TONIC und Form.io

Erstellung von interaktiven Formularen mit Form.io

Version: 0.2  
Stand: 12.02.2020

Inhalt

[1. Einführung 3](#_Toc32386850)

[2. Ein einfacher Eingabedialog 4](#_Toc32386851)

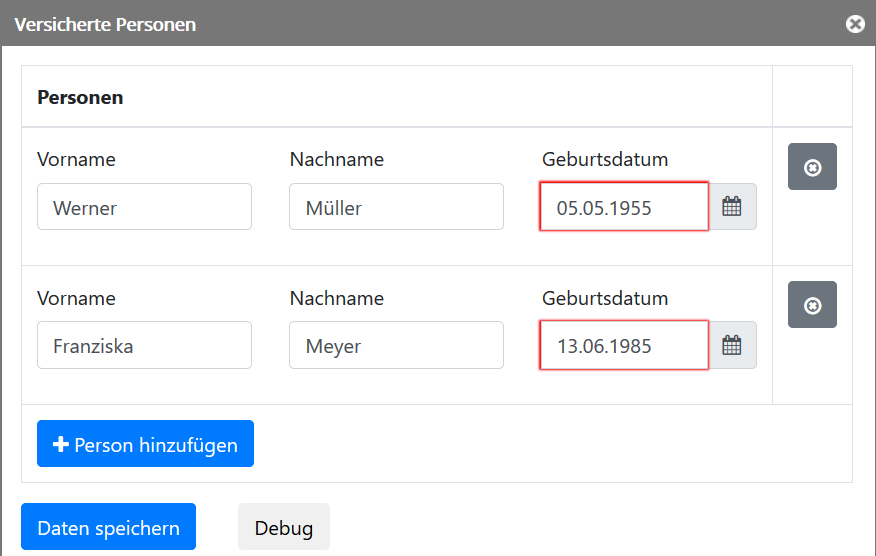
[3. Auswahlliste aus Mehrfach-Datenknoten 7](#_Toc32386852)

[4. Mehrfach-Datenknoten im Editor mit Werten füllen 10](#_Toc32386853)

# Einführung

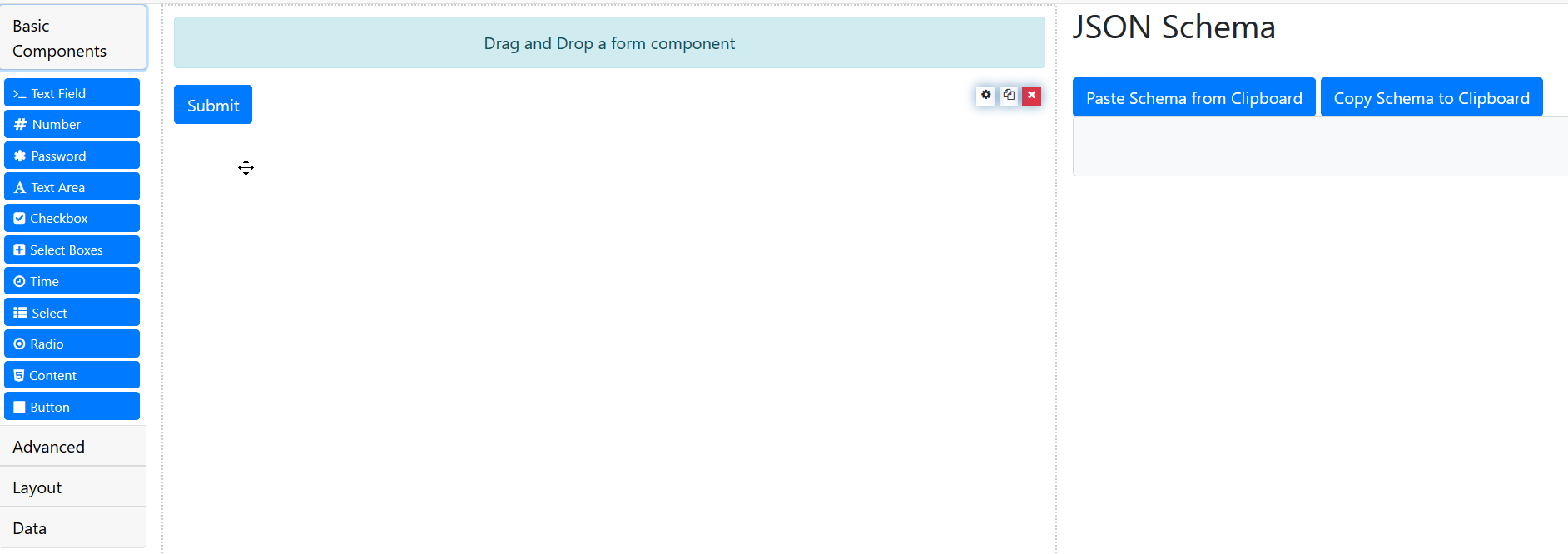
Form.IO bietet in Zusammenarbeit mit TONIC die Möglichkeit, individuelle Benutzerdialog zur Eingabe von Daten im TONIC-Editor zu erzeugen.

