

# WordPress: introduzione ad AJAX per temi e plugin

AJAX è perfettamente integrato in WordPress. In questo articolo forniremo un'introduzione all'uso di questo standard in WordPress.

## Anatomia di una richiesta AJAX in WordPress

1. JavaScript effettua la richiesta all'URL `https://sito.tld/wp-admin/admin-ajax.php` passando il parametro obbligatorio `action` con il nome della action da eseguire ed altri parametri facoltativi.
2. WordPress legge il nome della action passata come parametro ed esegue il codice associato tramite una funzione o un metodo di una classe restituendo infine un output definito nel codice associato.
3. JavaScript riceve l'output di WordPress e può elaborarlo.

## Definire le action AJAX

WordPress dispone di due tipi di action AJAX:

1. **Private**: eseguite solo per gli utenti loggati
2. **Pubbliche**: eseguite per tutti gli utenti

La prima prende la forma di `wp_ajax_nome_azione`, mentre la seconda è `wp_ajax_nopriv_nome_azione`. Entrambe vengono registrate usando la funzione `add_action()`:

```
function my_encode_value() {  
    $value = trim( $_POST['str'] ); // Parametro
```

```

passato da JavaScript
    $encoded_value = '';
    if( !empty( $value ) ) {
        $encoded_value = md5( $value );
    }
    echo $encoded_value; // Output della
richiesta
    exit; // Obbligatorio!
}

add_action( 'wp_ajax_my_encode_value',
'my_encode_value' );
add_action( 'wp_ajax_nopriv_my_encode_value',
'my_encode_value' );

```

Se non usiamo `exit` dopo aver restituito l'output, WordPress restituirà 0. In questo caso il nome della nostra action è `my_encode_value`. Non è necessario che il nome della funzione o del metodo della classe abbia lo stesso nome della action.

A questo punto possiamo usare JavaScript, in questo caso con jQuery:

```

$( "#form" ).on( "submit", function( e ) {
    e.preventDefault();
    var ajaxURL = "https://" + location.host +
"/wp-admin/admin-ajax.php",
        data = {
            action: "my_encode_value",
            str: $( "#str" ).val()
        };
    $.when( $.post( ajaxURL, data )
).done(function( resp ) {

```

```
        // resp contiene la stringa
codificata in MD5
    });
});
```