

# Select-Auswahl mit AJAX Abhängigkeit

== Code ==

index.html

## Quellcode

```
1. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
2. <html xmlns="http://www.w3.org/1999/xhtml" dir="ltr" xml:lang="de">
3. <head>
4. <meta http-equiv="content-type" content="text/html; charset=UTF-8" />
5. <title>Select-Auswahl mit AJAX Abhängigkeit</title>
6. <script type="text/javascript" src="script.js"></script>
7. <style type="text/css">
8. <!--
9. /* demostyle - not necessary to make it work*/
10. body{font-size:11pt;font-family:Verdana,Arial,Sans}
11. //-->
12. </style>
13. </head>
14. <body onload="sendRequest(null, 'elem1')">
16. <form method="post" action="" id="ajaxSelect">
18. <table>
19. <tr id="elem1" style="display:none">
20. <td>HauptKategorie:</td>
21. <td><select name="hauptkategorie" size="4" onchange="sendRequest(this, 'elem2')"><option
   value=""></option></select></td>
22. </tr>
23. <tr id="elem2" style="display:none">
24. <td>UnterKategorie:</td>
25. <td><select name="unterkategorie" size="4" onchange="sendRequest(this, 'elem3')"><option
   value=""></option></select></td>
26. </tr>
27. <tr id="elem3" style="display:none">
28. <td>UnterUnterKategorie1:</td>
29. <td><select name="unterunterkategorie" size="4" onchange="sendRequest(this, 'elem4')"><option
   value=""></option></select></td>
30. </tr>
31. <tr id="elem4" style="display:none">
32. <td>UnterUnterKategorie2:</td>
33. <td><select name="unterunterkategorie" size="4" onchange="sendRequest(this)"><option
   value=""></option></select></td>
34. </tr>
35. </table>
38. <input type="submit" />
39. </div>
40. </form>
42. </body></html>
```

Alles anzeigen

script.js

## Quellcode

1. /\*\*
2. \* clears a select form element

```

3. *
4. * @param targetSel - dom reference to the select element
5. */
6. function clearSelect(targetSel) {
7.   for (var i=targetSel.length; i>=0; i--) {
8.     targetSel.options[i] = null;
9.   }
10. }
11. /**
12. * sends a request under usage of a loading graphic - targettable has to be positioned relative
13. *
14. * @param srcref - reference to the select element with chosen data
15. * @param target - form element name of target element
16. */
17. function sendRequest(domref, target) {
18.   // skip if no target specified
19.   if(!target) return false;
20.   // save reference to next target
22.   if(domref) domref.followup = target;
23.   var req;
25.   try {
26.     req = window.XMLHttpRequest ? new XMLHttpRequest():
27.     new ActiveXObject("Microsoft.XMLHTTP");
28.   } catch (e) {
29.     // no AJAX Support
30.   }
31.   req.onreadystatechange = function() {
32.     if ((req.readyState == 4) && (req.status == 200)) {
33.       // merge empty line with response
34.       var data = eval('(' + req.responseText + ')');
35.       var targetRef = document.getElementById(target);
36.       var targetSel = targetRef.getElementsByTagName('select')[0];
37.       // make it visible
39.       targetRef.style.display = 'block';
40.       // clear old data
42.       clearSelect(targetSel);
43.       // fill with data from json response
45.       var i=0;
46.       for(var x in data) {
47.         targetSel.options[i++] = new Option(
48.           data[x].text,
49.           data[x].id
50.         );
51.       }
52.       // clear all followups
54.       while(targetSel.followup) {
55.         targetRef = document.getElementById(targetSel.followup);
56.         // make it hidden
58.         targetRef.style.display = 'none';
60.         // mark next select
61.         targetSel = targetRef.getElementsByTagName('select')[0];
62.         // clear old data
64.         clearSelect(targetSel);
65.       }
66.     }
67.   }
68.   req.open('post', 'ajax.php');
70.   req.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');

```

```

72. // send empty post with initial load
73. req.send(domref !== null ? 'id='+domref.value+'&name='+domref.name : '');
74. return false; // return false to avoid reload/recentering of the page
76. }

