

Containerwerte beeinflussen

Worker und [Formularfunktionen](#) können eingesetzt werden, um die im Container gespeicherten Werte zu beeinflussen.

Mögliche Anwendungen:

- Werte aus dem Container auslesen
- Werte in den Container schreiben
- Werte im Container modifizieren

Werte in den Container schreiben

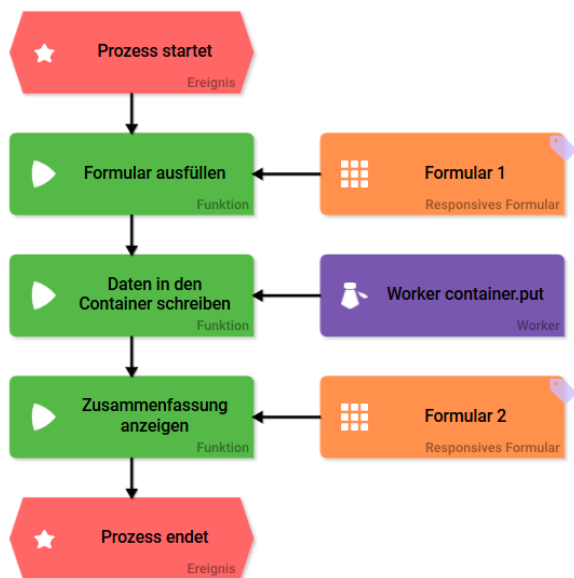
Werte werden mit `container.put(key, value)` in den Container geschrieben.

Einen Wert in den Container schreiben

```
// Die Variable "meinBeispiel" mit dem Wert 123 wird in den Container geschrieben
```

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

Beispiel: Werte über einen Worker in den Container schreiben und anschließend im Formular anzeigen



Ausgangspunkt des Beispiels ist eine einfache EPK mit zwei Formularen und einem Worker.

Hauptbereich

Manuelle Eingabe

ZUR
ÜBERSICHT

SPEICHERN

VOR

In Formular 1 wird vom Anwender eine Manuelle Eingabe erwartet.

On this Page:

- [Werte in den Container schreiben](#)
- [Werte aus dem Container auslesen](#)
- [Containerwerte modifizieren](#)

Related Pages:

- [Formular-Erstellung](#)
 - [Datum, Uhrzeit und Zeitstempel](#)

Related Documentation:

- [BPaaS](#)

Über den Worker werden mit `container.put()` zwei Werte in den Container geschrieben:

- Die Variable **Containerwert1** mit dem Inhalt "Containerwert mit Bezug auf Name des Formularelements."
- Die Variable **wert2** mit dem Inhalt "Containerwert mit Bezug auf Feldname im Container."

Eigenschaften des Workers Worker container.put

```
1 container.put ('Containerwert1', 'Containerwert mit Bezug auf Name des Formularelements.')
```

```
2 container.put ('wert2', 'Containerwert mit Bezug auf Feldname im Container.');
```

SPEICHERN **ABBRECHEN**

Sobald der Worker abgearbeitet ist, stehen im Container die Variablen **Containerwert1** und **wert2** mit ihren jeweiligen Inhalten zur Verfügung. Daher können die nachfolgenden Prozessschritte auf diese Werte zurück gegriffen werden.

Hauptbereich

Manuelle Eingabe

Containerwert1

Containerwert2

ZUR ÜBERSICHT **SPEICHERN** **VOR**

In Formular 2 sollen die Werte angezeigt werden, die über den Worker in den Container geschrieben wurden. Es werden zwei Eingabefelder eingefügt:

- Containerwert1
- Containerwert2

Hauptbereich

Manuelle Eingabe

Containerwert1

Containerwert2

ZUR ÜBERSICHT **SPEICHERN** **VOR**

• Bearbeiten: Eingabefeld - einzellig

Name Containerwert1

Beschreibung

Hashtags

CSS-Zusatzklassen

Aktueller Wert

☐ Pflichtfeld

☐ Nur lesbar

☐ Element fokussieren

☐ Name im Feld anzeigen

☐ Eingabe verstecken

Validierungsausdruck

Feldname im Container

☒ Wert in Instanz speichern

Das erste Eingabefeld erhält den Namen **Containerwert1**, dessen **Feldname im Container** bleibt leer. Dieses Eingabefeld referenziert daher auf den Wert im Container, der unter **Containerwert1** gespeichert ist.