Dabei stehen verschiedene Eingabeelemente, wie z. B. Text- und Zahleneingabefelder, oder auch Kalenderdialoge zur Verfügung.

**

Um den Formulareditor zu starten, rufen Sie folgende Webseite auf Ihrem M/TEXT-Server auf:

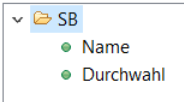
[{Servername}:8080/mtext/formIOBuilder.html](http://m-centos:8080/mtext/formIOBuilder.html)



# Ein einfacher Eingabedialog

An einem einfachen Beispiel soll die Erstellung eines Dialoges zur Eingabe eines Textes und einer Nummer gezeigt werden.

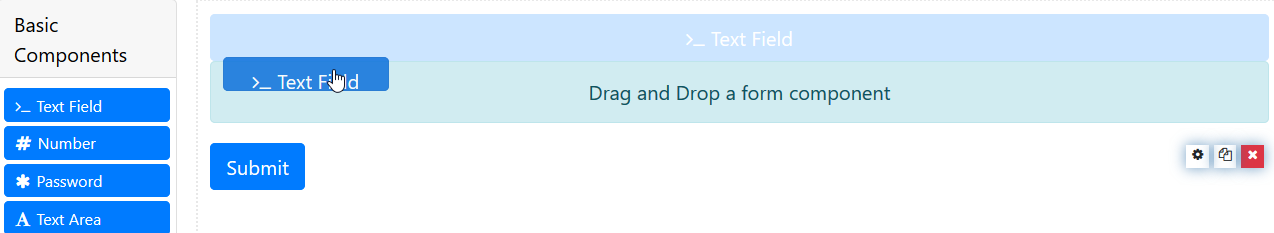
Das benutzte Datenmodell enthält die Knoten Name (Typ Text) und Durchwahl (Typ Numerisch).



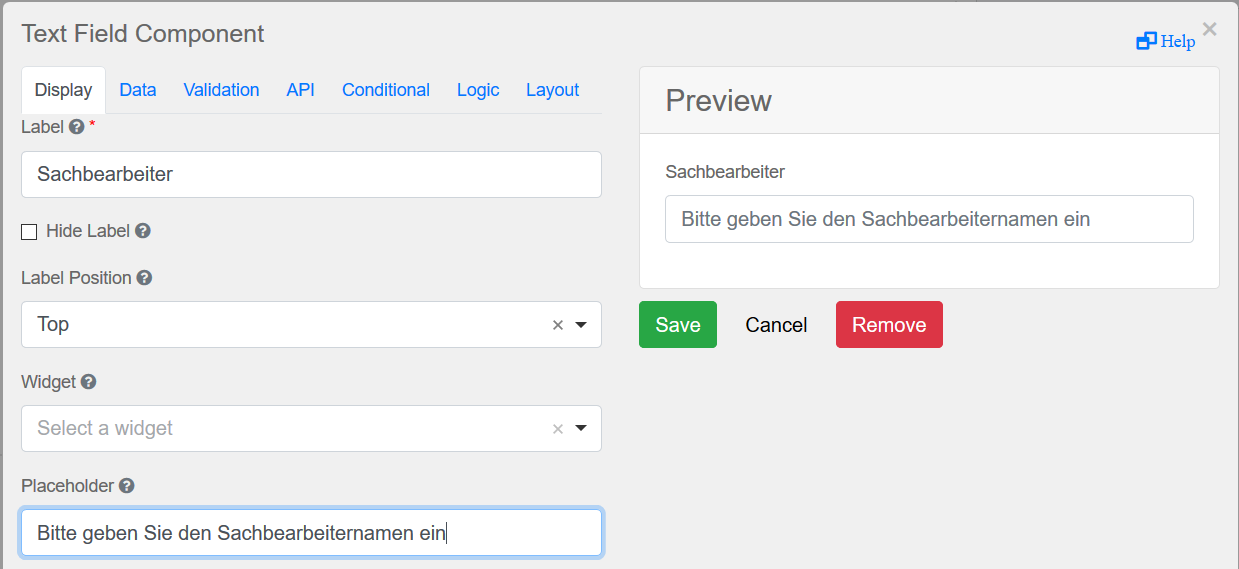
Diese Werte sollen über einen Form.io-Dialog gefüllt werden.

