<?php

// Vérification du serveur SQL

$host = 'localhost';

$user = 'nom\_utilisateur';

$password = 'mot\_de\_passe';

$database = 'nom\_base\_de\_donnees';

$mysqli = new mysqli($host, $user, $password, $database);

if ($mysqli->connect\_errno) {

    echo "Erreur de connexion au serveur SQL : " . $mysqli->connect\_error;

} else {

    echo "Connexion au serveur SQL réussie !";

    $mysqli->close();

}

// Vérification du serveur Apache

$url = 'http://localhost';

$ch = curl\_init($url);

curl\_setopt($ch, CURLOPT\_RETURNTRANSFER, true);

$response = curl\_exec($ch);

$httpCode = curl\_getinfo($ch, CURLINFO\_HTTP\_CODE);

curl\_close($ch);

if ($httpCode == 200) {

    echo "Le serveur Apache est accessible !";

} else {

    echo "Erreur de connexion au serveur Apache : Code HTTP " . $httpCode;

}

?>

Ce script PHP tente de se connecter au serveur SQL en utilisant les informations fournies, puis vérifie la connectivité avec le serveur Apache en effectuant une requête HTTP vers l'URL spécifiée. Les messages de réussite ou d'erreur seront affichés en fonction des résultats.

Cette ligne crée un nouvel objet de la classe `mysqli` en utilisant les paramètres de connexion spécifiés. Cet objet sera utilisé pour interagir avec le serveur SQL.

La méthode `connect\_errno` de l'objet `mysqli` vérifie s'il y a eu une erreur lors de la connexion. Si une erreur se produit, cela signifie que la connexion a échoué, et le message d'erreur est affiché à l'écran en utilisant la méthode `connect\_error`.

Si la connexion au serveur SQL réussit, le bloc `else` est exécuté, et le message "Connexion au serveur SQL réussie !" est affiché à l'écran. Ensuite, la méthode `close()` est appelée sur l'objet `mysqli` pour fermer la connexion avec le serveur SQL.

`curl\_setopt($ch, CURLOPT\_RETURNTRANSFER, true);` : Cette ligne configure l'option `CURLOPT\_RETURNTRANSFER` de l'objet cURL (`$ch`) à `true`. Cela indique à cURL de renvoyer la réponse de la requête sous la forme d'une chaîne de caractères plutôt que de l'afficher directement à l'écran. En configurant cette option à `true`, vous pouvez stocker la réponse dans une variable et l'utiliser ultérieurement dans votre code.

`$response = curl\_exec($ch);` : Cette ligne exécute la requête HTTP en utilisant l'objet cURL (`$ch`). La fonction `curl\_exec()` envoie la requête et renvoie la réponse du serveur. Dans ce cas, la réponse est stockée dans la variable `$response` pour une utilisation ultérieure.

`$httpCode = curl\_getinfo($ch, CURLINFO\_HTTP\_CODE);` : Cette ligne utilise la fonction `curl\_getinfo()` pour obtenir des informations sur la requête HTTP qui vient d'être exécutée. En passant l'option `CURLINFO\_HTTP\_CODE` à cette fonction, on obtient le code de statut HTTP de la réponse. Cette valeur est ensuite stockée dans la variable `$httpCode` pour une utilisation ultérieure.

En utilisant ces lignes de code, vous pouvez envoyer une requête HTTP vers un serveur spécifié, récupérer la réponse et obtenir le code de statut HTTP correspondant. Cela vous permet de vérifier si la requête a réussi (statut 200) ou si une erreur s'est produite (autre code de statut).

La ligne de code `if ($httpCode == 200)` est une instruction conditionnelle en PHP. Elle vérifie si la variable `$httpCode` est égale à 200.

Dans ce contexte, la variable `$httpCode` contient le code de statut HTTP retourné par la requête effectuée vers le serveur Apache. Le code de statut HTTP 200 signifie "OK" et indique que la requête a été traitée avec succès.

Donc, la condition `if ($httpCode == 200)` vérifie si la requête vers le serveur Apache a renvoyé un code de statut HTTP 200, ce qui signifie que le serveur Apache est accessible et fonctionnel. Si la condition est vraie, le code à l'intérieur du bloc `if` sera exécuté.

Vous pouvez personnaliser le comportement en fonction du code de statut retourné. Par exemple, vous pouvez ajouter des actions supplémentaires pour d'autres codes de statut, tels que la redirection en cas de code 301 ou l'affichage d'un message d'erreur pour les codes d'erreur 4xx ou 5xx.

Il est important de noter que le code de statut HTTP 200 est spécifique à une réponse HTTP réussie. Si vous souhaitez effectuer des vérifications plus détaillées de la fonctionnalité du serveur Apache, vous devrez utiliser des méthodes plus avancées, telles que l'envoi de requêtes spécifiques et l'analyse des réponses.