Une image contenant capture d’écran, Graphique, Police, graphisme

Description générée automatiquement

|  |
| --- |
| **CER U.E. [Mission Apache]** |
| **[Rami Mohamed Amine]** |

Animateur : Lamine

Scribe : Ana

Gestionnaire : Mimou

Secrétaire : Rami

00-00-2025 A2 – Groupe 01

# Mots clés :

# Environnement de développement local

# Mots à définir :

# Server apache : : Apache HTTP Server est un serveur web open-source qui permet d'héberger et de servir des sites internet. Il est largement utilisé pour sa flexibilité, sa compatibilité avec plusieurs systèmes d'exploitation et son extensibilité grâce à de nombreux modules.

# Document root d’apache : C'est le répertoire où sont stockés les fichiers d'un site web servi par Apache. Par défaut, sous Linux, il se trouve généralement dans /var/www/html, mais il peut être modifié dans la configuration d’Apache.

# Répertoire racine : En informatique, le répertoire racine est le premier niveau d’un système de fichiers. Dans le contexte d’Apache, il peut faire référence au **Document Root**, c'est-à-dire le dossier principal contenant les fichiers accessibles via le serveur web.

# VHost : Un hôte virtuel permet à Apache d’héberger plusieurs sites web sur un même serveur en les distinguant par des noms de domaine ou des adresses IP différentes. Chaque site peut avoir sa propre configuration.

# Logs apache : Ce sont les fichiers journaux générés par Apache, contenant des informations sur les requêtes des utilisateurs, les erreurs rencontrées et d'autres événements du serveur. Ils sont généralement stockés dans /var/log/apache2/ sous Linux et peuvent inclure des fichiers comme access.log et error.log.

# Redirection : Il s'agit du processus consistant à renvoyer les utilisateurs d’une URL vers une autre, soit pour des raisons de maintenance, soit pour rediriger le trafic vers une nouvelle adresse. Cela peut être fait via Apache en utilisant .htaccess ou la configuration du serveur.

# Fichier.htaccess : Un fichier de configuration utilisé par Apache pour appliquer des directives spécifiques à un répertoire. Il permet de gérer les redirections, la réécriture d’URL, les permissions d’accès et d’autres paramètres sans modifier la configuration principale du serveur.

1. **Analyse du contexte :**

* **L’équipe s’engage à suivre ces directives pour livrer un serveur Apache local configuré avec plusieurs hôtes virtuels, garantissant sécurité et performance pour le projet du CESI.**

# Définition de la problématique :

# Comment assurer une configuration sécurisée et performante d’un serveur Apache local avec des hôtes virtuels, tout en exploitant les logs pour le suivi des requêtes et erreurs ?