Öffnen Sie den Formulareditor mit einem Aufruf von {Servername}:8080/mtext/formIOBuilder.html

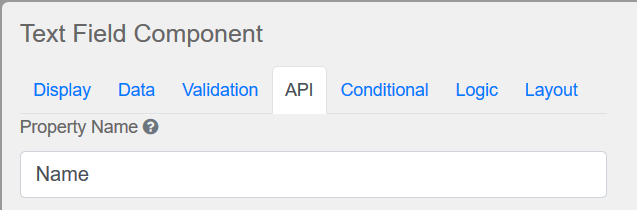
Ziehen Sie ein Textfeld in den Zielbereich:



In dem Reiter **Display** füllen Sie die Eingabefelder **Label** und **Placeholder**:

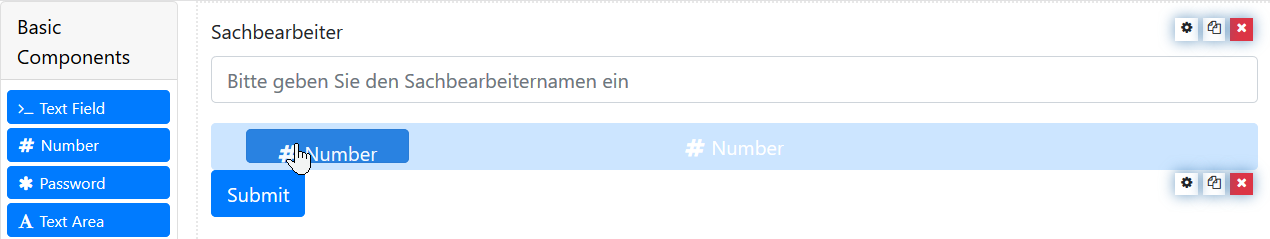


In dem Reiter **API** tragen Sie den Namen des Datenmodell-Knotens ein:

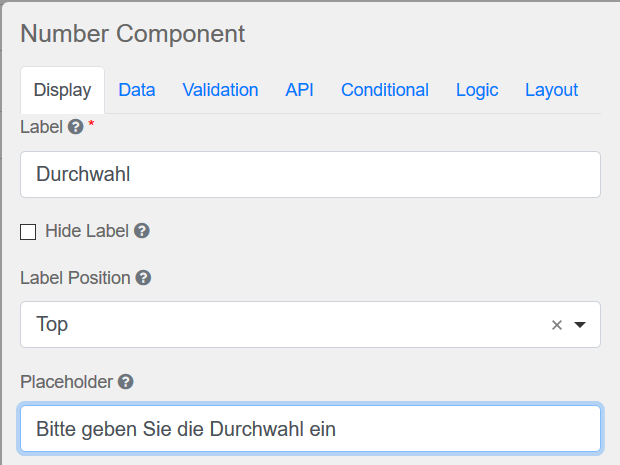


Klicken Sie anschließend auf **Save**.

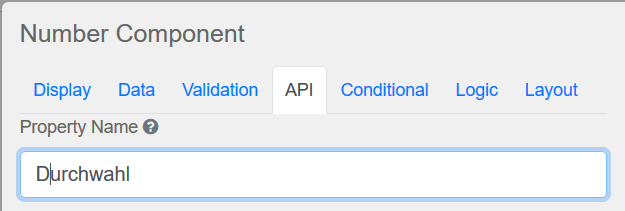
Verfahren Sie genauso für das zweite Eingabefeld. Dieses Mal ziehen Sie ein Feld vom Typ **Number** zwischen das Eingabefeld vom Sachbearbeiter und der **Submit**-Schaltfläche.



Füllen Sie erneut die Felder **Label** und **Placeholder**.

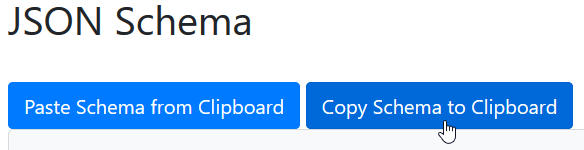


Tragen Sie im Reiter API den Datenmodellknoten **Durchwahl** ein. Sie werden feststellen, dass dieser bereits mit „durchwahl“ gefüllt ist. Dieser Wert wird aus dem Label abgeleitet. Da unser Knoten jedoch mit einem Großbuchstaben beginnt, müssen wir dies anpassen.



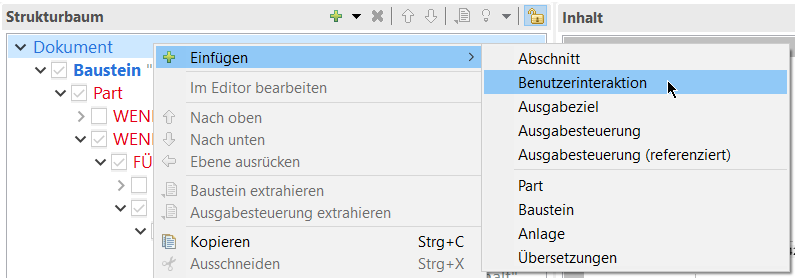
Klicken Sie erneut auf **Save**.

