Auf die Frage hin von Daniel, wie man in WordPress die Ajax Calls besser absichern kann, möchte ich nochmal auf das Thema Nonce eingehen.

Hier für wird mit der wp_create_nonce() Funktion ein Nonce erzeugt werden.

Dies kann in der functions.php des Themes gemacht werden oder noch besser in der PHP Datei des Plugins.

```
function ud_add_nonce() {
  /* Nonce erzeugen */
  $nonce = wp_create_nonce( 'ud-ajax-nonce' );
  /* Leerer div-Container mit data-Attribut ausgeben */
  echo '<div id="ud-nonce" data-nonce="' . esc_attr( $nonce )
. '"></div>';
}
add_action('wp_footer', 'ud_add_nonce');
```

Hierbei wird durch den Hook add_action im FOOTER Bereich das div ud-nonce angelegt und als Attribut data-nonce der Wert hineingeschrieben.

Logischerweise stellen wir das im CSS auf unsichtbar.

```
#ud-nonce{
        display:none;
}
```

Unseren Ajax Call ergänzen wir nun um das Folgende:

```
 security: jQuery( '#ud-nonce' ).data( 'nonce' )
```

Der Vollständigkeitshalber hier nochmal der gesamte Ajax Call.

```
$.ajax({
    url: „ud_ajax_callback“,
    type: „post",
 security: jQuery( '#ud-nonce' ).data( 'nonce' ),
    data:   {q: str},
    success: function (result) {
        console.log("Ajax Success");

        console.log(result);

        document.getElementById("txtHint").innerHTML =
result;

    },
    error: function(jqXHR, textStatus, errorThrown) {
        console.log(textStatus, errorThrown);
    }
});
```

Dem interessierten Leser wird nun aufgefallen sein, das sich oben auch die URL geändert hat.

Das ist deswegen, weil die Nonce in WP nur so funktionieren, das sie eine registrierte Funktion verwenden.

Die Verknüpfung von Ajax und PHP Function und dem Nonce Funktioniert dann mit wp_ajax_nopriv und wp_ajax. Wobei nopriv im Grunde das andere mit einschließt da alle user auch nicht angemeldete damit zugelassen werden.

```
 add_action('wp_ajax_nopriv_ud_ajax_callback',
'ud_ajax_callback');
 add_action('wp_ajax_ud_ajax_callback', 'ud_ajax_callback');
```

Jetzt fehlt nur noch die Implementierung der PHP Function auf die der Ajax Call hingeht.

```
function ud_ajax_callback(){

  /* Nonce prüfen */
  if ( ! wp_verify_nonce( $_POST['security'],
'ajax_nonce' ) ) {
    wp_die( 'Nonce ist ungültig!' );
  } // endif

Else{
$a[] = "Anna";
$a[] = "Brittany";
$a[] = "Cinderella";
$a[] = "Diana";
$a[] = "Eva";
$a[] = "Fiona";
$a[] = "Gunda";
$a[] = "Hege";
$a[] = "Inga";
$a[] = "Johanna";
$a[] = "Kitty";
$a[] = "Linda";
$a[] = "Nina";
$a[] = "Ophelia";
$a[] = "Petunia";
$a[] = "Amanda";
$a[] = "Raquel";
$a[] = "Cindy";
$a[] = "Doris";
```

```php
$a[] = "Eve";
$a[] = "Evita";
$a[] = "Sunniva";
$a[] = "Tove";
$a[] = "Unni";
$a[] = "Violet";
$a[] = "Liza";
$a[] = "Elizabeth";
$a[] = "Ellen";
$a[] = "Wenche";
$a[] = "Vicky";

//var_dump($_POST);
// get the q parameter from URL
$q = $_POST["q"];


$hint = "";

// lookup all hints from array if $q is different from ""
if ($q !== "") {
    $q = strtolower($q);
    $len=strlen($q);
    foreach($a as $name) {
        if (stristr($q, substr($name, 0, $len))) {
            if ($hint === "") {
                $hint = $name;
```

```
            } else {
                $hint .= ", $name";
            }
        }
    }
}

// Output "no suggestion" if no hint was found or output correct values
echo json_encode($hint);
}

  wp_die(); // benötigt um die AJAX-Anfrage zu beenden
}
```