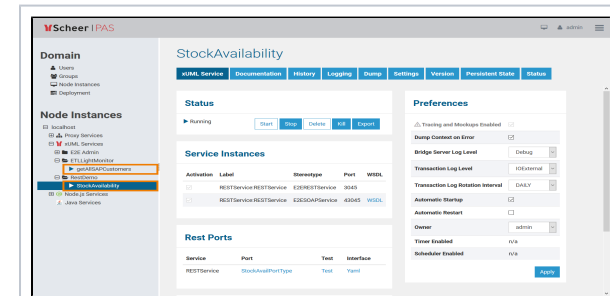


# Einsatz des Plattform Service

Das Element Plattform Service ( PAS 18.1) ist eine Erweiterung des REST-Elements und speziell darauf abgestimmt, um via REST-Protokoll einen xUML-Service der **Scheer PAS BRIDGE** aufzurufen.

## Beispiel: Daten aus dem SAP auslesen und konvertieren

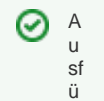
Max Mustermann möchte in einer Business App in **Scheer PAS BPaaS** Daten aus einem SAP-System verwenden. Die gewünschten Daten zu Waren und Kunden liegen zwar im SAP vor, aber noch nicht im benötigten Format. Da Max Mustermann aber auch das Integrationsmodul **Scheer PAS BRIDGE** im Einsatz hat, kann er einen xUML-Service modellieren, der die Daten abrufen und in das gewünschte Format konvertiert. Die konvertierten Daten kann er anschließend über das Element Plattform Service abrufen und in BPaaS weiterverwenden.



Zunächst modelliert Max außerhalb von BPaaS die benötigten xUML Services:

- [getAllSAPCustomers](#) ruft die Kundendaten aus dem SAP ab und konvertiert sie ins gewünschte Datenformat.
- [StockAvailability](#) ruft Daten zu Waren verfügbarkeit aus dem SAP ab und konvertiert sie ins gewünschte Datenformat.

Nachdem Max die Services deployed hat, sind sie in seiner BRIDGE verfügbar.



### On this Page:

- [Beispiel: Daten aus dem SAP auslesen und konvertieren](#)

### Related Pages:

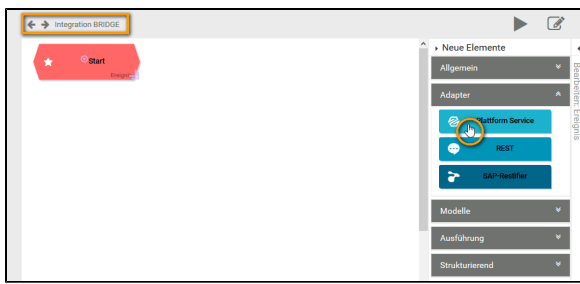
- [Worker einsetzen](#)
  - [Flüchtige Werte in die Datenbank speichern](#)
- [Services](#)
  - [Downloads](#)

### Related Documentation:

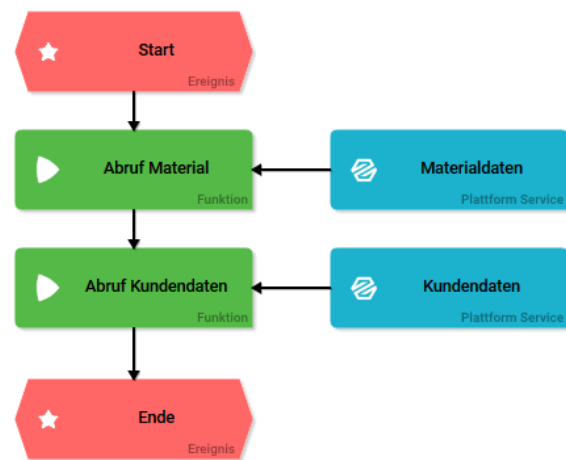
- [BPaaS](#)
- [BRIDGE \(engl.\)](#)
  - [xUML Services Reference Guide](#)



