Fluechtige Werte in die Datenbank speichern

Das Speichern von Werten, die von Anwendern in Formularen eingegeben werden, geschieht in der Regel über einen entsprechend konfigurierten Button. Werte müssen aber nicht ausschließlich über Formulare in Ihr BPaaS-System gelangen, sie können beispielsweise auch aus einem Fremdsystem übermittelt werden (vgl. Kapitel Integration externer Datenquellen). In diesen Fällen sind die Daten flüchtig: Ohne weitere Modellierung werden sie nur zur Laufzeit der App im Container hinterlegt, aber nicht in die Datenbank gespeichert. Wollen Sie diese temporären Daten dauerhaft in Ihren Kundenbereich speichern, setzen Sie einen Worker ein.

On this Page:

- Werte speichern mit container.save()
- Anwendungsbeispiel

Related Pages:

- Containerwerte beeinflussen
- Integration externer Datenquellen

Related Documentation:

- BPaaS

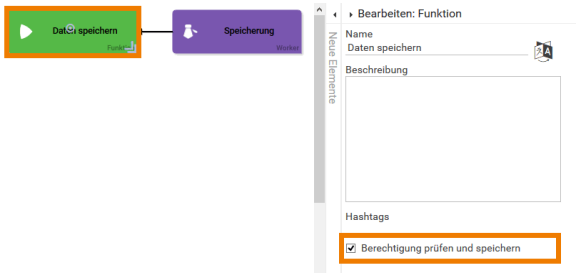
Werte speichern mit `container.save()`

Die Methode `container.save(key, value)` merkt einen Wert zum Speichern vor und schreibt ihn in den Container.

Einen Wert speichern

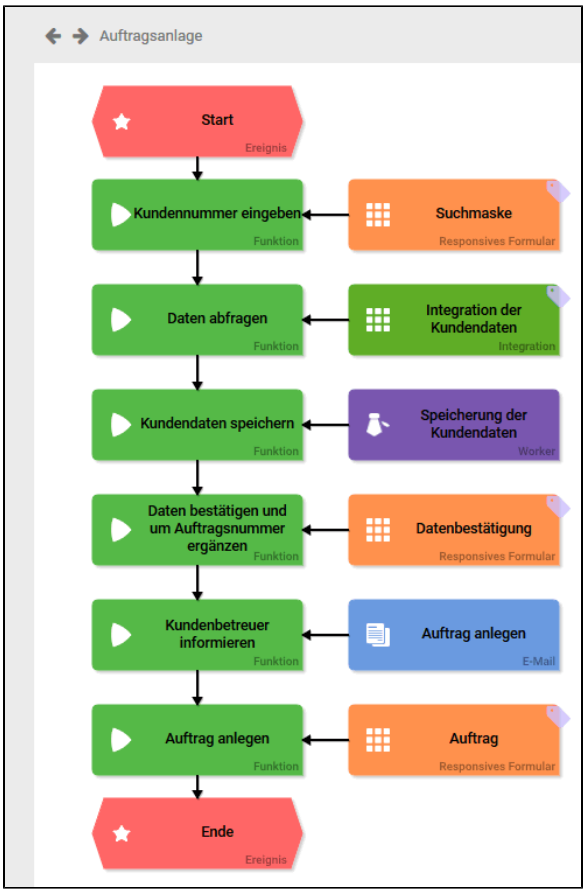
```
// Die Variable "meinBeispiel" mit dem Wert 123 wird zum Speichern markiert

container.save('meinBeispiel', '123');
```

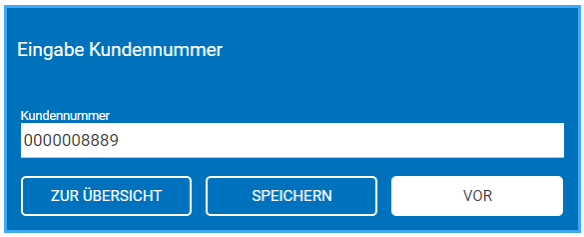
Damit die vorgemerkten Werte dauerhaft in die Datenbank gespeichert werden, muss die Option **Berechtigungen prüfen und speichern** der zugehörigen Funktion aktiviert werden:

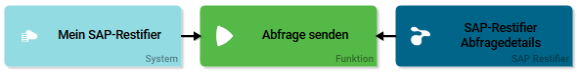


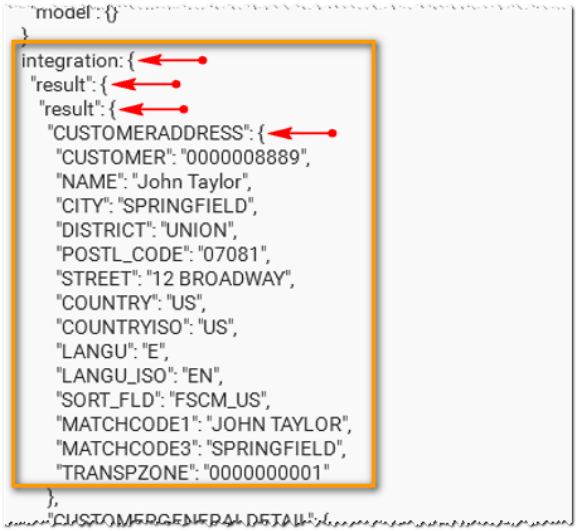
Anwendungsbeispiel

In der App Auftragsanlage soll ein Kundenbetreuer Aufträge anlegen können. Dafür sollen die Kundendaten aus dem firmeneigenen SAP-System verwendet werden, das im Element Integration angebunden wird. Die vom SAP übermittelten Daten sind zunächst flüchtig (vgl. Kapitel Einsatz des SAP-Restifiers), daher werden die Daten nach der Abfrage mit Hilfe eines Workers gespeichert. Im Anschluss kann jederzeit auf die gespeicherten Daten zurück gegriffen werden:

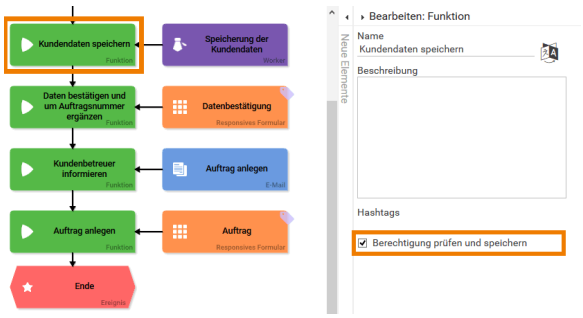
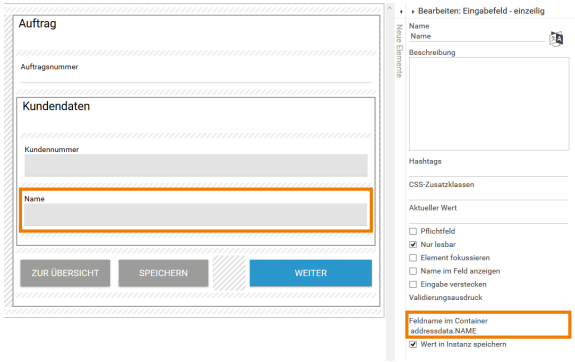


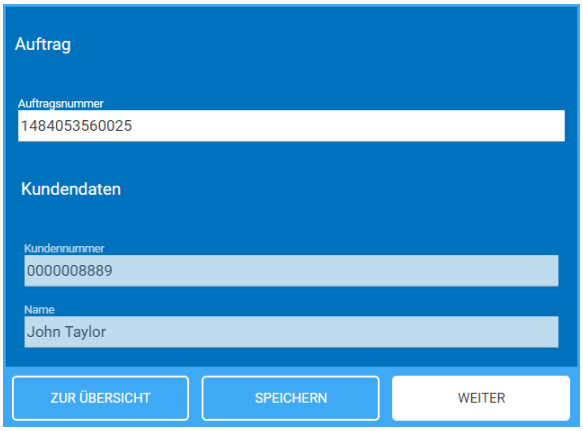
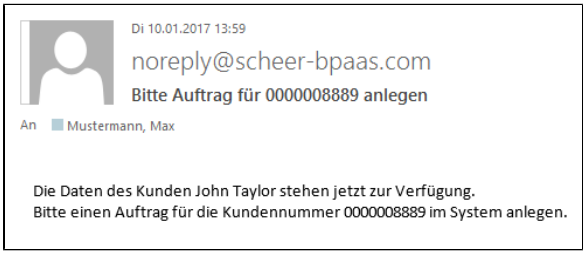
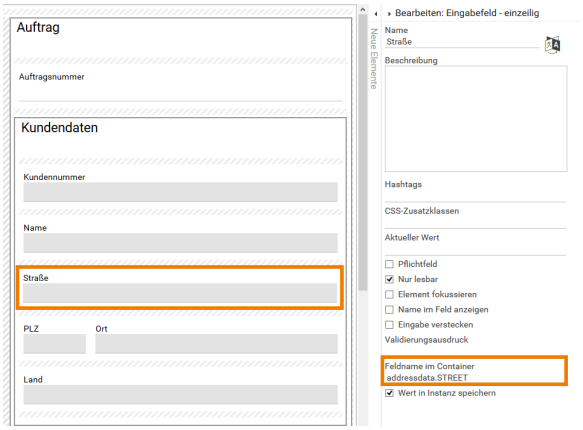
	In der Suchmaske wird die Kundennummer eingegeben. Die Kundennummer wird im Schritt Daten abfragen an das angebundene SAP-System übergeben.