Die so erzeugte JSON-Konfiguration kopieren Sie nun in die Zwischenablage.

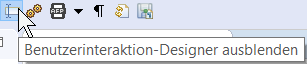


Erstellen Sie in der Workbench eine neue Datei mit dem Namen SB.form.io.json und kopieren den Inhalt dort hinein.

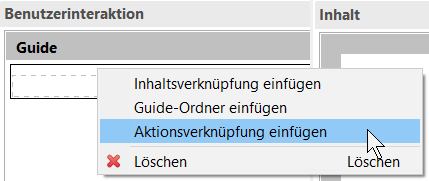
Um den Dialog nun aus Ihrer Vorlage aufrufen zu können, fügen Sie Ihrem Dokument eine Benutzerinteraktion hinzu.



Schalten Sie den Benutzerinteraktions-Designer ein.



Fügen Sie nun im Guide eine Aktionsverknüpfung hinzu.



Wechseln Sie in die Code-Ansicht und fügen Sie in der Aktionsverknüpfung das dazugehörigen Skript ein. Dem Aufruf $context.showForm() wird das erzeugte JSON sowie der Datenmodelknoten $SB und das Layout übergeben.

<UIContribution>

<GuideArea>

<ActionLink title=*"Sachbearbeiter"* hint=*""* inputAreaGroupID=*""*>

<Script><![CDATA[

var formFunc = $context.FormIO.createForm(

"\\\\Framework\_TONIC\_Demo\\Vorlagen\\FormIO\\SB.form.io.json");

var layout = {

"height": "700",

"width" : "700",

"title" : "Sachbearbeiter"

};

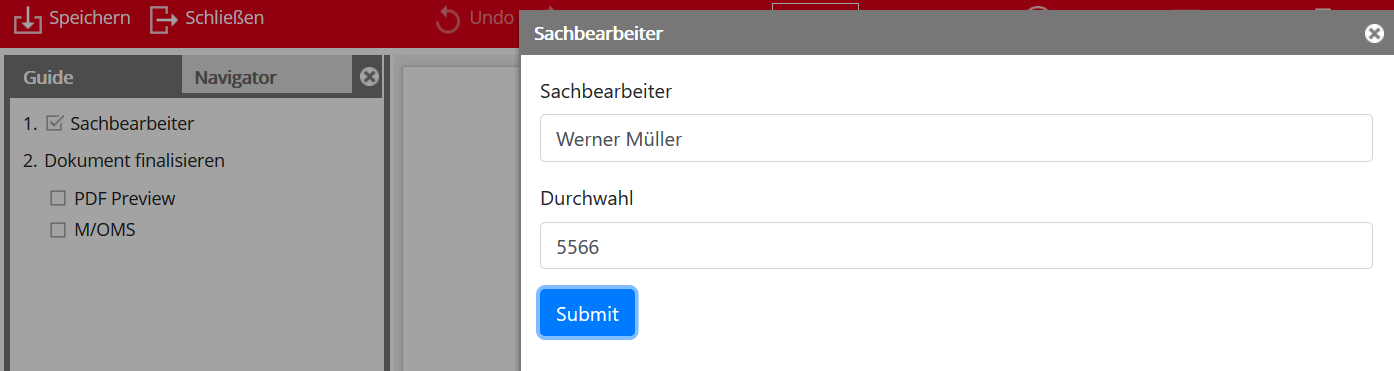
$context.showForm(formFunc, $SB, layout);]]></Script>

</ActionLink>

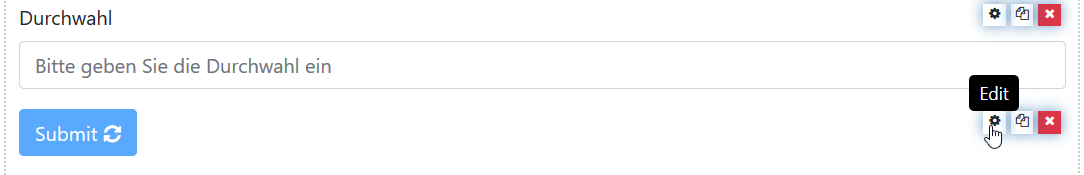
</GuideArea>

</UIContribution>

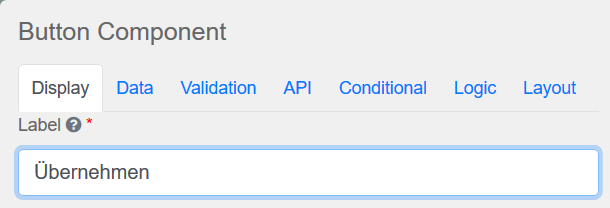
Wenn Sie Ihre Vorlage nun auf dem Server aufrufen, können Sie im Guide den Sachbearbeiter-Dialog anklicken und füllen diesen anschließend aus.



