Comment désactiver les messages d’erreur PHP

Panneau d’hébergement · https://kyzoe.be/fr/docs/comment-desactiver-les-messages-d-erreur-php/

Dans ce guide, vous apprendrez à masquer ou supprimer les erreurs et notifications PHP (comme les fréquents avertissements PHP « deprecated »).

Désactiver les messages d’erreur PHP

- Connectez-vous sur https://cp.vcloudhosting.eu
- Ouvrez **Sites web** dans la barre latérale gauche
- Sélectionnez le site web dont vous souhaitez modifier le document root
- Choisissez **Avancé** dans le menu du haut et sélectionnez le domaine
- Choisissez **Outils de développement**
- Faites défiler jusqu’à **php.ini editor** et cliquez sur **Ajouter une directive**
- Saisissez la directive **error_reporting** souhaitée

Les paramètres d’exécution PHP définis dans votre propre code peuvent écraser les paramètres ci-dessus spécifiés dans **php.ini**. Recherchez **ini_set** ou **error_reporting** dans vos scripts PHP pour voir où cela se produit.

Vous pouvez éventuellement aussi utiliser une version de PHP plus ancienne qui ne considère pas comme obsolètes les fonctions dont dépend votre code. Cela permet parfois de résoudre le problème sans modifier les scripts.

Utilisation de error reporting

Voici un aperçu des directives que vous pouvez modifier :

- **error_reporting** : détermine quelles erreurs et quels avertissements doivent être signalés. Réglez-la sur 0 pour tout désactiver, mais cela est déconseillé en production. Pour le développement, vous pouvez par exemple utiliser E_ALL & ~E_NOTICE.
- **display_errors** : détermine si les erreurs sont affichées à l’utilisateur (dans le navigateur). Réglez-la sur Off pour ne pas afficher les erreurs.

- **log_errors** : si vous souhaitez journaliser les erreurs plutôt que les afficher, réglez-la sur On.
- **error_log** : vous pouvez indiquer ici le chemin d'un fichier journal pour l'enregistrement des erreurs.

Masques de bits pour error reporting

- **E_ALL** : signale toutes les erreurs et tous les avertissements PHP.
- **E_ERROR** : signale uniquement les erreurs fatales.
- **E_WARNING** : signale les avertissements d'exécution (non fatals).
- **E_NOTICE** : signale les notifications d'exécution.
- **E_PARSE** : signale les erreurs (de syntaxe) lors de l'analyse.
- **E_DEPRECATED** : signale les fonctions obsolètes.
- **E_STRICT** : donne des recommandations pour une meilleure compatibilité du code.

Vous pouvez combiner ces niveaux d'erreur avec les opérateurs bit à bit AND (&), OR (|) et NOT (~).

Exemple : pour afficher toutes les erreurs sauf les notices et les normes strictes :
`error_reporting = E_ALL & ~E_NOTICE & ~E_STRICT`