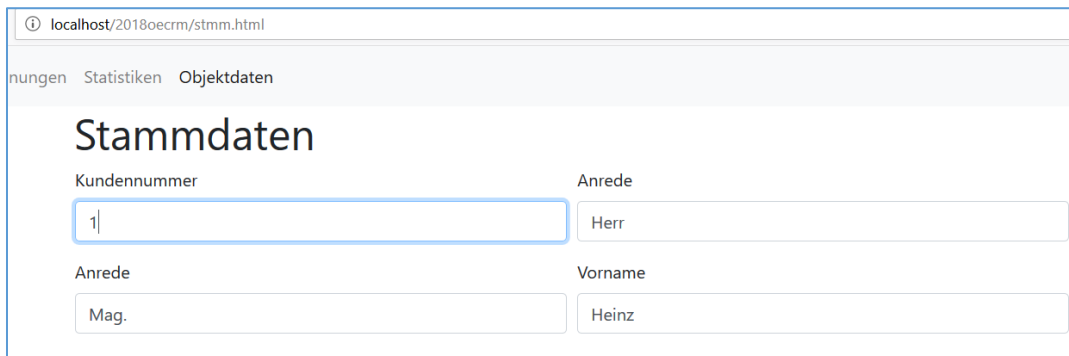


## Inputfeld automatisch ausfüllen

Ziel: mit der Eingabe einer ID-Nummer sollen die dazugehörigen Daten automatisch angezeigt werden. Beispiel: Eingabe der Kundennummer ergibt:



localhost/2018oecrm/stmm.html

nungen Statistiken Objektdaten

### Stammdaten

Kundennummer: 1

Anrede: Herr

Anrede: Mag.

Vorname: Heinz

Inhalt:




1. HTML-Datei mit JavaScript und jQuery und Ajax
2. Datenbank anlegen
3. PHP-Datei erstellen, die Daten aus der Datenbank holt

### 1.)HTML-Datei mit JavaScript und jQuery und Ajax

Datei „stmm.html“ mit Bootstrap erstellen. Es wird auch JavaScript benötigt, daher sollen die „bootstrap.min.js“ und auch (wegen ajax) die aktuelle „jquery.min.js“ im <head> eingebunden werden. Die „jquery.min.js“ lade runter von [www.jquery.com](http://www.jquery.com)

```
1 <!DOCTYPE html>
2 <html lang="de">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7   <title>Stammdaten</title>
8
9   <!-- Bootstrap CSS -->
10  <link rel="stylesheet" href="css/bootstrap.min.css">
11  <script src="js/jquery-3.3.1.min.js"></script>
12  <script src="js/bootstrap.min.js"></script>
13
14 </head>
15 <body>
```

Beide js-Dateien sollen im Ordner „js“ liegen.

Name	Änderungsdatum	Typ	Größe
 bootstrap	25.02.2018 13:09	JavaScriptdatei	113 KB
 bootstrap.min	25.02.2018 13:10	JavaScriptdatei	48 KB
 jquery-3.3.1.min	09.08.2018 09:41	JavaScriptdatei	85 KB

## Formular erstellen:

In einem Container.

Das Formular hat **KEINEN Submit-Button**. Dafür gleich am Beginn **eine ID**. Hier die id="myTest". Damit wird das Formular später in der JavaScript-Funktion (Zeile 53) angesprochen.

```
21 ▼ <form id="myTest">
```

Wichtig:

Jedes Inputfeld erhält eine eindeutige ID (z.B. „Anrede“). Diese wird später in der JavaScript-Funktion gefüllt (mit den Daten, die man aus der PHP-Datei erhält):

```
67         $("#Anrede").val(sp[1]);
68         $("#Titel").val(sp[2]);
69         $("#Vorname").val(sp[3]);
70     },
```

```
17 ▼ <div class="container">
18 ▼     <div class="offset-md-1">
19     <h1>Stammdaten</h1>
20     </div>
21 ▼ <form id="myTest">
22 ▼     <div class="form-row">
23 ▼         <div class="form-group offset-md-1 col-md-5">
24             <label for="KundenNr" >Kundennummer</label>
25             <input type="text" class="form-control" name="KundenNr" id="KundenNr">
26         </div>
27 ▼         <div class="form-group col-md-5 offset-md-12">
28             <label for="Anrede" >Anrede</label>
29             <input type="text" class="form-control" name="Anrede" id="Anrede">
30         </div>
31 ▼         <div class="form-group offset-md-1 col-md-5">
32             <label for="Titel" >Anrede</label>
33             <input type="text" class="form-control" name="Titel" id="Titel">
34         </div>
35 ▼         <div class="form-group col-md-5 offset-md-12">
36             <label for="Vorname" >Vorname</label>
37             <input type="text" class="form-control" name="Vorname" id="Vorname">
38         </div>
39     </div>
40 </form>
41     <div id="error"></div> <!--error anzeigen-->
42 </div> <!--ende container-->
43 </body>
44 </html>
```

