



---

# FONDAMENTAUX BACK END : PHP, SQL, FRAMEWORKS

---

Executive summary

Alexis Persinette-Gautrez

# I. Fondamentaux de PHP

## A. Introduction à PHP

- **Origine et histoire de PHP:** PHP (Hypertext Preprocessor) est un langage de programmation open-source créé en 1994 par Rasmus Lerdorf. Initialement, PHP signifiait "Personal Home Page", mais il est maintenant considéré comme un acronyme récursif, signifiant "PHP: Hypertext Preprocessor". C'est devenu l'un des langages les plus populaires pour le développement web côté serveur.
- **Installation et configuration:** Pour débuter avec PHP, vous devez installer un serveur web comme Apache, Nginx ou IIS, puis configurer PHP en tant que module ou en utilisant un serveur PHP tel que PHP-FPM. Vous pouvez également gérer les versions de PHP via des gestionnaires de paquets tels que apt, yum ou Composer.
- **Syntaxe de base:** La syntaxe de PHP est intégrée dans le code HTML. Un script PHP commence par `<?php` et se termine par `?>`. Les instructions PHP sont généralement suivies d'un point-virgule. PHP prend en charge les variables, les opérations arithmétiques, les boucles, les conditions, les fonctions, etc.
- **Variables, constantes et types de données:** PHP offre une grande flexibilité en matière de gestion des données. Les variables commencent par `$`, suivies du nom de la variable. Les types de données pris en charge incluent les entiers, les chaînes de caractères, les tableaux, les objets, etc. Les constantes, définies avec `define()`, sont immuables une fois définies.

## B. Gestion des requêtes HTTP en PHP

- **Traitement des formulaires HTML:** PHP est couramment utilisé pour traiter les données des formulaires HTML soumis par les utilisateurs. Les données peuvent être récupérées à l'aide de la superglobale `$_POST` ou `$_GET`, selon la méthode de soumission.
- **Méthodes GET et POST:** PHP prend en charge les méthodes GET et POST pour l'envoi de données entre le navigateur et le serveur. GET est utilisé pour les requêtes avec des données visibles dans l'URL, tandis que POST est utilisé pour les données sensibles.
- **Validation des données utilisateur:** La validation des données entrées par l'utilisateur est essentielle pour garantir leur exactitude et leur sécurité. PHP propose des fonctions de validation telles que `filter_var()` et `preg_match()`.
- **Prévention des attaques XSS et CSRF:** PHP offre des mécanismes de sécurité pour prévenir les attaques de type Cross-Site Scripting (XSS) en échappant les données de sortie. La prévention contre les attaques Cross-Site Request Forgery (CSRF) implique l'utilisation de jetons CSRF.

## C. Bases de données SQL avec PHP

- **Connexion à une base de données MySQL:** PHP permet de se connecter à une base de données MySQL à l'aide de fonctions telles que `mysqli_connect()` ou `PDO`. Cette connexion permet d'interagir avec la base de données pour stocker et récupérer des données.
- **Exécution de requêtes SQL:** PHP facilite l'exécution de requêtes SQL à partir de scripts en utilisant des fonctions comme `mysqli_query()` ou des requêtes préparées avec `PDO`. Cela permet de récupérer et de manipuler les données de la base de données.
- **Sécurité des requêtes SQL:** La sécurité est essentielle lors de l'exécution de requêtes SQL. PHP propose des méthodes pour prévenir les injections SQL, notamment l'utilisation de requêtes préparées et l'échappement des données.
- **Utilisation de transactions:** Les transactions SQL garantissent la cohérence des données lors de l'exécution de plusieurs requêtes. PHP permet de gérer les transactions avec des fonctions telles que `mysqli_begin_transaction()` et `mysqli_commit()`.

## D. Sessions et authentification en PHP

- **Gestion des sessions utilisateur:** Les sessions sont utilisées pour stocker des données utilisateur persistantes entre les requêtes. PHP offre des fonctions pour démarrer, gérer et détruire les sessions utilisateur.
- **Création et vérification de l'authentification:** PHP permet la mise en place de mécanismes d'authentification en vérifiant les identifiants des utilisateurs par rapport à une base de données ou un système d'authentification.
- **Sécurité des sessions et des mots de passe:** Il est essentiel de sécuriser les sessions utilisateur et le stockage des mots de passe. PHP offre des fonctions pour sécuriser les sessions et le hachage des mots de passe à l'aide d'algorithmes appropriés.
- **Implémentation des autorisations:** PHP permet de définir des autorisations d'accès aux ressources en fonction du rôle de l'utilisateur, garantissant que seuls les utilisateurs autorisés peuvent accéder à certaines parties de l'application.

