

Développement Web

TP n° 5 : Serveur statique

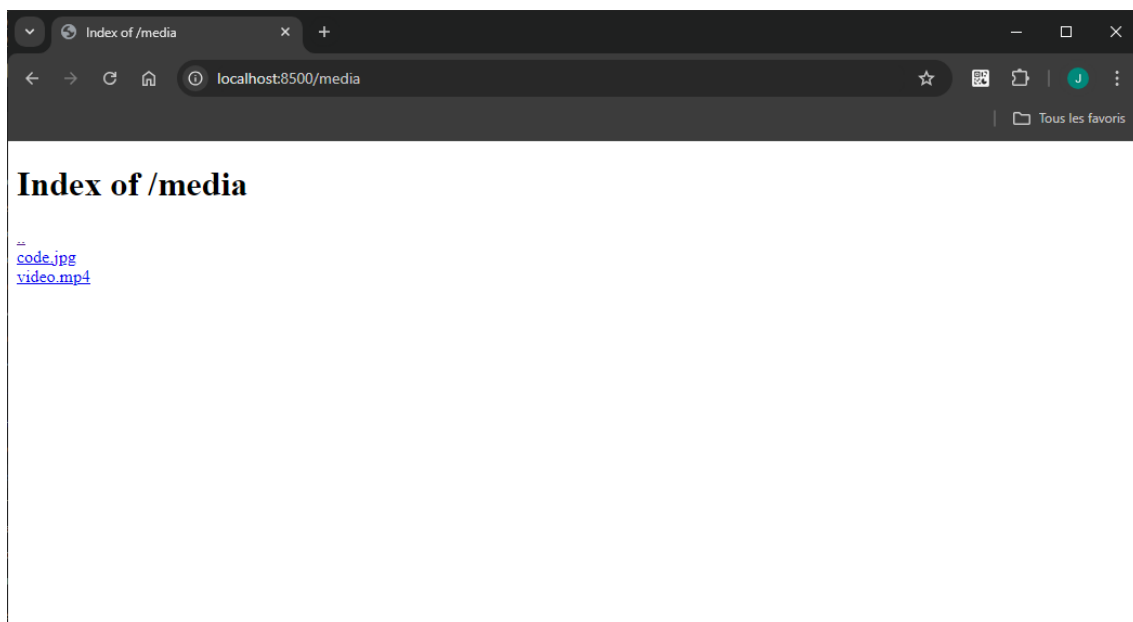
Le but de ce TP est de créer un serveur de fichier statique. Celui-ci permet de rendre disponible en ligne toute une arborescence de fichiers.

Contexte

On souhaite donner accès à un dossier du disque dur via un serveur. Le serveur accepte des requêtes **GET** (uniquement) et transpose l'URL requise en chemin local. Il répond ensuite en transmettant la ressource demandée.

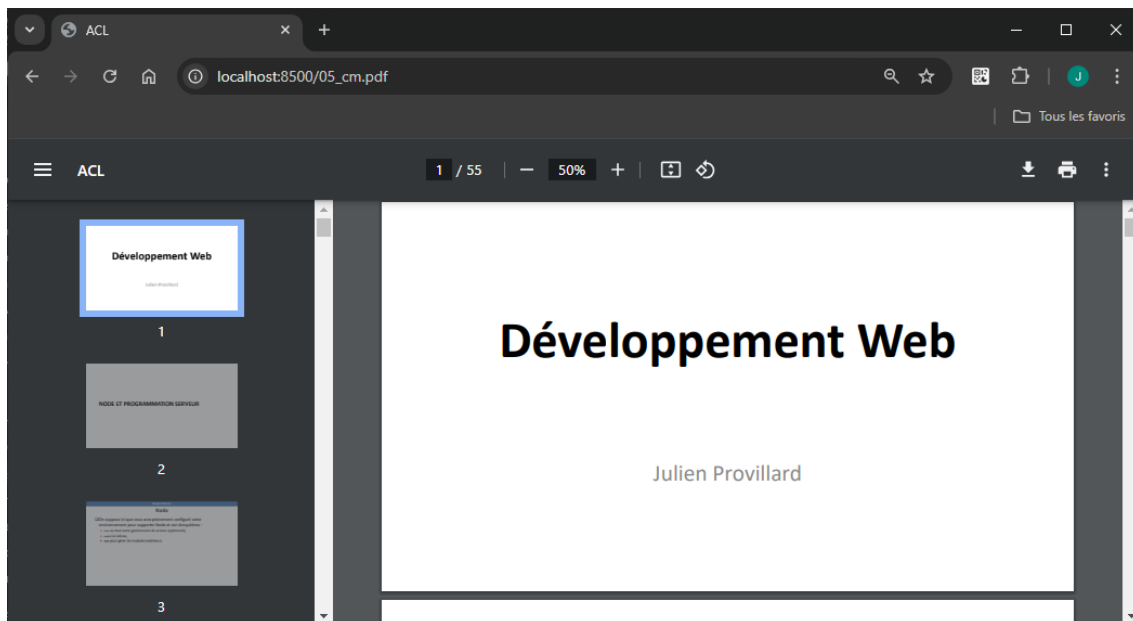
On suppose ici que le serveur écoute l'adresse **localhost:8500** et donne accès au répertoire local **public**. Pour accéder à un fichier **resource.ext** situé dans le répertoire **public/relative/path/**, il faut effectuer une requête **GET** sur le chemin **/relative/path/ressource.ext**.¹

Si le chemin demandé correspond à un dossier, son contenu est listé dans une page html avec un lien vers son répertoire parent, ses sous-répertoires et ses fichiers.



Si le chemin demandé est un fichier, le navigateur déclenche son action par défaut pour ce type de fichier.

1. Par exemple en entrant l'adresse <http://localhost:8500/relative/path/ressource.ext> dans un navigateur



Si le chemin n'existe pas, le serveur doit répondre avec le statut 404.

Si la méthode utilisée pour la requête n'est pas **GET**, le serveur doit répondre avec le statut 405.

Si le serveur rencontre une erreur (ce sera probablement le cas au cours du développement), il est préférable de rattraper l'erreur sans arrêter le serveur et de répondre avec le statut 500.

Tests

Pour que vous puissiez tester le serveur, un dossier **public** est disponible sur Arche. Servez-le et vérifiez que toutes les URL valides sont bien accessibles. Adaptez votre code au besoin.

Aide technique

Module **fs**

- La fonction **stat** analyse un chemin d'accès. Elle permet notamment de déterminer si le chemin correspond à un répertoire ou à un fichier.
- La fonction **readdir** renvoie un itérateur sur les fichiers d'un dossier.
- Ces deux fonctions renvoient une erreur si le chemin d'accès est invalide. Cette erreur a un champ **code** de valeur **"ENOENT"**. Elles sont également disponibles dans le module **fs/promises**.
- Pour envoyer un fichier vers le client, il faut ouvrir un flux sur celui-ci avec la fonction **createReadStream**. Ce flux doit ensuite être passée à la méthode **pipe** de la réponse. Il ne faudra pas utiliser **write** et **end** dans ce cas (**pipe** les gère automatiquement).

Module **path**

- Pour concaténer proprement des chemins, utilisez la fonction **join** (concaténation) ou **resolve** (concaténation + normalisation).

Module **mime**

- Lors de l'envoi d'un fichier dans la réponse (via un flux), il faut indiquer le type de la ressource dans l'entête. Il est possible de déterminer ce type à partir de l'extension du fichier en utilisant le module **mime**. Ce n'est pas un module standard, il vous faudra passer par **npm** pour l'installer.