	<div> <div>R</div> <div>I</div> <div>D</div> <div>G</div> <div>E</div> <div>D</div> <div>o</div> <div>k</div> <div>u</div> <div>m</div> <div>e</div> <div>n</div> <div>t</div> <div>a</div> <div>t</div> <div>i</div> <div>o</div> <div>n</div> <div>(</div> <div>e</div> <div>n</div> <div>g</div> <div>l</div> <div>.)</div> </div>
<div> <div> <div> <div></div> <div>Order Management</div> <div>App Projekt</div> </div> </div> </div>	<p>Anschließend loggt sich Max in BPaaS ein und legt das App Projekt Order Management an.</p>
<div> <div> <div> <div>★</div> <div>Kunde will eine Bestellung aufgeben</div> <div>Ereignis</div> </div> <div>↓</div> <div> <div>▶</div> <div>Services aufrufen</div> <div>Funktion</div> </div> <div>←</div> <div> <div> <div></div> <div>Integration BRIDGE</div> <div>Integration</div> </div> </div> </div> </div>	<p>Hier modelliert er nun seine EPK zur Aufgabe von Bestellungen. Will ein Kunde eine Bestellung aufgeben, sollen zunächst die Waren- und Kundendaten abgerufen werden, damit sie dem Kunden im Bestellformular bereits zur Verfügung stehen.</p> <p>Deshalb modelliert Max zuerst eine Integration zu seiner BRIDGE.</p>

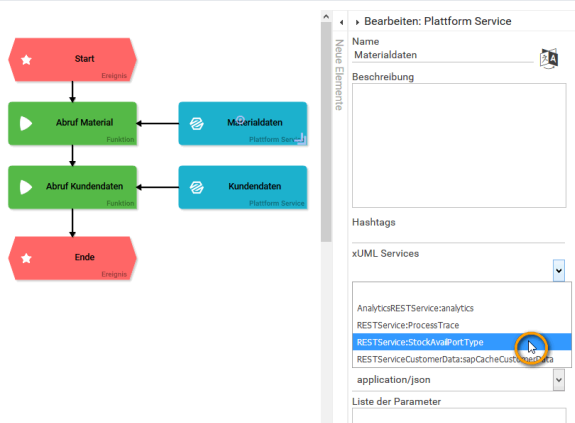


Das Element Plattform Service findet sich ausschließlich innerhalb einer Integration im Menüpunkt **Adapter**.



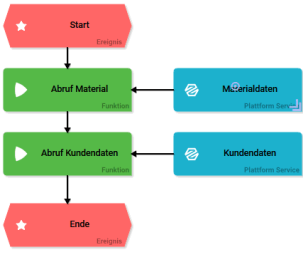
Innerhalb des Elements Integration BRIDGE modelliert Max nun eine kleine EPK: Zunächst werden die Materialdaten abgerufen. In einem zweiten Schritt werden die Kundendaten abgerufen.

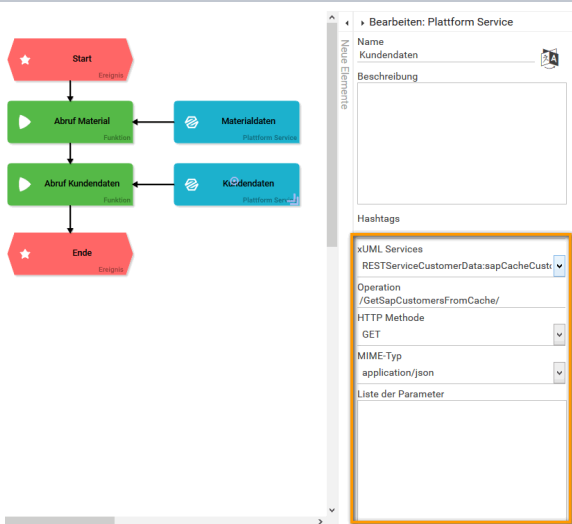
Für die Abrufe der Daten verwendet Max jeweils ein Plattform Service Element.



In den Eigenschaften des Plattform Service Elements Materialdaten findet Max die Option **xUML Services**. Hier werden alle xUML-Services mit REST-Schnittstelle angezeigt, die in Max BRIDGE verfügbar sind.