Hauptbereich

Manuelle Eingabe

Containerwert1

Containerwert2

ZUR ÜBERSICHT **SPEICHERN** **VOR**

• Bearbeiten: Eingabefeld - einzellig

Name Containerwert2

Beschreibung

Hashtags

CSS-Zusatzklassen

Aktueller Wert

☐ Pflichtfeld

☐ Nur lesbar

☐ Element fokussieren

☐ Name im Feld anzeigen

☐ Eingabe verstecken

Validierungsausdruck

Feldname im Container wert2

☒ Wert in Instanz speichern

Das zweite Eingabefeld mit dem Namen **Containerwert2** erhält den **Feldname im Container** **wert2**. Das Eingabefeld referenziert daher auf den Wert, der im Container unter dem Namen **wert2** abgelegt wird.

Hauptbereich

Manuelle Eingabe

Testeingabe

Zur Übersicht **Speichern** **Vor**

Beim Ausführen der App wird zunächst die **Manuelle Eingabe** getätigt.

Der Klick auf **Vor** stößt den nächsten Prozessschritt an. Der Worker wird ausgeführt, die Variablen **Containerwert1** und **wert2** werden in den Container geschrieben.

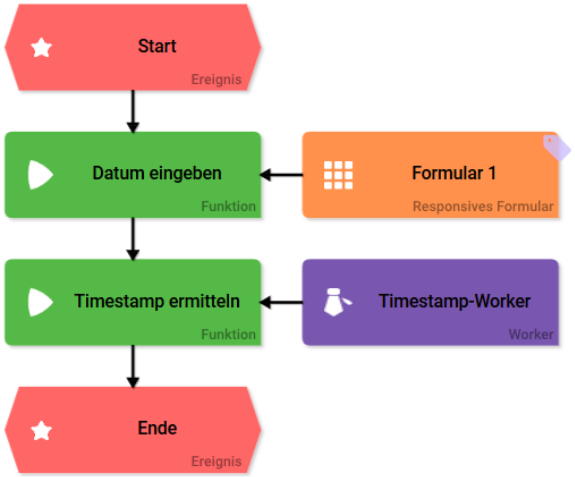
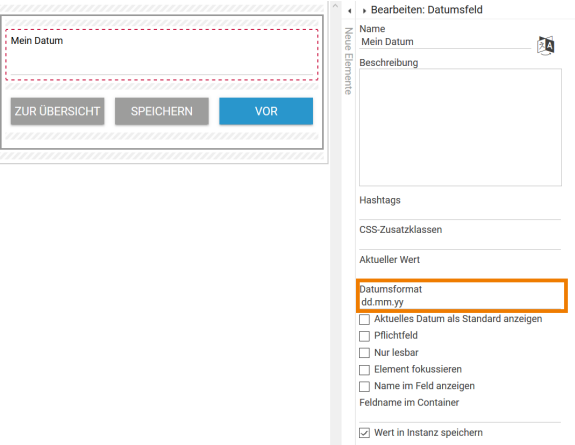
<div data-bbox="159 136 730 420"> <div>Hauptbereich</div> <div> <div>Manuelle Eingabe</div> <div>Testeingabe</div> </div> <div> <div>Containerwert1</div> <div>Containerwert mit Bezug auf Name des Formularelements.</div> </div> <div> <div>Containerwert2</div> <div>Containerwert mit Bezug auf Feldname im Container.</div> </div> <div> <div>Zur Übersicht</div> <div>Speichern</div> <div>Vor</div> </div> </div>	<p>In Formular 2 wird nicht nur der Inhalt der manuellen Eingabe angezeigt, sondern auch die in den Container geschriebenen Werte der beiden Variablen.</p>
--	---

Werte aus dem Container auslesen

Werte werden mit `container.get(key)` aus dem Container ausgelesen.