Um noch die Schaltfläche **Submit** umzubenennen, wechseln Sie noch einmal in den Dialog-Editor auf dem Server. Klicken Sie rechts von Submit auf die **Edit**-Schaltfläche.



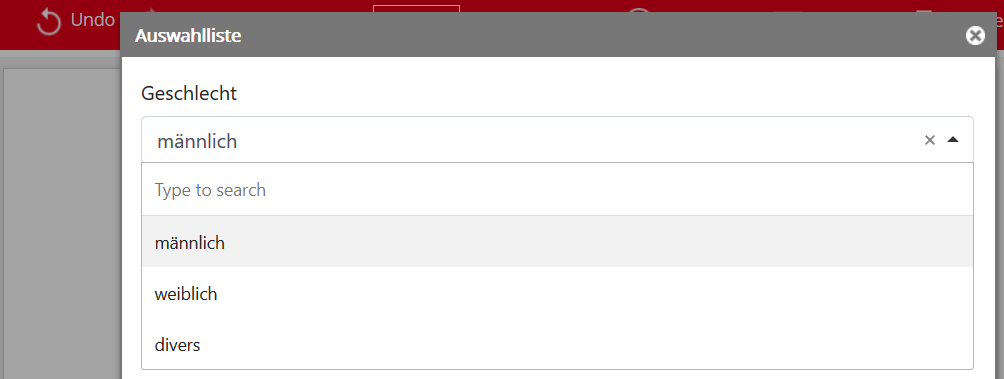
Jetzt können Sie die Schaltfläche umbenennen.



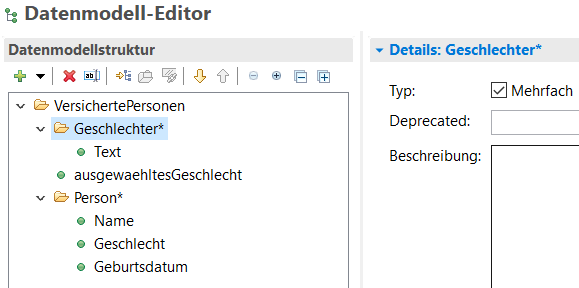
Speichern Sie die Änderungen und exportieren noch einmal das JSON-Schema. Übernehmen Sie die Änderungen in die Datei SB.form.io.json

# Auswahlliste aus Mehrfach-Datenknoten

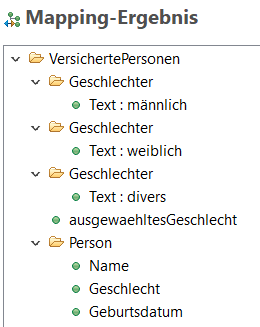
Ziel ist es, dem Anwender im TONIC Editor eine Auswahlliste bereitzustellen, die durch vorhandene Daten aus einem XML gefüllt wird.



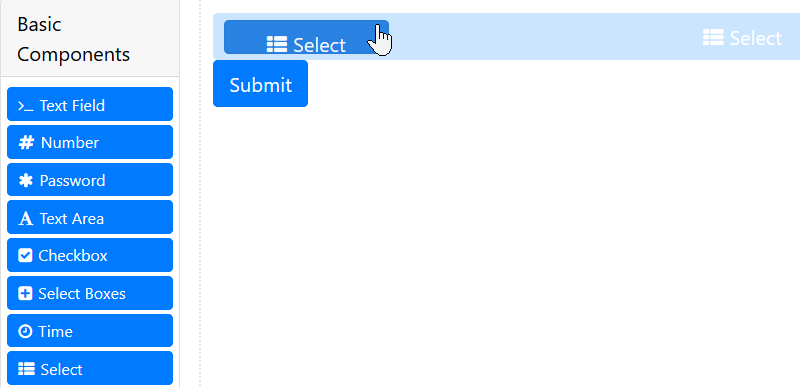
In dem Beispiel-Datenmodell existiert ein Mehrfachknoten **Geschlechter\*** mit einem Unterknoten **Text**. Dieser wird durch eine XML-Datei mit Werten befüllt.