1. **Les contraintes :**

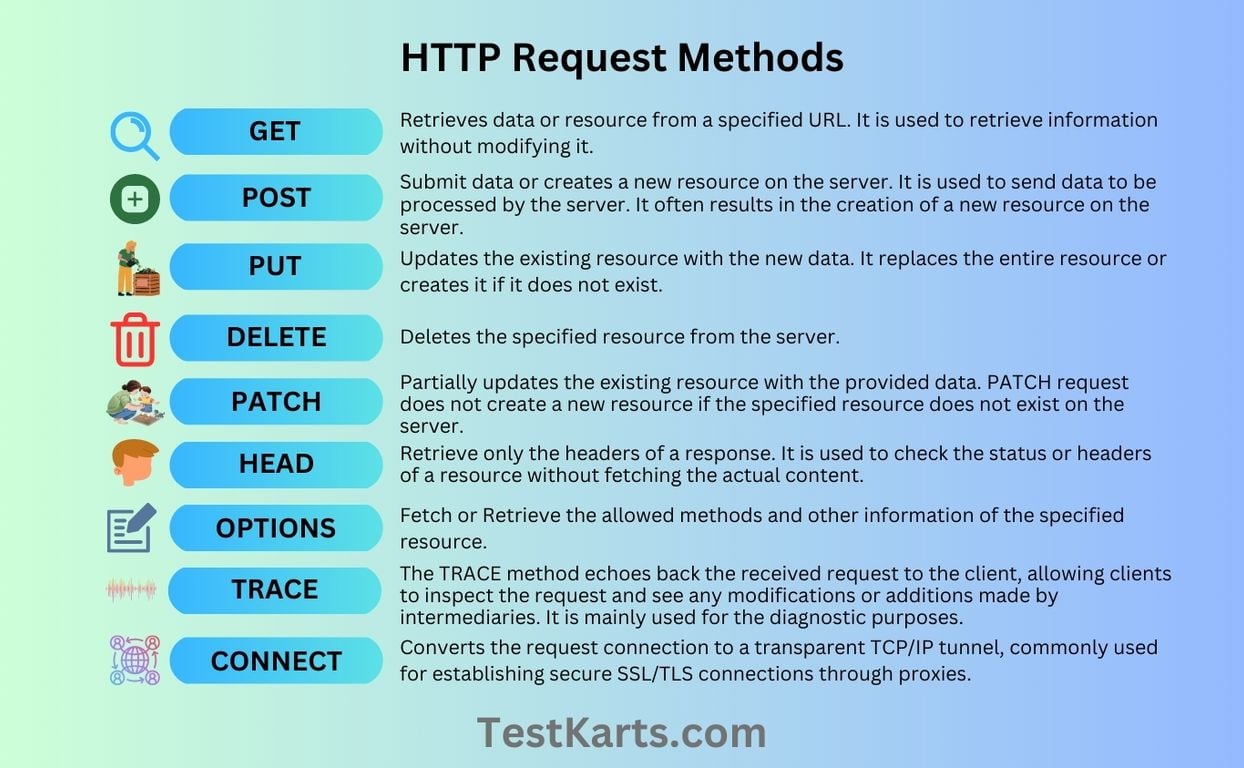
* **Temps de 2 jours**

# Plan d’actions

Plan d’action 1 : effectuer des recherches sur http (méthode, code d’état, entête) :

Le ****protocole HTTP (Hypertext Transfer Protocol)**** est un protocole de communication utilisé pour transférer des données sur le Web. Il fonctionne selon un modèle client-serveur et repose sur un échange de requêtes et de réponses.

Les méthodes HTTP indiquent l'action à effectuer sur une ressource. Voici les plus courantes :



## **Les Codes d’État http :**

Les codes d’état HTTP indiquent le résultat d’une requête. Ils sont classés en 5 catégories :

### **1xx : Informations**

* 100 Continue → Le serveur a reçu une partie de la requête et attend la suite.
* 101 Switching Protocols → Changement de protocole demandé (ex: HTTP vers WebSocket).

### **2xx : Succès**

* 200 OK → La requête a réussi (réponse avec du contenu).
* 201 Created → Une ressource a été créée avec succès.
* 204 No Content → La requête a réussi, mais sans retour de contenu.

### **3xx : Redirections**

* 301 Moved Permanently → La ressource a changé d'URL de façon permanente.
* 302 Found → La ressource est temporairement déplacée.
* 304 Not Modified → Le client peut utiliser une version en cache.

### **4xx : Erreurs côté client**

* 400 Bad Request → Mauvaise syntaxe de requête.
* 401 Unauthorized → Authentification requise.
* 403 Forbidden → Accès interdit.
* 404 Not Found → Ressource non trouvée.

### **5xx : Erreurs côté serveur**

* 500 Internal Server Error → Erreur interne du serveur.
* 502 Bad Gateway → Problème de communication entre serveurs.
* 503 Service Unavailable → Serveur indisponible (surcharge, maintenance).

## **Les En-têtes http :**

Les **en-têtes HTTP** (headers) fournissent des informations supplémentaires dans les requêtes et réponses.

### **En-têtes de requête (Request Headers) :**

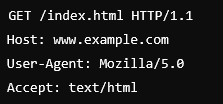
* User-Agent → Identifie le client (navigateur, bot, etc.).
* Accept → Indique les formats de réponse acceptés (text/html, application/json).
* Authorization → Envoie un jeton d'authentification.
* Cookie → Envoie les cookies du client au serveur.