## II. Fondamentaux de SQL

### A. Introduction à SQL

- **Définition de SQL (Structured Query Language):** SQL, acronyme de "Structured Query Language", est un langage de programmation utilisé pour gérer et manipuler des bases de données relationnelles. Il permet de communiquer avec les bases de données pour effectuer diverses opérations telles que la récupération, l'insertion, la mise à jour et la suppression de données.
- **Types de bases de données relationnelles:** SQL est principalement utilisé avec des bases de données relationnelles telles que MySQL, PostgreSQL, Oracle, Microsoft SQL Server, etc. Ces bases de données stockent les données dans des tables liées par des relations.
- **Création, modification et suppression de bases de données et de tables:** SQL permet de créer des bases de données et des tables pour organiser les données de manière structurée. Vous pouvez également modifier la structure des tables et supprimer des bases de données ou des tables existantes.
- **Principales commandes SQL (SELECT, INSERT, UPDATE, DELETE):** Les commandes SQL essentielles incluent SELECT (pour la récupération de données), INSERT (pour l'insertion de nouvelles données), UPDATE (pour la mise à jour des données) et DELETE (pour la suppression de données).

### B. Requêtes SQL avancées

- **Jointures de tables (INNER, LEFT, RIGHT, FULL):** Les jointures SQL permettent de combiner des données de plusieurs tables en fonction de certaines conditions. Les types courants de jointures sont INNER, LEFT, RIGHT et FULL, chacune ayant ses propres règles de correspondance.
- **Fonctions SQL (agrégation, date, texte):** SQL propose un large éventail de fonctions pour effectuer des opérations sur les données. Cela inclut les fonctions d'agrégation (SUM, AVG, COUNT, etc.), les fonctions de manipulation de dates (DATEADD, DATEDIFF, etc.) et les fonctions de manipulation de texte (CONCAT, SUBSTRING, etc.).
- **Sous-requêtes (subqueries) et expressions communes de table (CTE):** Les sous-requêtes sont des requêtes imbriquées à l'intérieur d'une requête principale, permettant de récupérer des données plus complexes. Les expressions communes de table (CTE) permettent de créer des résultats temporaires pour les requêtes.

- **Indexation et optimisation des requêtes:** L'indexation des tables est essentielle pour accélérer les opérations de recherche. L'optimisation des requêtes implique l'utilisation de techniques telles que l'ajout d'index appropriés, la réécriture de requêtes et l'utilisation de plans d'exécution optimaux pour améliorer les performances.

## C. Sécurité et gestion des données

- **Protection contre les injections SQL:** Les injections SQL sont des vulnérabilités graves qui peuvent permettre à des attaquants d'exécuter des requêtes SQL malveillantes. Les bonnes pratiques de sécurité, telles que l'utilisation de requêtes préparées, sont essentielles pour prévenir les injections SQL.
- **Gestion des privilèges et des utilisateurs:** La gestion des privilèges et des utilisateurs garantit un accès approprié aux données. Les bases de données SQL offrent des mécanismes pour attribuer des privilèges et contrôler qui peut accéder à quelles données.
- **Sauvegarde et restauration de bases de données:** La sauvegarde régulière des bases de données est essentielle pour la récupération en cas de défaillance ou de perte de données. La restauration permet de ramener la base de données à un état précédent.
- **Bonnes pratiques de gestion des données sensibles:** Les données sensibles doivent être traitées avec soin. Les bonnes pratiques incluent le chiffrement des données, la gestion des mots de passe, la suppression sécurisée des données, et la conformité aux réglementations sur la protection des données.

## D. Intégration de SQL dans le Back End

- **Utilisation de PHP, Python, Node.js, etc., pour interagir avec des bases de données SQL:** Les langages de programmation tels que PHP, Python et Node.js offrent des bibliothèques et des outils pour interagir avec des bases de données SQL. Cela permet d'intégrer efficacement les données dans les applications Back End.
- **Création de requêtes dynamiques en fonction des besoins de l'application:** Les applications Back End doivent souvent générer des requêtes SQL en fonction des besoins de l'utilisateur. Cela peut inclure la création de requêtes dynamiques basées sur les filtres et les paramètres de l'application.
- **Sécurité des transactions et des opérations de base de données:** La sécurité des transactions garantit que les opérations de base de données sont effectuées de manière fiable, même en cas de panne du système ou d'erreur logicielle. Les transactions permettent de garantir la cohérence des données.
- **Utilisation de frameworks ORM (Object-Relational Mapping):** Les frameworks ORM simplifient l'interaction avec les bases de données SQL en permettant de manipuler les données comme des objets dans le langage de programmation plutôt que d'écrire des requêtes SQL brutes. Cela accélère le développement et réduit les erreurs.