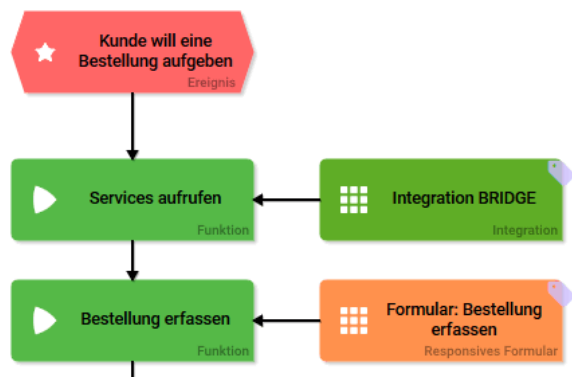
Er wählt für die Materialdaten den REST Service: StockAvailPortType aus.

 <div data-bbox="472 132 727 552"> <p>Bearbeiten: Plattform Service</p> <p>Neue Elemente</p> <p>Name Materialdaten</p> <p>Beschreibung</p> <p>Hashtags</p> <p>xUML Services RESTService:StockAvailPortType</p> <p>Operation /StockInfo/</p> <p>HTTP Methode GET</p> <p>MIME-Typ application/json</p> <p>Liste der Parameter</p> </div>	<p>Nun muss <b>Max</b> noch den REST-Pfad der Operation eingeben.</p> <p>Da <b>Max</b> den xUML-Service selbst modelliert hat, weiß er, dass er hier <b>/StockInfo/</b> eingeben muss.</p>
 <div data-bbox="472 562 727 1098"> <p>Bearbeiten: Plattform Service</p> <p>Neue Elemente</p> <p>Name Materialdaten</p> <p>Beschreibung</p> <p>Hashtags</p> <p>xUML Services RESTService:StockAvailPortType</p> <p>Operation /StockInfo/</p> <p>HTTP Methode GET</p> <p>MIME-Typ application/json</p> <p>Liste der Parameter</p> </div>	<p><b>Max</b> möchte mit dem Plattform Service Daten abrufen, die vorausgewählte <b>HTTP Methode GET</b> muss er daher nicht verändern.</p> <p>Auch den <b>MIME-Typ application/json</b> kann er unverändert lassen.</p> <p>Weitere Parameter muss <b>Max</b> ebenfalls nicht mitgeben, so kann er die <b>Liste der Parameter</b> leer lassen.</p> <p>Schon ist der erste Plattform Service fertig konfiguriert.</p>



Für das Plattform Service Element Kundendaten wählt Max entsprechend eine andere Konfiguration:

- **xUML Service:** Rest Service CustomerData: sapCacheCustomerData
- **Operation:** /GetSapCustomersFromCache/
- **HTTP Methode:** GET
- **MIME-Typ:** application/json
- **Liste der Parameter:** leer, da nicht benötigt



Die abgerufenen Daten sollen im Bestellformular angezeigt werden. Also hängt Max an den nächsten Prozessschritt das Formular: Bestellung erfassen.

← → Formular: Bestellung erfassen

### Kopf der Bestellung

Kunde

Ersteller

Erstelldatum

### Kundendaten

Kundenfirma

Strasse

PLZ

Stadt

Land

Germany

### Bestellposition

Materialen suchen

Gesamtpreis

Positionen

Materialnamen	Material	Anzahl	Preis

Bearbeiten: Suchfeld

Name

Customer

Beschreibung

Hashtags

CSS-Zusatzklassen

☐ Pflichtfeld
 ☐ Nur lesbar
 ☐ Element fokussieren
 ☐ Name im Feld anzeigen

Feldname im Container

searchValue1

Containername für mögliche Werte

searchValue1

☒ Wert in Instanz speichern

ZURÜCK ZUR ÜBERSICHT

SPEICHERN

WEITER

Im Formular : Bestellung erfassen möchte Max die abgerufenen Daten wie folgt verwenden:

Im Suchfeld Kunde sollen die abgerufenen Kundendaten verfügbar sein, so dass Kunden mit Eingabe eines Suchbegriffes Ihre vorhandenen Kundendaten übernehmen können.

Im Suchfeld Materialien suchen sollen die verfügbaren Waren durchsucht und die gewünschten Daten für die Bestellung übernommen werden können.