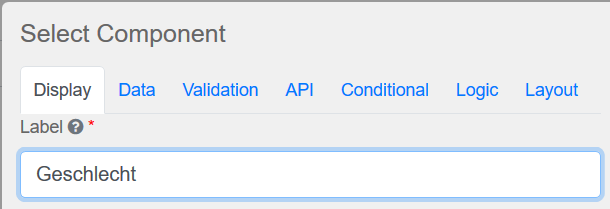
Nach dem Mapping sieht das Ergebnis so aus:



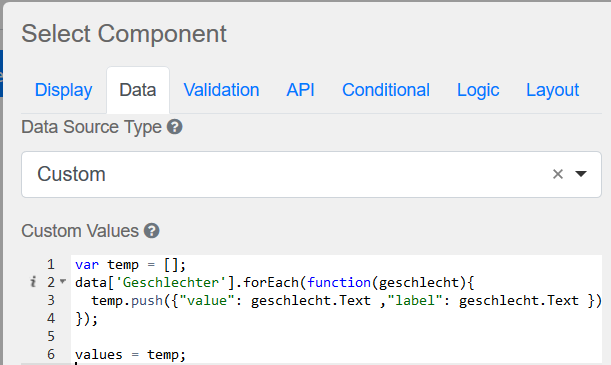
Jetzt erstellen Sie den Dialog, indem Sie im Formular-Editor zunächst ein **Select** in den Arbeitsbereich ziehen.



Ändern Sie das Label auf **Geschlecht**.

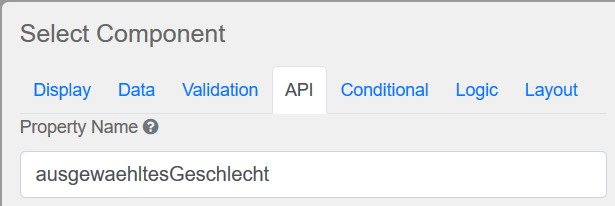


Im Data-Reiter wählen Sie als Data Source Type – **Custom** aus. Damit haben Sie die Möglichkeit ein individuelles Skript zu schreiben, welches auf den übergebenen Datenknoten zugreifen kann.



Das Skript erzeugt zunächst ein leeres Array mit dem Namen temp. Mit data['Geschlechter'].forEach() wird über alle Mehrfachknoten **Geschlechter\*** iteriert. Dieser Iteration wird eine Funktion mit der Variablen geschlecht übergeben, die nacheinander alle Werte von **Geschlechter** enthält. In der Funktion wird nun je Wert ein JSON erzeugt und dem temp-Array hinzugefügt, in welchem value und label jeweils den Inhalt aus dem Unterknoten **Text** zugewiesen bekommen. Das so erzeugt Array wird der Variablen values übergeben, welche dann im Editor die Werte widerspiegelt.

Schlussendlich übergeben Sie dem Datenmodellknoten **ausgewaehltesGeschlecht** das Ergebnis der Auswahl im TONIC-Editor.

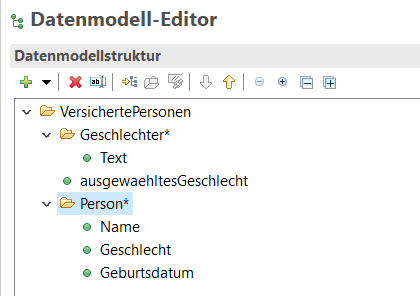


# Mehrfach-Datenknoten im Editor mit Werten füllen

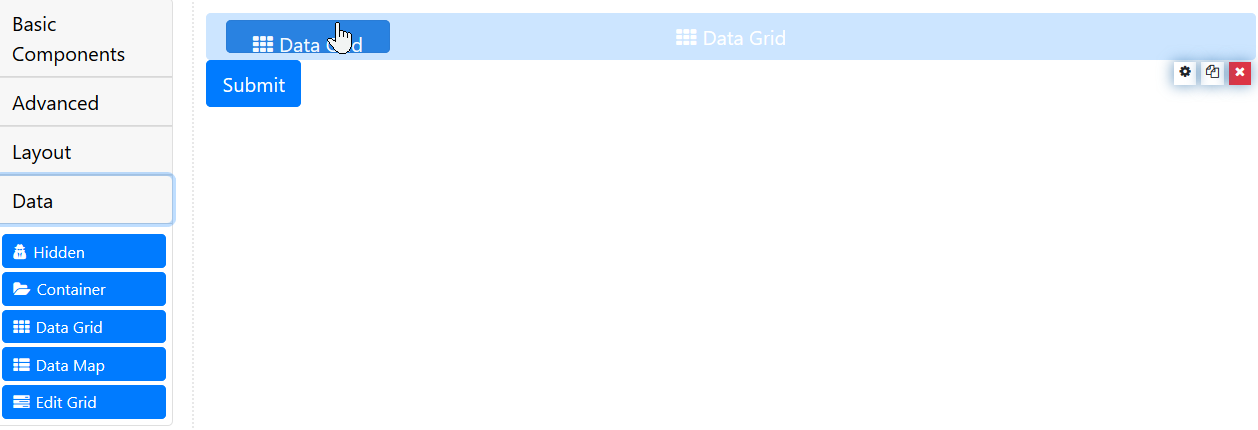
Dem Anwender wird in diesem Beispiel die Möglichkeit gegeben, über den TONIC-Editor eine Werteliste zu füllen, die im Dokument tabellarisch dargestellt werden kann. Dazu wird ein Dialog aufgerufen, in dem die Daten zeilenweise hinzugefügt werden können.



