INCLUSION DE FICHIER

Signification de ./../ dans cette ligne :

php

require './../vendor/autoload.php';

En PHP, lorsque vous incluez ou requérez un fichier, le chemin spécifié est relatif au fichier courant. Dans ce cas, ./../ est une notation spéciale qui indique de remonter d'un niveau dans l'arborescence des répertoires. Voici la décomposition de cette notation :

* . représente le répertoire courant.
* .. représente le répertoire parent (le répertoire qui contient le répertoire courant).
* / est le séparateur de répertoires (utilisé sur les systèmes Unix/Linux/macOS, tandis que sur Windows, on utilise \).

Ainsi, ./../ signifie "remonter d'un niveau dans l'arborescence des répertoires par rapport au répertoire courant". Dans le contexte de require './../vendor/autoload.php';, cela indique que le fichier autoload.php se trouve dans le répertoire vendor, qui lui-même se trouve un niveau au-dessus du répertoire courant. Cette notation est couramment utilisée lorsque vous travaillez avec des frameworks ou des bibliothèques tierces qui sont installées dans un répertoire spécifique (souvent vendor) à l'extérieur du répertoire de votre application. Par exemple, si votre structure de répertoires ressemble à ceci :

- projet/

- app/

- monFichier.php (fichier courant)

- vendor/

- autoload.php

Alors, depuis monFichier.php dans le répertoire app, vous devez remonter d'un niveau (..) pour atteindre le répertoire projet, puis entrer dans le répertoire vendor pour trouver le fichier autoload.php. Cette notation de chemin relatif avec ./../ est couramment utilisée pour naviguer entre les répertoires de manière flexible, sans avoir à spécifier de chemin absolu complet.

L'inclusion de fichiers permet d'incorporer le contenu d'un fichier PHP externe dans un autre fichier PHP. Cela facilite la réutilisation de code et la modularité dans les applications web. Il existe principalement deux fonctions pour inclure des fichiers en PHP :

1. include()
2. require()

La différence principale est que require() génère une erreur fatale si le fichier n'est pas trouvé, tandis que include() génère seulement un avertissement. Voici quelques points clés sur l'inclusion de fichiers en PHP :

* Les chemins des fichiers à inclure peuvent être spécifiés de manière relative (ex: './fichier.php') ou absolue (ex: '/var/www/fichier.php').
* La notation '../' permet de remonter d'un niveau dans l'arborescence des répertoires.
* include() et require() vérifient d'abord le include\_path défini, puis le répertoire du script appelant, puis le répertoire de travail courant.
* On peut inclure des fichiers distants via HTTP si les gestionnaires d'URL sont activés.
* include\_once() et require\_once() permettent d'éviter les redéclarations de fonctions/classes en n'incluant un fichier qu'une seule fois.
* La sortie d'un fichier inclus peut être capturée dans une variable en utilisant ob\_start() et ob\_get\_clean().

Donc en résumé, l'inclusion de fichiers en PHP, que vous appelez "navigation de fichiers", permet de séparer le code en différents fichiers modulaires et de les incorporer où nécessaire, facilitant ainsi la réutilisation et la maintenance du code.