```

Alles anzeigen

ajax.php

Die meisten werden in dem Umgang mit einer Datenbank vertraut sind. Die ajax.php empfängt eine \$\_REQUEST Variable "id" und muss eine zweidimensionales Array mit den Schlüsseln "id" und "key" zurückgeben. Auf Basis des [PDO Tutorials](#) habe ich folgenden Demo Code erstellt:

## Quellcode

```

1. <?php
2. $sql = "SELECT id, text
3. FROM category
4. WHERE parentID = :id";
5. $stmt = MyDB::getInstance()->prepare($sql);
6. $stmt->execute(array(
7. 'id' => $_REQUEST['id']
8. ));
9. $return = $stmt->fetchAll();
10. echo json_encode($return);
12. ?>

```

Alles anzeigen

Für das Demoscript habe ich folgenden Code bereitgestellt:

## Quellcode

```

1. <?php
2. // u can use $_POST['name'] to build your controller depending on the form name
3. // but in a category tree the name is not important, we just need the parent id
4. $id = isset($_REQUEST['id']) ? $_REQUEST['id'] : 0;
6. // (1) example array with four main categories
7. for($i=0; $i<4; $i++) {
8. // four subcategories each
9. for($j=($i*10); $j<($i*10)+3; $j++) {
10. $idx = $j+1;
11. $data[$i][$j] = array(
12. 'id' => $idx,
13. 'text' => sprintf('cat %d (parent:%d)', $idx, $i),
14. );
16. // with three childs eiter
17. for($k=($idx*10); $k<($idx*10)+3; $k++) {
18. $data[$idx][$k] = array(
19. 'id' => $k,
20. 'text' => sprintf('subcat %d (parent:%d)', $k, $idx),
21. );
22. }
23. }
24. }
26. // access the childs with its parent index
27. $return = $data[$id];
28. echo json_encode($return);

```

30. ?>

Alles anzeigen

== Abhängige Felder definieren ==

Das Beispiel lässt sich um beliebig viele Select-Auswahlfelder erweitern. Ihr müsst der Methode `sendRequest` als ersten Parameter immer `this` übergeben und als **zweiten Parameter** die ID des Containers, der das abhängige Select-Feld enthält. Am Ende der Kette - wenn also keine abhängigen Felder mehr nachgeladen werden sollen - übergebt ihr einfach gar keinen Parameter.

### Quellcode

```
1. <tr id="elem1">
2. <td>Haupt Kategorie:</td>
3. <td><select onchange="sendRequest(this, 'elem2')"></select></td>
4. </tr>
5. <tr id="elem2">
6. <td>Unter Kategorie:</td>
7. <td><select></select></td>
8. </tr>
```

== Felder nicht verstecken ==

Im konkreten Beispiel wurde entschieden abhängige Select-Auswahlfelder erst anzuzeigen, sobald das Elternelement die notwendige Auswahl getroffen hat. Sollte es besser ins Layout passen und ihr wollt stattdessen leere Listenelemente verwendet, so entfernt einfach die style Eigenschaft `display:none`. Das müsst ihr sowohl im HTML Code machen, als auch alle `display` Aufrufe im Quelltext entfernen.

== Demo ==

Eine Live Demo findet ihr unter [demo.easy-coding.de/ajax/selec...hl-mit-ajax-abhaengigkeit](http://demo.easy-coding.de/ajax/selec...hl-mit-ajax-abhaengigkeit). Des weiteren wird der kompletten Code hier als ZIP Archiv zur Verfügung gestellt: [download.zip](#).

[easy-coding.de/Attachment/564/...a4c2e0b4cccf3f02786740bca](http://easy-coding.de/Attachment/564/...a4c2e0b4cccf3f02786740bca)

== Fortsetzung ==

Mehr Informationen zum Datenbankbeispiel findet ihr auch hier: [wiki][Select-Auswahl mit AJAX Abhängigkeit und verschiedenen Tabellen](#)[/wiki]