Formular: Bestellung erfassen

Responsives Formular

Bearbeiten: Responsives Formular

Name

Form: Create order

Beschreibung

Hashtags

CSS-Modell

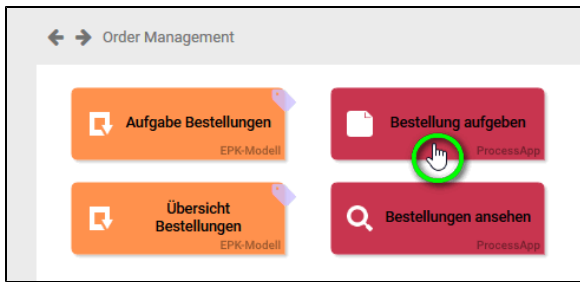
Entwickler-Optionen

Funktion: Vor der Formulareausführung

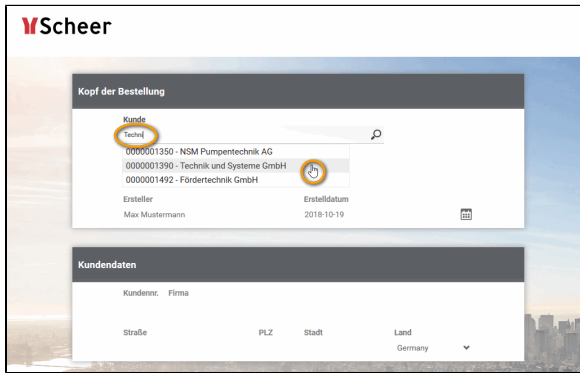
Funktion: Nach der Formularanzeige

Funktion: Nach der Formulareausführung

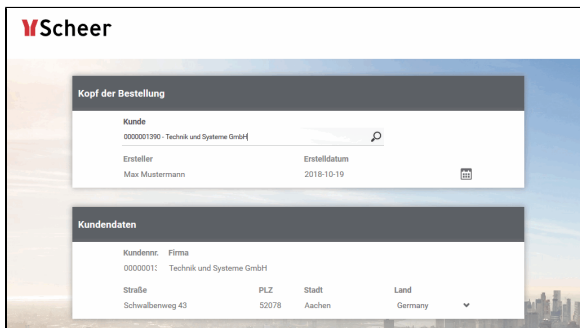
Über entsprechen des Coding in der **Formularfunktion: Vor der Formulareausführung** werden die abgerufenen Daten zur Befüllung der Suchfelder bereit gestellt.



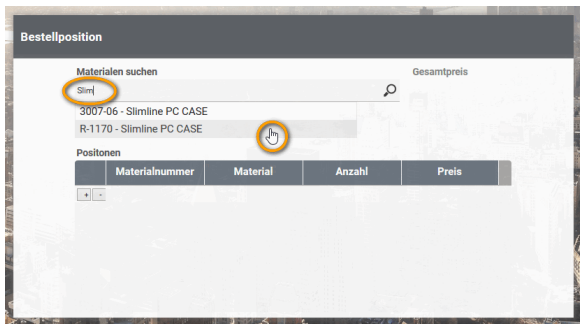
Nun kann **Max** testen, ob die abgerufenen Daten im Formular verfügbar sind. Er startet eine Neuanlage.



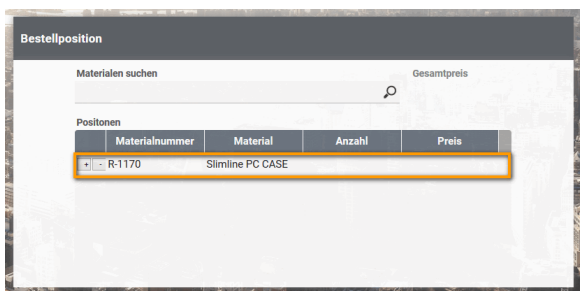
Im Suchfeld **Kunde** gibt er einen Suchbegriff ein. Es werden entsprechende Treffer angezeigt und **Max** kann den gewünschten Kundendatensatz auswählen.



Die Daten des ausgewählten Kunden werden in den Abschnitt **Kundendaten** übernommen.



Auch im Abschnitt **Bestellposition** funktioniert der Datenabruf: **Max** kann im Feld **Materialien suchen** einen Suchbegriff eingeben und erhält entsprechende Treffer.



Der ausgewählte Datensatz wird in die Tabelle **Positionen** übernommen.