Einen Wert aus dem Container auslesen
<pre>// Der Wert der Variable "meinBeispiel" wird aus dem Container ausgelesen container.get('meinBeispiel');</pre>

Beispiel: Ein gespeichertes Datum wird ausgelesen und in einem Pop-up-Fenster als Timestamp angezeigt

 <pre> graph TD Start([Start Ereignis]) --> Datum[Datum eingeben Funktion] Datum --> Timestamp[Timestamp ermitteln Funktion] Timestamp --> Ende([Ende Ereignis]) Formular1[Formular 1 Responsives Formular] --> Datum Worker[Timestamp-Worker Worker] --> Timestamp </pre>	<p>Grundlage für das Beispiel ist eine EPK mit zwei Prozessschritten.</p>
	<p>In Formular 1 wurde das Datumsfeld Mein Datum eingefügt</p> <p>Ein Datum wird intern immer als Millisekundenwert gespeichert, die Ausgabe in einem Datumsfeld erfolgt aber im gewünschten Datumsformat.</p> <div data-bbox="976 1528 1498 1644"> <p>✓ Hilfreiche Hinweise für die Verwendung von Datumsformaten haben wir auf der Seite Datum, Uhrzeit und Zeitstempel zusammengestellt.</p> </div>

Über den Timestamp-Worker soll der Inhalt von **Mein Datum** ausgelesen und in die Variable **myTimestamp** geschrieben werden. Der Inhalt von **myTimestamp** wird anschließend in einem Pop-up-Fenster ausgegeben (weitere Informationen dazu finden Sie auf der Seite [Einen Alert einbauen](#)):

```
Eigenschaften des Workers Timestamp-Worker
1 vor myTimestamp = container.get('Mein Datum');
2 alertLog(myTimestamp);
```

SPEICHERN **ABBRECHEN**

Mein Datum

Zur Übersicht **Speichern** **Vor**

< Sep 2004 >

Wo	Mo	Di	Mi	Do	Fr	Sa	So
36			1	2	3	4	5
37	6	7	8	9	10	11	12
38	13	14	15	16	17	18	19
39	20	21	22	23	24	25	26
40	27	28	29	30			

Heute **Löschen** **Abbrechen**

Während der Ausführung der App gibt der Anwender ein Datum ein.

Mein Datum

Zur Übersicht **Speichern** **Vor**

Das Datum wird im angegebenen Format (dd.mm.yy) angegeben. Mit dem Klick auf **Vor** wird der Inhalt des Feldes unter **Mein Datum** in den Container geschrieben und die Ausführung des Workers wird angestoßen.