In Zeile 41 wird eine mögliche Meldung angezeigt, die hier „error“ heißt. Es ist ein leerer <div>, die mit Hilfe der JavaScript-Funktion gefüllt wird, wenn keine Zahl eingegeben wird:

```
50         if(KundenNr=== "")
51 ▼         {
52             $("#error").html("Feld verpflichtend");
53             $("#myTest").trigger("reset");
54             return false;
55         }
56         else
```

Ergebnis:

Stammdaten

Kundennummer  Anrede

Anrede  Vorname

Feld verpflichtend

### JavaScript-Funktion mit jQuery-Methoden:

Die Funktion liegt außerhalb der <html>. Es werden bekannte jQuery-Befehle verwendet:

```
44 </html>
45 <script>
46 $(document).ready(function(){
47     $('#KundenNr').on("keyup",function(){
48         var KundenNr = $('#KundenNr').val(); //KundenNr von input-Zeile oben
49
50         if(KundenNr=== "")
51         {
52             $('#error').html("Feld verpflichtend");
53             $('#myTest').trigger("reset");
54             return false;
55         }
56         else
57         {
58             $('#error').html("");
59             $.ajax({
60                 url:"stmmaction.php",
61                 method:"POST",
62                 data: {KundenNr:KundenNr},
63                 success:function(data)
64                 {
65                     var fdata=$.trim(data);
66                     var sp=fdata.split(",");
67                     $('#Anrede').val(sp[1]);
68                     $('#Titel').val(sp[2]);
69                     $('#Vorname').val(sp[3]);
70                 }
71             });
72         }
73     });
74
75 });
76 </script>
```

- Die \$(document).ready()-Funktion stellt sicher, dass das Dokument fertig geladen ist.  
Info: [https://www.w3schools.com/jquery/event\\_ready.asp](https://www.w3schools.com/jquery/event_ready.asp)
- **OnKeyup-Methode:**

\$(selector).keyup(function)

Das keyup Ereignis tritt auf, wenn eine Taste der Tastatur losgelassen wird. In unserem Fall, wenn eine Nummer eingegeben wird. Die Raute vor der Kundennummer macht die Beziehung zur ID dort, wo die ID erstellt wurde, nämlich in Zeile 25 beim „input“-Feld.

```

45 <script>
46 $(document).ready(function(){
47     $('#KundenNr').on("keyup",function(){
48         var KundenNr = $("#KundenNr").val();

```

Info: [https://www.w3schools.com/jquery/event\\_keyup.asp](https://www.w3schools.com/jquery/event_keyup.asp)

Wenn keine Zahl eingegeben wird, erfolgt die Fehlermeldung mit „error“:

```

50         if(KundenNr=="")
51         {
52             $("#error").html("Feld verpflichtend");
53             $("#myTest").trigger("reset");
54             return false;
55         }
56         else

```

Ansonsten wird ganz korrekt die AJAX-Funktion durchgeführt, die die Felder mit Hilfe der PHP-Datei gefüllt:

```

56         else
57         {
58             $("#error").html("");
59             $.ajax({
60                 url:"stmmaction.php",
61                 method:"POST",
62                 data: {KundenNr:KundenNr},
63                 success:function(data)
64                 {
65                     var fdata=$.trim(data);
66                     var sp=fdata.split(",");
67                     $("#Anrede").val(sp[1]);
68                     $("#Titel").val(sp[2]);
69                     $("#Vorname").val(sp[3]);
70                 }
71             });

```

- **AJAX-Funktion:**

Damit werden asynchron die angegebenen Felder in Zeile 67-69 gefüllt, ohne dass das Formular neu geladen werden muss.

Info: [https://www.w3schools.com/jquery/ajax\\_ajax.asp](https://www.w3schools.com/jquery/ajax_ajax.asp)

In Zeile 65 und 66 wird die Zeichenkette, die aus der PHP-Datei kommt zerlegt:

- **Trim():**

Die trim () Methode entfernt Leerzeichen von beiden Seiten einer Zeichenkette.

Info: [https://www.w3schools.com/jsref/jsref\\_trim\\_string.asp](https://www.w3schools.com/jsref/jsref_trim_string.asp)

```

65         var fdata=$.trim(data);
66         var sp=fdata.split(",");

```

- **Split():**

Als Separator wird hier der Beistrich verwendet.

Info: Wenn ein leerer String ( „“ ) als Separator verwendet wird, wird die Zeichenkette zwischen den Zeichen aufgeteilt.

Info: [https://www.w3schools.com/jsref/jsref\\_split.asp](https://www.w3schools.com/jsref/jsref_split.asp)

Die Zeichenkette kommt von der PHP-Datei, die dort in Zeile 10 erzeugt wird:

```
7     $query = "SELECT * FROM stammdaten WHERE KundenNr = $KundenNr ";
8     $result = mysqli_query($connect, $query);
9     while($serg = mysqli_fetch_assoc($result))
10        echo $serg["KundenNr"] . "," . $serg["Anrede"] . "," . $serg["Titel"] . "," . $serg["Vorname"];
11     ?>
12
```

- **Val()-Methode:**

Val () Methode gibt oder legt das Wertattribut der ausgewählten Elemente.