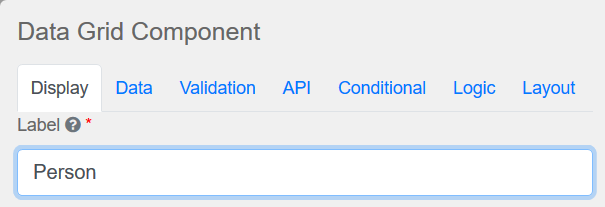
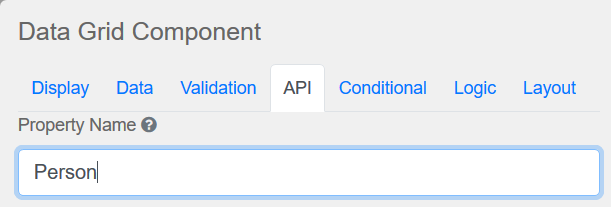
Die Daten sollen anschließend im Mehrfachknoten **Person\*** gespeichert werden.



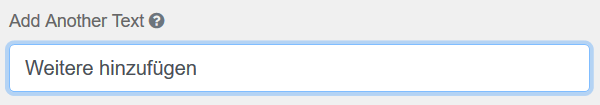
Starten Sie den Dialog-Editor und ziehen Sie aus den Data-Komponenten ein **Data Grid** in den Arbeitsbereich.



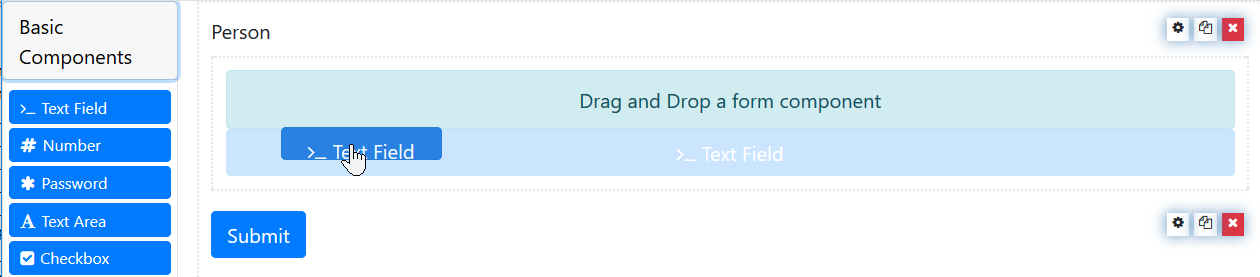
Ändern Sie das Label und API Property Name auf **Person**.

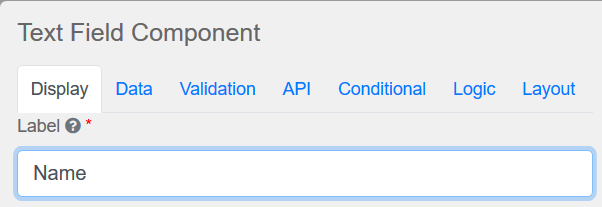
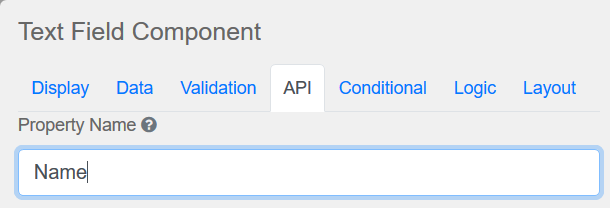
Ändern Sie den Anzeigetext der Schaltfläche, die weitere Zeilen hinzuzufügt. Klicken Sie anschließend auf Save.



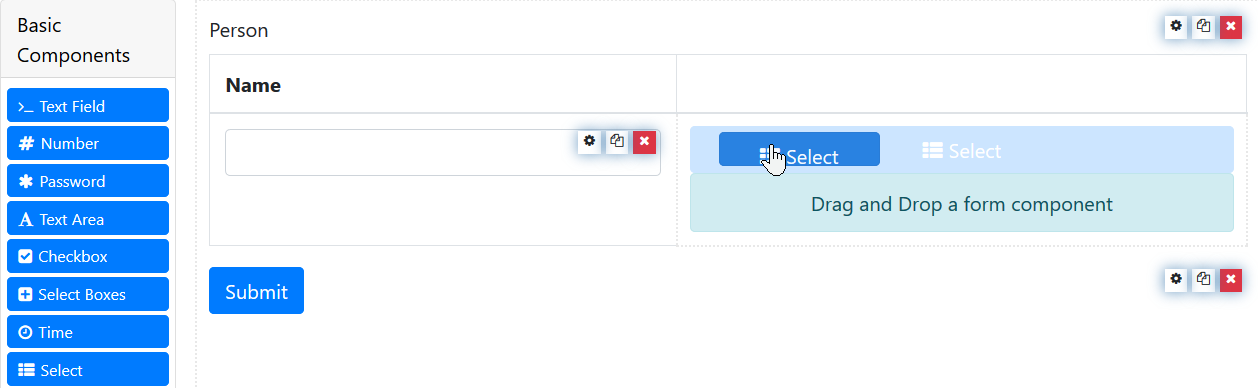
Ziehen Sie nun ein **Textfeld** in das Feld unter Person.