1095285600000

OK

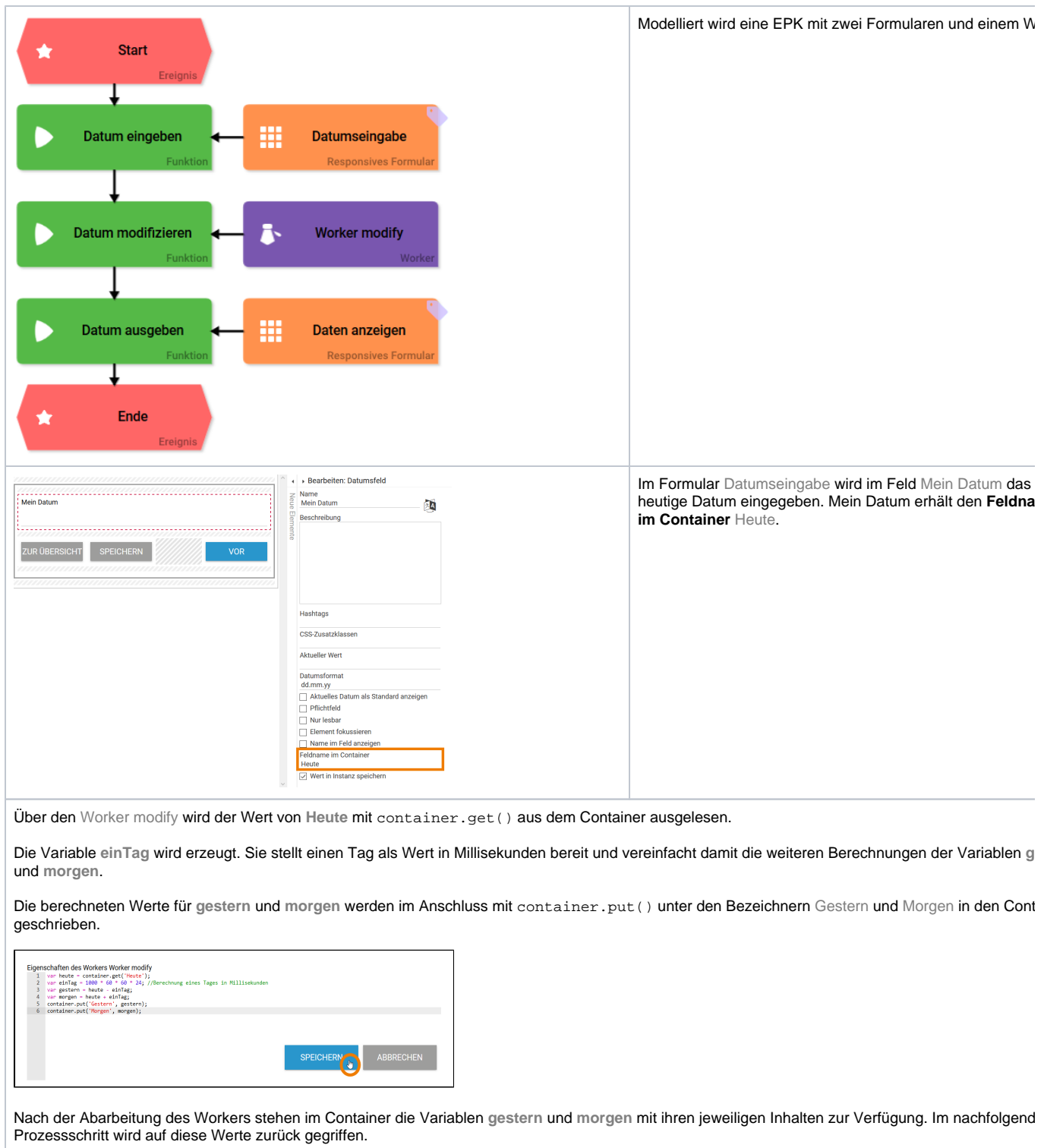
Der Inhalt von **Mein Datum** wird in die Variable **myTimestamp** geschrieben und **myTimestamp** in einem Pop-up-Fenster eingeblendet.


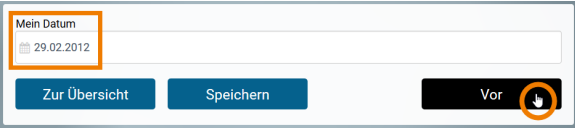

Containerwerte modifizieren

Das Modifizieren von Werten im Container ist eine Kombination aus Lesen und Schreiben: Zunächst werden Werte ausgelesen, um diese anschließend zu ändern und dann in den Container zurückzuschreiben.

i Enthält eine Variable bereits einen Wert, wird durch Einsatz von `container.put()` der alte Wert mit dem neuen überschrieben.

Beispiel: Ein gespeichertes Datum wird modifiziert



	<p>Das Formular Daten anzeigen enthält die drei Datumsfelder</p> <ul style="list-style-type: none"> • Gestern • Heute • Morgen
	<p>Die App wird ausgeführt, ein Datum wird eingegeben.</p> <p>Der Klick auf Vor startet den nächsten Prozessschritt, der v modify führt die Berechnungen durch.</p>
	<p>Mein Datum wird im Feld Heute angezeigt.</p> <p>Gestern zeigt das Datum von Heute minus einen Tag, im Morgen wird das Datum von Heute plus einen Tag eingeblendet.</p>

Experten-Tipp

Wenn ausgeschlossen werden soll, dass Werte überschrieben werden, sollte man vor dem Schreiben von Werten prüfen, ob die Variable leer ist oder nicht.