

# GABRIELE ROMANATO

Menu

## Effettuare richieste HTTP con le funzioni core di WordPress

WordPress, la piattaforma di gestione dei contenuti più popolare al mondo, offre una vasta gamma di funzionalità per sviluppatori, inclusa la possibilità di effettuare richieste HTTP. Questo è particolarmente utile quando si devono integrare servizi esterni, recuperare dati da API o inviare informazioni ad altre applicazioni. In questo articolo, esploreremo come effettuare richieste HTTP utilizzando le funzioni core di WordPress, concentrandoci principalmente su `wp_remote_get` e `wp_remote_post`.

WordPress include una serie di funzioni per gestire le richieste HTTP, che sono costruite sopra la libreria di richieste HTTP di WordPress, basata su PHP cURL o Streams (a seconda di cosa è disponibile nel server). Le due funzioni principali sono:

- `wp_remote_get()`: Utilizzata per inviare richieste HTTP GET.
- `wp_remote_post()`: Utilizzata per inviare richieste HTTP POST.

Entrambe le funzioni restituiscono un array associativo che include la risposta del server, che può essere manipolata per estrarre le informazioni desiderate.

### `wp_remote_get()`

Questa funzione è utilizzata per inviare richieste HTTP GET, spesso utilizzata per recuperare dati da un'API.

```
$url = 'https://api.example.com/data';
$response = wp_remote_get( $url );

if ( is_wp_error( $response ) ) {
    // Gestione degli errori
}
```

```

$error_message = $response->get_error_message();
echo "Qualcosa è andato storto: $error_message";
} else {
    // Estrazione dei dati dalla risposta
    $body = wp_remote_retrieve_body( $response );
    $data = json_decode( $body, true );

    // Utilizzo dei dati
    if ( is_array( $data ) && count( $data ) > 0 ) {
        foreach ( $data as $item ) {
            echo 'Elemento: ' . $item['name'];
        }
    }
}
}

```

## wp\_remote\_post()

Questa funzione è utilizzata per inviare dati ad un server tramite una richiesta HTTP POST, spesso utilizzata per inviare dati a un'API.

```

$url = 'https://api.example.com/submit';
$args = [
    'body' => [
        'key1' => 'value1',
        'key2' => 'value2',
    ]
];

$response = wp_remote_post( $url, $args );

if ( is_wp_error( $response ) ) {
    // Gestione degli errori
    $error_message = $response->get_error_message();
    echo "Qualcosa è andato storto: $error_message";
} else {
    // Estrazione dei dati dalla risposta
    $body = wp_remote_retrieve_body( $response );
    $data = json_decode( $body, true );
}

```

```
// Utilizzo dei dati
if ( is_array( $data ) && array_key_exists( 'message',
$data ) ) {
    echo 'Risposta: ' . $data['message'];
}
}
```

Le risposte HTTP possono contenere varie informazioni utili. Ecco alcune funzioni utili per lavorare con le risposte:

- `wp_remote_retrieve_body()`: Recupera il corpo della risposta.
- `wp_remote_retrieve_headers()`: Recupera gli header della risposta.
- `wp_remote_retrieve_response_code()`: Recupera il codice di risposta HTTP.

## Conclusione

Effettuare richieste HTTP in WordPress è semplice grazie alle funzioni `wp_remote_get` e `wp_remote_post`. Queste funzioni permettono di interagire con API esterne in modo sicuro e affidabile, consentendo di estendere le funzionalità del proprio sito WordPress in modi nuovi e interessanti.

Sfruttare queste funzioni permette di integrare servizi esterni, automatizzare processi e migliorare l'interazione con gli utenti. Sperimentare con queste funzioni può aprire nuove possibilità per il tuo sviluppo WordPress.