Bei der Verwendung zur Rückgabewert:

Diese Methode gibt den Wert des Wertattributs des angegebenen Elements wieder.

Null wäre ja unnötig, da es ja die Kundennummer ist.

```
67         $("#Anrede").val(sp[1]);
68         $("#Titel").val(sp[2]);
69         $("#Vorname").val(sp[3]);
70     }
    }
```

## 2) Datenbank-Tabelle erstellen

Tabelle: stammdaten

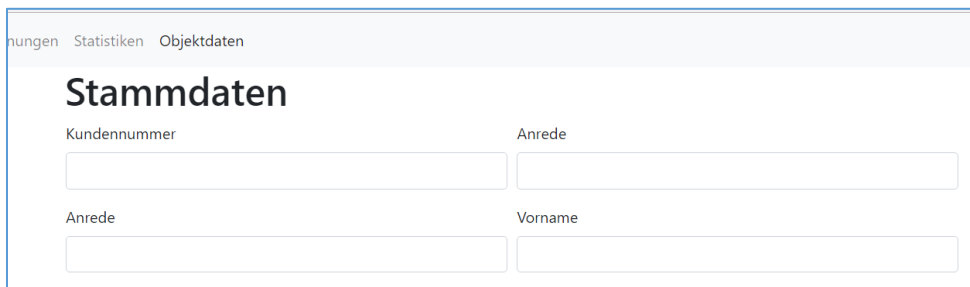
**Achtung: in der Datenbank darf die KundenNr nicht „Kunden-Nr“ heißen – kein Bindestrich!!!**



The screenshot shows the MySQL database interface. On the left, a tree view shows the database structure with 'stammdaten' selected. On the right, the table's data is displayed. The table has columns: KundenNr, Anrede, Titel, Vorname, Nachname, and Firmenname. Two rows of data are shown: (1, Herr, Mag., Heinz, Nichtheinz) and (2, Frau, Lisa, Immernochlisa). There are also options to 'Alles anzeigen' and a dropdown for 'Anzahl der Datensätze' set to 25.

KundenNr	Anrede	Titel	Vorname	Nachname	Firmenname
1	Herr	Mag.	Heinz	Nichtheinz	
2	Frau		Lisa	Immernochlisa	

Test mit vier Feldern aus der „stmm.html“



The screenshot shows a web form titled 'Stammdaten'. It has four input fields arranged in a 2x2 grid. The top-left field is labeled 'Kundennummer', the top-right 'Anrede', the bottom-left 'Anrede', and the bottom-right 'Vorname'. The form is part of a larger application with navigation links for 'Anfragen', 'Statistiken', and 'Objektdaten'.

### 3)PHP-Datei

Erstelle die Verbindung zur Datenbank.

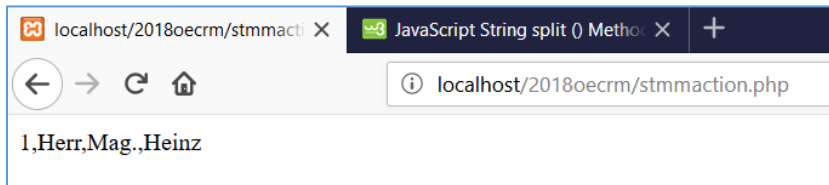
```
1 <?php
2 session_start();
3 $connect = mysqli_connect("localhost", "root", "", "meintest");
4
```

Danach wird in Zeile 5 eine Variable erstellt, die aus der „stmm.html“ die Eingabe der Kundennummer mit der passenden ID holt:

```
1 <?php
2 session_start();
3 $connect = mysqli_connect("localhost", "root", "", "meintest");
4
5 $KundenNr = $_POST["KundenNr"];
6
7 $query = "SELECT * FROM stammdaten WHERE KundenNr = $KundenNr ";
8 $result = mysqli_query($connect, $query);
9 while($erg = mysqli_fetch_assoc($result))
10     echo $erg["KundenNr"] . "," . $erg["Anrede"]
11     |. "," . $erg["Titel"] . "," . $erg["Vorname"];
12 ?>
13
```

Die einfache Abfrage bringt mit der while-Schleife alle Daten aus der Datenbank hervor, die zur Kundennummer passen. Diese werden in einer Zeichenkette mit Komma getrennt ausgegeben.

Das würde (etwas umgeformt) so aussehen:



Beispiel für „1“:

```
1 <?php
2 session_start();
3 $connect = mysqli_connect("localhost", "root", "", "meintest");
4
5
6
7 $query = "SELECT * FROM stammdaten WHERE KundenNr = 1| ";
8 $result = mysqli_query($connect, $query);
9 while($erg = mysqli_fetch_assoc($result))
10     echo $erg["KundenNr"] . "," . $erg["Anrede"]
11     . "," . $erg["Titel"] . "," . $erg["Vorname"];
12 ?>
```

Quelle: [https://www.youtube.com/watch?v=sO\\_gGo\\_rsHA](https://www.youtube.com/watch?v=sO_gGo_rsHA)