### **En-têtes de réponse (Response Headers) :**

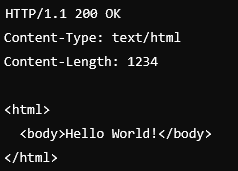
* Content-Type → Indique le format de la réponse (text/html, application/json).
* Set-Cookie → Définit un cookie sur le client.
* Cache-Control → Contrôle la mise en cache (no-cache, max-age=3600).
* Location → Spécifie une URL pour rediriger (301, 302).

**Exemple d'échange HTTP :**

**Requête HTTP :**

****

**Réponse HTTP :**

****

Plan d’action 2 : configurer le ficher racine :

Le **fichier racine** (aussi appelé **document root**) est le répertoire où sont stockés les fichiers accessibles publiquement par le serveur web. Par défaut :

* **Apache** → /var/www/html

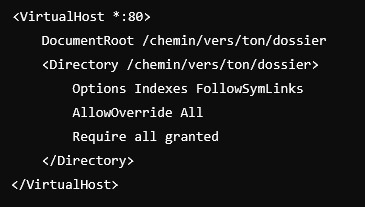
## **Configurer le fichier racine sur Apache**

### 📌 **Étape 1 : Modifier la configuration du VirtualHost**

* Ouvre le fichier de configuration du site :

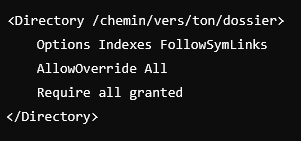


Modifie la ligne DocumentRoot :



Modifier apache2.conf :





Appliquer les modifications :



Redémarre Apache :



Plan d’action 3 : crée configurer les VHosts

### **Créer un nouveau fichier de configuration :**

Par défaut, Apache stocke les VHosts dans **/etc/apache2/sites-available/**.  
Créer un fichier pour ton site, par exemple **monsite.conf**



On ajoute le contenu suivant en adaptant le **nom de domaine** et le **chemin du dossier racine** :



Activer le VHost et redémarrer Apache :





Tester avec un nom de domaine local.

Plan d’action 5 : tester les requêtes http sur un server

Pour tester les requêtes HTTP sur un serveur web (Apache, Nginx, etc.), tu peux utiliser plusieurs outils : ****cURL, wget, Postman, HTTPie**** et des outils intégrés comme ****telnet ou netcat****.

cURL est un outil puissant pour tester les requêtes HTTP.



**Explication** :

* -I : Affiche uniquement l’en-tête HTTP de la réponse.
* Cela permet de voir le **code de statut HTTP** (200, 404, 500, etc.).

****Envoyer une requête POST :****



Plan d’action 4 : implémenté https et configurez .htaccess

Pour activer HTTPS sur votre site, vous devez obtenir un certificat SSL et configurer votre serveur pour l'utiliser.

#### **1.1 Obtenir un certificat SSL**

* **Let’s Encrypt** (Gratuit) : Vous pouvez utiliser Let’s Encrypt avec Certbot.
* **Certificats payants** : Acheter un certificat auprès d’un fournisseur (GoDaddy, DigiCert, etc.).

#### **1.2 Installer Let’s Encrypt avec Certbot**

Sur un serveur **Apache** sous Linux :

****

**Générer un certificat SSL** :

****

Suivez les instructions pour sélectionner votre domaine et activer la redirection HTTP → HTTPS.

#### **1.3 Renouvellement automatique**

Let’s Encrypt expire tous les 90 jours, alors ajoutez un cron job :

****

Ajoutez cette ligne pour renouveler automatiquement :

****

### Configurer .htaccess pour rediriger HTTP vers HTTPS

Dans Apache, vous pouvez forcer HTTPS en ajoutant ces règles dans .htaccess (situé à la racine du site) :

****

#### **Forcer www ou non-www**

Si vous voulez que toutes les requêtes passent par www.domaine.com :

****

Ou pour forcer sans-www :

****