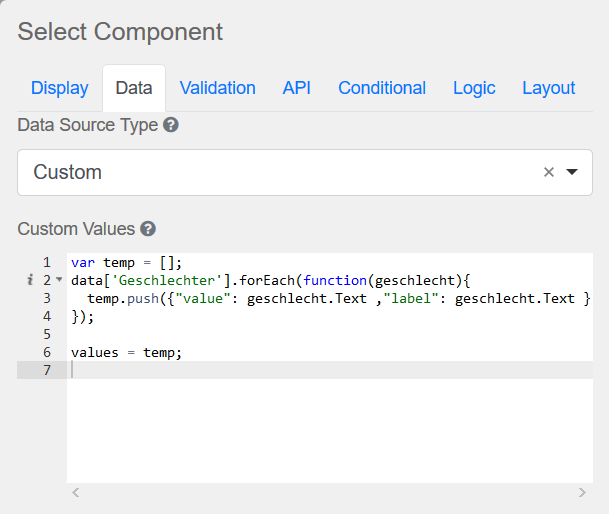
Ändern Sie Label und API Property Name auf **Name**. Klicken Sie anschließend wieder auf Save.

Als nächstes ziehen Sie eine **Select**-Komponente neben das Name-Eingabefeld.



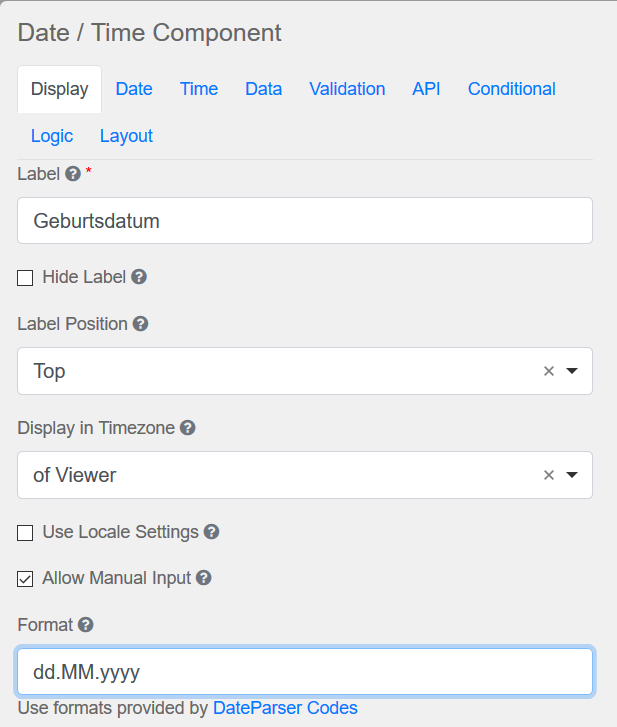
Ändern Sie Label und API auf **Geschlecht**. Wechseln Sie dann auf den Data-Reiter, wählen Sie Custom aus und geben Sie folgendes Skript ein. Dies entspricht dem Beispiel aus dem vorherigen Kapitel.



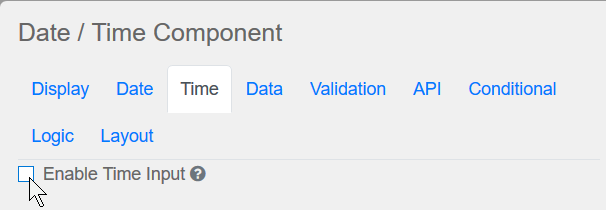
Ziehen Sie nun eine Date/Time-Komponente in den Formularbereich.



Ändern Sie Label und API auf **Geburtsdatum**. Ändern Sie im **Display**-Reiter das Datumsformat.



Schalten Sie im **Time**-Reiter die Eingabe einer Uhrzeit aus und klicken Sie auf Save.



Ändern Sie nun noch das Label der Submit-Schaltfläche auf **Speichern**.

Exportieren Sie anschließend das JSON und kopieren dieses in der Workbench in eine form.io.json-Datei. Testen Sie den Dialog auf dem Server.