

# Roadmap Webserv

## Phase 1 - Mise en place du squelette

- Créer ton Makefile (compilation basique).
- Organiser ton projet (src/, include/, conf/).
- Préparer une structure de base avec Server, Client, Request, Response, ConfigParser.
- Ajouter gestion des erreurs globales -> ton serveur ne doit jamais crasher.

## Phase 2 - Socket & Poll

- Créer un serveur simple qui écoute sur un port (socket, bind, listen).
- Accepter une connexion et répondre Hello World (hardcodé).
- Passer en non-bloquant (fcntl O\_NONBLOCK).
- Implémenter la boucle poll() unique qui gère plusieurs clients.
- Vérifier que plusieurs navigateurs/curl peuvent se connecter simultanément.

## Phase 3 - Parsing HTTP (Request)

- Lire une requête ligne par ligne (method, uri, version).
- Extraire les headers (Content-Length, Host, ...).
- Lire le body selon Content-Length ou chunked.
- Stocker la requête dans une structure Request.
- Gérer les erreurs de parsing (400 Bad Request).

## Phase 4 - Réponse simple

- Créer une classe Response.
- Implémenter en dur : 200 OK, 404 Not Found.
- Vérifier affichage correct dans un navigateur et avec curl.
- Générer des error pages par défaut si besoin.

## Phase 5 - Méthodes HTTP

- GET : renvoyer un fichier statique depuis un répertoire racine.
- DELETE : supprimer un fichier si présent.
- POST : écrire un fichier uploadé (dans un dossier défini).
- Retourner 405 Method Not Allowed si la méthode n'est pas autorisée.

## **Phase 6 - Configuration file**

- Écrire un parser minimal pour : listen, root, index.
- Étendre avec : error\_page, client\_max\_body\_size, methods, autoindex, upload\_store, cgi\_pass.
- Gérer plusieurs serveurs virtuels (multi-port).

## **Phase 7 - CGI**

- Lancer un script avec fork() + execve().
- Passer la requête dans stdin du CGI.
- Lire la sortie du CGI sur stdout.
- Gérer EOF si pas de Content-Length.
- Retour correct des headers CGI (Content-Type, Content-Length).
- Tester avec php-cgi ou un script Python simple.

## **Phase 8 - Robustesse**

- Vérifier gestion des timeouts (aucune requête ne doit bloquer).
- Stress tests avec ab -n 1000 -c 100 http://127.0.0.1:8080/.
- Vérifier compatibilité avec nginx sur les headers.
- Vérifier avec valgrind -> zéro fuite mémoire.
- Tester déconnexions brutales (fermeture navigateur).

## **Phase 9 - Bonus**

- Ajouter cookies & sessions simples.
- Gérer plusieurs types de CGI (PHP, Python, etc.).
- Ajouter virtual hosts avec server\_name.