** : Apprenez à concevoir, créer et interroger des bases de données. Familiarisez-vous avec différents systèmes de gestion de base de données (SGBD) tels que MySQL, PostgreSQL, MongoDB, etc.
5. **Maîtrise des concepts web** : Comprenez les concepts fondamentaux du développement web, y compris les protocoles HTTP/HTTPS, les méthodes de requête (GET, POST, etc.), et les architectures client-serveur.
6. **Pratique et Projets personnels** : Mettez en pratique vos connaissances en travaillant sur des projets personnels ou en contribuant à des projets open source. Cela vous permettra de développer votre portfolio et de démontrer vos compétences.
7. **Formation complémentaire** : Participez à des cours en ligne, des ateliers ou des formations professionnelles pour rester à jour sur les dernières technologies et tendances du développement back-end.

Ingénieur Back-end :

▼ SLIDE

1. Outils de gestion de versions :

- Git et GitHub , Gitlab

2. Outil de script :

- python avec faire scraping des données

3. Logiciel de communication :

- Slack

4. Langages de programmations :

- **JavaScript :**

Nest.js , **Node.js**, Un framework extensible pour construire des applications serveur robustes en utilisant TypeScript.

- **Python :**

Flask , Un micro-framework qui est simple et extensible.

Django, Un framework web complet qui facilite le développement rapide et propre en Python.

- **PHP :**

Symfony , Un ensemble de composants PHP réutilisables et un framework web.

- **Java :**

Spring Boot , Un framework de développement d'applications Java basé sur Spring, conçu pour simplifier le développement d'applications.

5. Base de données :

- **MySQL** : Une base de données relationnelle populaire.
- **SQLite** : Une base de données légère intégrée dans l'application.

▼ Quelles possibilités d'évolutions dans ce domaine ?

plusieurs possibilités d'évolution s'offrent à vous, notamment :

1. **Chef de Projet** : Vous pourriez évoluer vers un rôle de chef de projet, où vous seriez responsable de la coordination de l'équipe de développement, de la gestion des projets et des délais.
2. **Architecte Logiciel** : En tant qu'architecte logiciel, vous seriez impliqué dans la conception et l'architecture de systèmes complexes, en guidant les choix technologiques et les décisions d'implémentation.
3. **Développeur Full-stack** : Certains ingénieurs back-end élargissent leurs compétences pour devenir des développeurs full-stack, capables de travailler à la fois sur le front-end et le back-end.
4. **Expert en Sécurité Informatique** : Si vous avez un intérêt particulier pour la sécurité informatique, vous pourriez évoluer vers un rôle spécialisé dans la sécurisation des applications et des systèmes.
5. **Entrepreneuriat** : Certains ingénieurs back-end choisissent de créer leur propre entreprise ou de participer à des startups en tant que fondateurs ou membres clés de l'équipe technique.

Ingénieur Back-end :

▼ SLIDE

1. Outils de gestion de versions :

- Git et GitHub , Gitlab

2. Outil de script :

- python avec faire scraping des données

3. Logiciel de communication :

- Slack

4. Langages de programmations :

- **JavaScript** :

Nest.js , **Node.js**, Un framework extensible pour construire des applications serveur robustes en utilisant TypeScript.

- **Python** :

Flask , Un micro-framework qui est simple et extensible.

Django, Un framework web complet qui facilite le développement rapide et propre en Python.

- **PHP :**

Symfony , Un ensemble de composants PHP réutilisables et un framework web.

- **Java :**

Spring Boot , Un framework de développement d'applications Java basé sur Spring, conçu pour simplifier le développement d'applications.

5. **Base de données :**

- **MySQL** : Une base de données relationnelle populaire.
- **SQLite** : Une base de données légère intégrée dans l'application.

▼ **Quels outils utilise-t-il ?**

Les développeurs backend utilisent un ensemble d'outils pour concevoir, développer, déployer et maintenir des applications côté serveur. Voici quelques-uns des outils couramment utilisés dans le domaine du développement backend :

1. **Environnements de Développement Intégré (IDE) :**

- Visual Studio Code
- PyCharm (pour le développement Python)

2. **Systèmes de Contrôle de Version :**

- Git
- GitHub
- GitLab

3. **Outils de Gestion de Dépendances :**

- npm (pour Node.js)
- pip (pour Python)
- Composer (pour PHP)
- Maven (pour Java)

4. **Frameworks Backend :**

- Django (Python)
- Spring (Java)
- Express.js (Node.js)
- Laravel (PHP)

5. Bases de Données :

- MySQL
- PostgreSQL
- MongoDB
- Redis (pour le stockage de données en mémoire)

6. Serveurs Web :

- Apache
- Nginx

7. Outils de Test :

- JUnit (pour Java)
- PyTest (pour Python)
- PHPUnit (pour PHP)
- Mocha (pour Node.js)

8. Outils de Documentation :

- Swagger
- API Blueprint
- Javadoc (pour Java)
- Sphinx (pour Python)

9. Outils de Déploiement et d'Orchestration :

- Docker
- Kubernetes
- Ansible
- Jenkins

10. Outils de Surveillance et de Journalisation :

- Prometheus
- Grafana
- ELK Stack (Elasticsearch, Logstash, Kibana)

11. Outils de Sécurité :

- OWASP ZAP (pour les tests de sécurité)
- Nessus (pour la détection de vulnérabilités)
- Vault (pour la gestion des secrets)

12. Outils de Collaboration :

- Slack
- Jira
- Confluence
- Microsoft Teams

13. Outils de Communication :

- Postman (pour tester les APIs)
- cURL (pour effectuer des requêtes HTTP en ligne de commande)

14. Environnements de Développement Virtuel :

- Virtualenv (pour Python)
- npm (pour Node.js)
- Vagrant

15. Éditeurs de Texte :

- Vim
- Emacs
- Sublime Text

▼ Ingénieur Back-end (soft skills)

- **Communication**
- **Adaptabilité**
- **Gestion du temps**

- **Pensée analytique**
- **Autonomie**