

## Autocomplete Funktion in der Smartform

Eine Autocomplete Funktion ist eine Funktion, die bei Eingabe in ein Textfeld Vorschläge gibt welche dann ausgewählt werden können. Ein Beispiel wäre ein Feld für die Auswahl eines Mitarbeiters.

### jQuery

Für den ersten Schritt muss die Javascript Bibliothek jQuery eingebunden werden. Diese beinhaltet bereits einfache Möglichkeiten zur Einbindung von Autocomplete Funktionen. Die Einbindung muss in der initMethod der Smartform geschehen, damit jQuery zu jederzeit verfügbar ist. Die Einbindung erfolgt über die folgende Zeile:

```
jq = (this.form.ownerDocument.defaultView!=null) ?  
this.form.ownerDocument.defaultView.jQuery :  
this.form.ownerDocument.parentWindow.jQuery;
```

Der „\$“ Operator ist nun über den „jq“ Operator verfügbar und ist nötig um jQuery Funktionalitäten benutzen zu können.

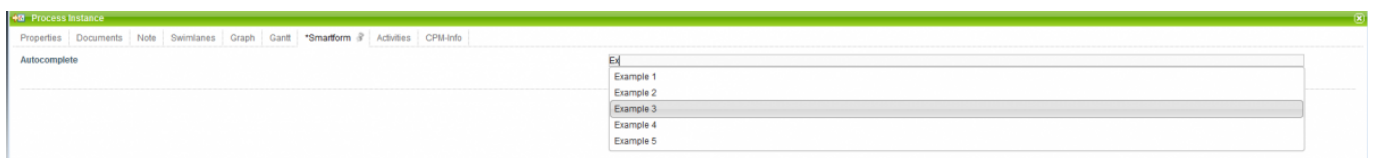
### Autocomplete

Jeder beliebige Textfeld kann nun ausgewählt werden um einen Autocomplete zu beinhalten. Hierfür muss das Feld über einen sogenannten Selektor gefunden werden. Ein Feld kann mit Hilfe seiner ID über die folgende Zeile gefunden werden:

```
jq('#ID_DES_FELDES');
```

Die erste Ausbaustufe des Autocomplete wird an Hand eines einfachen Beispiels erörtert. Hierbei wird dem Feld ein Autocomplete aufgesetzt, welchem ein sogenannter Array (stellt eine Liste von Werten dar) als Quelle dient.

```
jq( "#ID_DES_FELDES" ).autocomplete({  
    source: [ "Example 1", "Example 2", "Example 3", "Example 4",  
    "Example 5"]  
});
```



Im nächsten Schritt muss diese Funktion nun erweitert werden, sodass die Auswahl die getroffen wird auch gespeichert wird. Hierfür muss der Code folgendermaßen angepasst werden. Es muss eine 'change' Funktion hinzugefügt werden welche bei Auswahl einer Vorschlagsmöglichkeit aufgerufen wird. In dieser Methode wird nun der so genannten 'entity' der Name und der Wert des Inputfeldes gegeben. Nun ist es TIM möglich diese Prozessvariable abzuspeichern. In der change Methode ist es

ebenfalls möglich eine Validierung einzubauen die überprüft ob eine Auswahl getroffen wurde.

```
jq = (this.form.ownerDocument.defaultView!=null) ?
this.form.ownerDocument.defaultView.jQuery :
this.form.ownerDocument.parentWindow.jQuery;
jq( "#auto" ).autocomplete({
  source: [ "Example 1", "Example 2", "Example 3", "Example 4", "Example
5"],
  change:function(ui){
    if(ui.item == null){
      alert("Please choose an entry from the list!");
    } else {
      window['entity'].setValue(this.id, ui.item.value);
    }
  }
});
```

Für das Autocomplete können nun noch zusätzliche Parameter eingestellt werden. Um das Beispiel weiter auszubauen, fügen wir hinzu dass erst ab einer Länge von 3 Zeichen der Eingabe die Vorschläge angezeigt werden. Dies hat vor allem bei großen Datensätzen den Vorteil dass die Auswahlliste nicht übermäßig lang wird. Um diese zu realisieren fügen wir die Einstellung „minLength“ hinzu. Weiter Parameter können in der [API](#) von jQuery eingesehen werden.

```
jq = (this.form.ownerDocument.defaultView!=null) ?
this.form.ownerDocument.defaultView.jQuery :
this.form.ownerDocument.parentWindow.jQuery;
jq( "#auto" ).autocomplete({
  source: [ "Example 1", "Example 2", "Example 3", "Example 4", "Example
5"],
  minLength:3,
  change:function(ui){
    if(ui.item == null){
      alert("Please choose an entry from the list!");
    } else {
      window['entity'].setValue(this.id, ui.item.value);
    }
  }
});
```

## Autocomplete mit CSV Datei als Quelle

Die Vorschläge für das Autocomplete können natürlich auch komplexer sein und mehrere Informationen enthalten. Hierfür muss dem Autocomplete zum Beispiel ein Textfile übergeben. Für dieses Beispiel benutzen wir eine Beispieldatei mit Personaldaten. Diese enthält Vornamen, Nachnamen, Email und Telefonnummer in den einzelnen Spalte der CSV. Die Datei liegt in dem Beispiel im Custom Ordner. Zuerst muss diese Datei geholt werden und ein Objekt aus dessen Inhalten erzeugt werden. Dies geschieht per AJAX Request. Der Inhalt der Datei wird zuerst an jedem Zeilenumbruch geteilt und anschließend an jedem Semikolon. Es wird nun also ein Objekt employee,,

mit den Daten aus der Textdatei generiert. Das Objekt employee wird nun als Source dem Autocomplete übergeben. Im Falle einer Auswahl, wird Vorname, Nachname, Email und Telefonnummer in einem Popup ausgegeben. Kann das Textfile nicht geladen werden wird eine Fehlermeldung ausgegeben.

```
jq.ajax({
  type: "post",
  url: "/loom-portal/custom/tim/personaldaten.txt",
  dataType: "html",
  success: function(data){
    var employees = new Array();
    var allRows = data.split("\n");
    for(var r=0;r<allRows.length;r++){
      var columns = allRows[r].split(";");
      employees[r]={
        value: columns[0] + ", " + columns[1],
        firstname: columns[0],
        lastname: columns[1],
        email: columns[2],
        phone: columns[3]
      };
    }
    jq( "#ID_DES_INPUTFELDES" ).autocomplete({
      source: employees,
      minLength: 0,
      select: function(event, ui){
        this.value = ui.item.value;
        window['entity'].setValue(this.id,ui.item.value);
        window['entity'].mergeLocal(true);
        alert("Suche: " + ui.item.value + "\nFirstname: " +
ui.item.firstname + "\nLastname: " + ui.item.lastname + "\nEmail: " +
ui.item.email + "\nPhone: " + ui.item.phone);
      },
      change: function(event, ui){
        if(ui.item == null){
          alert("Please choose an entry from the list!");
          this.value="";
        }
      }
    });
  },
  error: function(){
    alert('Could not load data');
  }
});
```

Last update: 2021/07/01 09:56 software:tim:smartform:autocomplete <https://wiki.tim-solutions.de/doku.php?id=software:tim:smartform:autocomplete&rev=1400767332>

---

From: <https://wiki.tim-solutions.de/> - **TIM Wiki** / **[NEW TIM 6 Documentation](#)**

Permanent link: <https://wiki.tim-solutions.de/doku.php?id=software:tim:smartform:autocomplete&rev=1400767332>

Last update: **2021/07/01 09:56**