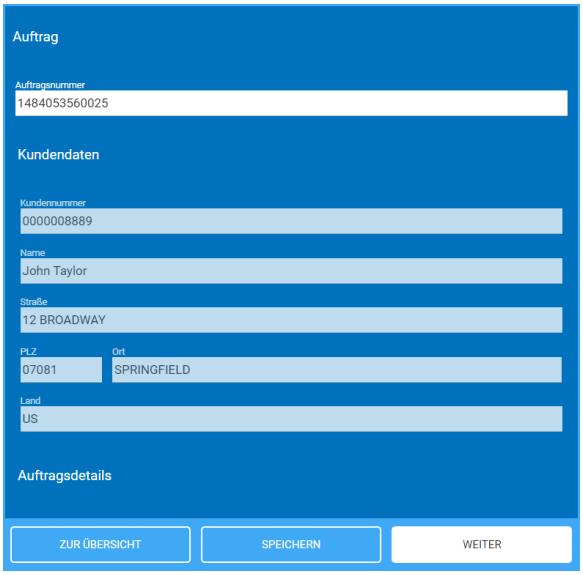
	Innerhalb der **Integration** wird die Anbindung an das SAP-System modelliert. Die Abfrage liefert die zur Kundennummer zugehörigen Kundendaten zurück.

 "model": {}
}
integration: {
"result": {
"result": {
"CUSTOMERADDRESS": {
"CUSTOMER": "0000008889",
"NAME": "John Taylor",
"CITY": "SPRINGFIELD",
"DISTRICT": "UNION",
"POSTL_CODE": "07081",
"STREET": "12 BROADWAY",
"COUNTRY": "US",
"COUNTRYISO": "US",
"LANGU": "E",
"LANGU_ISO": "EN",
"SORT_FLD": "FSCM_US",
"MATCHCODE1": "JOHN TAYLOR",
"MATCHCODE3": "SPRINGFIELD",
"TRANSPZONE": "0000000001"
},
	Die Daten liegen zwar im Container vor, sind aber flüchtig. Werden sie nicht gespeichert, gehen sie wieder verloren.
	
	Die Ergebnisse der Abfrage sind geschachtelt. Für jede geschweifte Klammer muss im Schlüssel des Wertes ein Punkt gesetzt werden.
	
	Beispiel:
	
	Der Name des Kunden steht in der Variable
	
	integration.result.result.CUSTOMERADRESS.NAME

Name und Adresse des Kunden sollen im Verlauf der Auftragsanlage verwendet werden. Im Worker Speicherung der Kundendaten wird daher der komplette Inhalt von CUSTOMERADRESS zum Speichern markiert und in die Variable addressdata geschrieben:

	In den Optionen der Funktion Kundendaten speichern wird die Checkbox **Berechtigungen prüfen und speichern** aktiviert, damit die zum Speichern vorgemerkten Daten nach der Ausführung des Workers in die Datenbank gespeichert werden.
	Im Formular Datenbestätigung soll der Name des Kunden angezeigt werden. Dieser ist gespeichert im Objekt addressdata unter dem Schlüssel NAME. Das Formularfeld Name erhält daher unter **Feldname im Container** den Eintrag addressdata.NAME.

	Nach dem Speichern der Daten wird das Formular Datenbestätigung eingeblendet: Der im Container liegende Name des Kunden wird angezeigt.
	Im Anschluss wird eine E-Mail an den Kundenbetreuer versendet, in der er aufgefordert wird, einen Auftrag anzulegen.
	Wenn der Kundenbetreuer im Schritt Auftrag anlegen das Formular Auftrag öffnet, soll sein flüchtiger Container mit den gespeicherten Instanzdaten gefüllt werden. Dafür erhalten die Felder Straße, PLZ, Ort und Land entsprechende Einträge unter **Feldname im Container**.



Nach dem Öffnen des Formulars Auftrag sind die Adressfelder mit den gespeicherten Daten vorbelegt. Der Kundenbetreuer muss nur noch die Auftragsdetails eingeben.