## III. Frameworks Back End

### A. Qu'est-ce qu'un framework Back End ?

- **Définition et objectifs des frameworks Back End:** Un framework Back End est une structure logicielle qui fournit un ensemble de composants et de conventions pour faciliter le développement d'applications web côté serveur. Il offre une structure organisée pour gérer les tâches courantes de développement.

- **Avantages de l'utilisation de frameworks:** Les frameworks Back End permettent de gagner du temps en fournissant des fonctionnalités prêtes à l'emploi, une organisation du code, une sécurité renforcée, et une maintenance simplifiée.
- **Déférence entre un framework et une bibliothèque:** Contrairement aux bibliothèques qui sont des collections d'outils indépendants, les frameworks imposent une structure et un flux de travail spécifiques. Ils définissent la manière dont le code doit être écrit.
- **Exemples populaires de frameworks Back End:** Quelques exemples populaires de frameworks Back End incluent Ruby on Rails (pour Ruby), Django (pour Python), Express.js (pour Node.js), Laravel (pour PHP), et Spring (pour Java).

## B. Utilisation de Frameworks Back End en PHP

- **Présentation des frameworks PHP tels que Laravel, Symfony, CodeIgniter:** Les frameworks PHP offrent des outils pour simplifier le développement Back End. Laravel se distingue par sa simplicité et son élégance, Symfony par sa modularité, et CodeIgniter par sa légèreté.
- **Installation et configuration d'un framework Back End en PHP:** L'installation d'un framework PHP implique généralement l'utilisation d'un gestionnaire de dépendances tel que Composer. La configuration initiale peut inclure la définition de bases de données, de routes, et d'autres paramètres.
- **Structure MVC (Modèle-Vue-Contrôleur) dans les frameworks PHP:** Les frameworks PHP suivent souvent le modèle de conception MVC, qui sépare la logique de l'application en trois composants : le Modèle (pour la gestion des données), la Vue (pour l'interface utilisateur) et le Contrôleur (pour la gestion des requêtes et des réponses).
- **Création de routes, de contrôleurs et de vues avec un framework PHP:** Les routes définissent les URL de l'application, les contrôleurs gèrent la logique métier, et les vues génèrent l'interface utilisateur. Le framework PHP simplifie la création et la gestion de ces composants.

## C. Bases de données et ORM dans les frameworks Back End

- **Intégration de bases de données avec un framework Back End:** Les frameworks Back End permettent d'intégrer facilement des bases de données. Vous pouvez définir des modèles de données, des schémas de bases de données, et effectuer des opérations CRUD (Create, Read, Update, Delete) de manière simplifiée.
- **Utilisation d'ORM (Object-Relational Mapping) pour simplifier l'accès aux données:** Les ORM facilitent la gestion des données en permettant de manipuler des objets en lieu et place de requêtes SQL. Ils simplifient l'accès aux données, garantissent la portabilité entre les bases de données, et améliorent la sécurité.
- **Migration de schémas de bases de données avec des frameworks:** Les frameworks Back End fournissent souvent des outils de migration de schémas, permettant de mettre à jour la structure de la base de données de manière contrôlée et réversible.
- **Sécurité et gestion des requêtes dans le contexte d'un framework Back End:** Les frameworks Back End intègrent des fonctionnalités de sécurité pour protéger contre les attaques courantes, telles que les injections SQL. Ils gèrent également les requêtes HTTP entrantes de manière sécurisée.

## D. Sécurité, performances et déploiement

- **Méthodes de sécurité dans les frameworks Back End:** Les frameworks Back End proposent des mécanismes de sécurité intégrés, tels que la validation des données, l'authentification, l'autorisation, et la protection contre les attaques courantes (XSS, CSRF).
- **Optimisation des performances avec des frameworks Back End:** Les frameworks Back End offrent des outils pour optimiser les performances, notamment la mise en cache, la gestion

des requêtes asynchrones, la réduction des requêtes inutiles, et l'optimisation de la base de données.

- **Déploiement d'applications Back End avec des frameworks:** Le déploiement d'applications Back End implique la mise en place de serveurs web, de bases de données et de configurations spécifiques au framework. Les frameworks fournissent souvent des guides de déploiement pour faciliter cette étape.
- **Gestion des erreurs et surveillance dans un environnement de production:** En production, la gestion des erreurs est cruciale. Les frameworks Back End permettent de configurer des journaux d'erreurs et des mécanismes de surveillance pour garantir la stabilité de l'application en temps réel.