

Control includes

El control d'inclusions és una pràctica de seguretat fonamental per evitar que els usuaris accedeixin directament a fitxers PHP que no han estat dissenyats per ser executats per si mateixos. Aquesta tècnica garanteix que el codi s'executi només en el context adequat, prevenint l'exposició de dades sensibles o la manipulació del sistema.

Mètode de la constant de control

Un mètode és utilitzar una constant de control, és una de les maneres més simples i efectives de fer un control d'inclusions. La lògica és la següent:

1. **Arxiu principal:** Crea una constant abans de qualsevol crida a `include` o `require`. Aquesta constant servirà de "bandera" o "clau d'accés".
2. **Arxiu inclòs:** Al principi de l'arxiu que serà inclòs, comprova si aquesta constant existeix. Si no existeix, significa que el fitxer s'està accedint directament, per tant, detens l'execució.

Això assegura que el codi del fitxer inclòs només s'executarà quan sigui cridat des de l'script principal que ha definit la constant.

Exemple pràctic

1. Arxiu principal (`index.php`)

Aquest és el punt d'entrada a l'aplicació. Abans d'incloure qualsevol altre fitxer, es defineix una constant que servirà com a comprovació de seguretat.

```
<?php
// Defineix una constant per controlar les inclusions
// en algunes ocasions es defineixen constants amb el nom
// de la app __NEXT_CLOUD__ o __WORDPRESS__
define('__ACCES_PERMES__', true);

// Ara pots incloure els altres arxius de forma segura
require_once 'config.php';
include 'utilitats.php';

// La resta del teu codi principal
echo "Benvingut al lloc web!";
?>
```

2. Arxiu de configuració (`config.php`)

Aquest fitxer conté informació sensible i no hauria de ser accessible directament per un visitant.

```
<?php
// Comprova si la constant existeix abans de continuar
if (!defined('__ACCES_PERMES__')) {
    // Si la constant no existeix, l'accés és denegat
    // Es pot redirigir l'usuari, mostrar un missatge d'error, etc.
    // En aquest cas, simplement aturem l'execució.
    die('Accés denegat!');
}

// Les dades sensibles de configuració de la base de dades
define('DB_USER', 'el_teu_usuari');
define('DB_PASS', 'la_teva_contrasenya');
```

3. Arxiu d'utilitats (utilitats.php)

El mateix principi s'aplica a altres fitxers inclosos que contenen funcions o classes.

```
<?php
// Comprova la constant de control
// if (!defined('__ACCES_PERMES__')) die('Accés denegat!');

defined('__ACCES_PERMES__') or die('Accés denegat!');

function saludar($nom) {
    return "Hola, " . $nom . "!";
}

echo "El fitxer d'utilitats s'ha carregat correctament.<br>";
```

Si un usuari intenta accedir directament a `config.php` des del seu navegador (per exemple, `http://localhost/config.php`), el script comprovarà si la constant `__ACCES_PERMES__` existeix. Com que no s'ha definit, el codi s'aturarà amb el missatge "Accés denegat!". Aquest mètode és una mesura de seguretat efectiva per protegir la teva aplicació de possibles vulnerabilitats.

🕒 Revisió núm. 1

★ Admin l'ha creat 2025-09-23 13:17:51 UTC

✎ Admin l'ha actualitzat 2025-10-13 15:38:16 UTC