GR

PHP: usare cURL con le API

In questo tutorial vedremo come effettuare delle chiamate a delle API remote con PHP e cURL.

Il nostro scopo è quello di ottenere una descrizione delle previsioni meteo di una città. Useremo OpenWeatherMap come API e l'estensione cURL per effettuare le richieste HTTP.

In tal senso creeremo una funzione di utility.

```
define('API_KEY', 'chiave API');
define('API_URL',
'https://api.openweathermap.org/data/2.5/weather');

function get_weather_data($city) {

    if(empty($city)) {
        return null;
    }

    $url = API_URL . '?q=' . urlencode($city) .
    '&appid=' . API_KEY;
    $ch = curl_init();

    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
    curl_setopt($ch, CURLOPT_FOLLOWLOCATION, true);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($ch, CURLOPT_URL, $url);
    curl_setopt($ch, CURLOPT_TIMEOUT, 10);
```

```

$response = curl_exec($ch);

if(curl_error($ch)) {
    return null;
}

$data = json_decode($response, true);

if(intval($data['cod']) !== 200) {
    return null;
}

$weather = $data['weather'];

return $weather[0]['description'];
}

```

La prima verifica è sul parametro passato, che non può essere vuoto. Quindi creiamo l'URL completo della query string da passare a cURL. Non sappiamo se le API remote dispongano di un certificato SSL valido, quindi impostiamo `CURLOPT_SSL_VERIFYPEER` su `false` per impedire la verifica del certificato SSL. Un'altro dettaglio che non conosciamo è se le API restituiranno direttamente i risultati o se effettueranno un redirect HTTP, quindi impostiamo `CURLOPT_FOLLOWLOCATION` su `true`.

Quindi istruiamo cURL a restituire il risultato con l'opzione `CURLOPT_RETURNTRANSFER` e impostiamo l'URL appena creato con `CURLOPT_URL`. Infine per concludere il setup di cURL impostiamo il timeout della connessione in secondi con l'opzione `CURLOPT_TIMEOUT`.

A questo punto se cURL non restituisce errori e se le API remote restituiscono un codice di stato uguale a 200, trasformiamo la stringa JSON ottenuta in un array associativo e prendiamo l'informazione richiesta.

Possiamo quindi creare la pagina di ricerca in questo modo:

```
<?php
require_once 'functions.php';
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <title>PHP: API example</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-
width, initial-scale=1">
    <link rel="stylesheet"
href="assets/css/style.css" type="text/css"
media="screen">
</head>
<body>
    <main>
        <form action="index.php" method="post">
            <div class="form-group">
                <label for="city">City</label>
                <input type="text" name="city"
id="city" class="form-control">
            </div>
            <p>
                <input type="submit" class="btn btn-
outline-primary" value="Get weather info">
            </p>
        </form>
        <?php if(isset($_POST['city'])) {
            $weather_data =
get_weather_data($_POST['city']);
            if(is_null($weather_data)) {
                ?>
```

```
        <div class="alert alert-danger mt-4">Error.</div>
        <?php
            } else {
        ?>
        <div class="alert alert-info mt-4">
            <?php echo $weather_data; ?>
        </div>
        <?php
            }
        }
        ?>
    </main>
</body>
</html>
```

La procedura mostrata è simile anche per tutte le altre API, fermo restando il fatto che potrebbe essere necessario modificare le impostazioni di cURL se le API hanno esigenze particolari (ad esempio speciali header HTTP).