

# **COURS WEB ET MOBILE AVANCEE**

**Ousman GNALY**

**Universite Islamique au Niger**

*HTTP*

# **Introduction au Protocole HTTP et son Rôle dans Notre Architecture**

**Le protocole HTTP (HyperText Transfer Protocol)** est la pierre angulaire du web moderne. Il s'agit d'un protocole de communication qui permet aux clients (généralement des navigateurs web) d'envoyer des requêtes à des serveurs, qui à leur tour renvoient des réponses.

## **Rôle de HTTP dans Notre Architecture**

Dans le contexte de notre architecture backend composée de routers, controllers, services et DAO, HTTP joue un rôle crucial :

1. **Router (Routeur)** : Chaque route définie dans le routeur correspond à un endpoint HTTP spécifique. Le routeur écoute les requêtes HTTP entrantes et les dirige vers le contrôleur approprié en fonction de l'URL et de la méthode HTTP utilisée.

**2. Controller (Contrôleur)** : Le contrôleur reçoit la requête HTTP du routeur, traite les données entrantes, et renvoie une réponse HTTP au client après avoir interagi avec les services et DAO.

3. **Service & DAO** : Bien que ces couches n'interagissent pas directement avec HTTP, elles traitent la logique métier et les opérations de base de données déclenchées par les requêtes HTTP.



## **Importance de Comprendre HTTP**

Comprendre le fonctionnement de HTTP est essentiel pour plusieurs raisons :

- **Sécurité** : Connaître les différentes méthodes HTTP et leurs implications peut aider à sécuriser votre application contre des attaques courantes.

- **Performance** : Une bonne compréhension de HTTP peut aider à optimiser les temps de réponse et à améliorer l'expérience utilisateur.

- **Débogage** : Lorsque des problèmes surviennent, une connaissance solide de HTTP est inestimable pour déboguer et résoudre les problèmes rapidement.

Bref , HTTP est bien plus qu'un simple protocole de communication. C'est le mécanisme qui alimente la communication entre le client et le serveur dans notre architecture, et une compréhension approfondie de son fonctionnement est essentielle pour tout développeur backend.

# **Les Méthodes HTTP : Une Exploration Approfondie**

Après avoir compris le rôle fondamental du protocole HTTP dans notre architecture, il est essentiel de se pencher sur les méthodes HTTP spécifiques qui définissent le type d'opération à effectuer.

# GET

La méthode **GET** est utilisée pour récupérer des informations à partir d'un serveur. Dans le contexte de notre architecture, une requête GET pourrait être utilisée pour récupérer un utilisateur ou une liste d'utilisateurs.

**Exemple :** GET /users pourrait récupérer une liste de tous les utilisateurs.



# POST

**POST** est utilisé pour envoyer des données à un serveur afin de créer une nouvelle ressource. C'est la méthode que vous utiliseriez si vous vouliez ajouter un nouvel utilisateur à votre base de données.

**Exemple :** `POST /users` avec un corps de requête contenant les détails d'un utilisateur pourrait créer un nouvel utilisateur.

# *PUT*

La méthode **PUT** est utilisée pour mettre à jour une ressource existante sur le serveur. Si vous aviez un utilisateur existant et que vous vouliez mettre à jour ses détails, vous utiliseriez PUT.

**Exemple :** `PUT /users/:id` avec un corps de requête contenant les nouveaux détails de l'utilisateur mettrait à jour l'utilisateur spécifié.

# ***DELETE***

Comme son nom l'indique, **DELETE** est utilisé pour supprimer une ressource sur le serveur.

**Exemple :** `DELETE /users/:id` supprimerait l'utilisateur avec l'ID spécifié.

## *Autres méthodes*

Il existe d'autres méthodes HTTP comme **PATCH** (pour les mises à jour partielles), **HEAD** (similaire à GET mais sans corps de réponse), et **OPTIONS** (utilisé pour décrire les options de communication pour la ressource cible), mais les quatre méthodes ci-dessus sont les plus couramment utilisées dans la plupart des applications web.

# Conclusion

Les méthodes HTTP définissent l'action à effectuer sur la ressource spécifiée. En comprenant chaque méthode et son utilisation appropriée, vous pouvez garantir que votre application est à la fois fonctionnelle et conforme aux normes de l'industrie.