

# Einsatz von REST

Das REST-Element setzen Sie ein, wenn Sie via REST-Protokoll Daten mit einem Fremdsystem austauschen möchten. Das REST-Element kann nur gemeinsam mit dem Element **System** eingesetzt werden: Während im REST-Element die eigentliche Abfrage formuliert wird, enthält das Element System die Angabe zum Adressat der Abfrage.

## Statische lesende Abfrage mit REST

Über eine REST-Abfrage sollen von einem Fremdsystem Benutzerdaten abgerufen werden.

### On this Page:

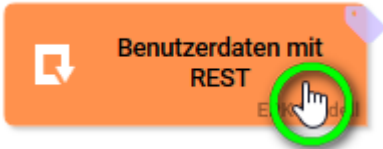
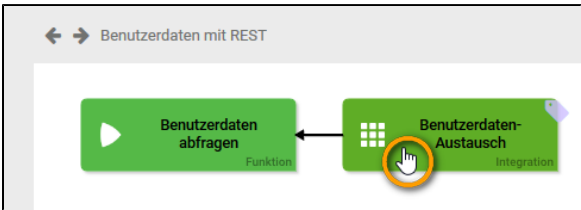
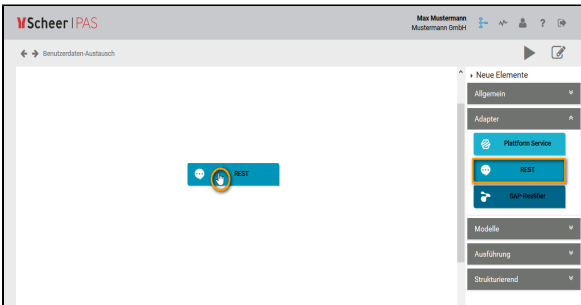
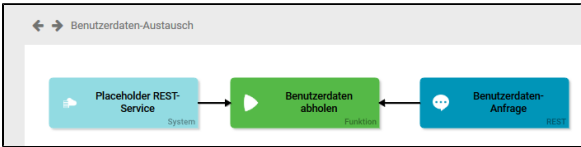
- [Statische lesende Abfrage mit REST](#)
- [Schreibende Abfrage mit REST](#)

### Related Pages:

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

### Related Documentation:

- [BPaaS](#)

	<p>Zunächst wird ein EPK-Modell angelegt und geöffnet.</p>
	<p>Innerhalb des EPK-Modells wird eine einfache EPK angelegt.</p> <p>Diese enthält die Funktion <i>Benutzerdaten abfragen</i> und die mit der Funktion verbundene Integration <i>Benutzerdaten-Austausch</i>.</p> <p>Um das REST-Element verwenden zu können, muss die Integration geöffnet werden.</p>
	<p>Nur der Editor innerhalb der Integration enthält das Element REST, das nun auf der Arbeitsfläche angelegt werden kann.</p>
	<p>Benötigt wird eine kleine EPK mit System, Funktion und REST-Element.</p>

Placeholder REST-Service

System

Neue Elemente

Bearbeiten: System

Name

Placeholder REST-Service

Beschreibung

jsonplaceholder.typicode.com ist eine kostenlose, frei zugängliche Attrappe, die genutzt werden kann, um zu Testzwecken REST-Anfragen zu senden.

Hashtags

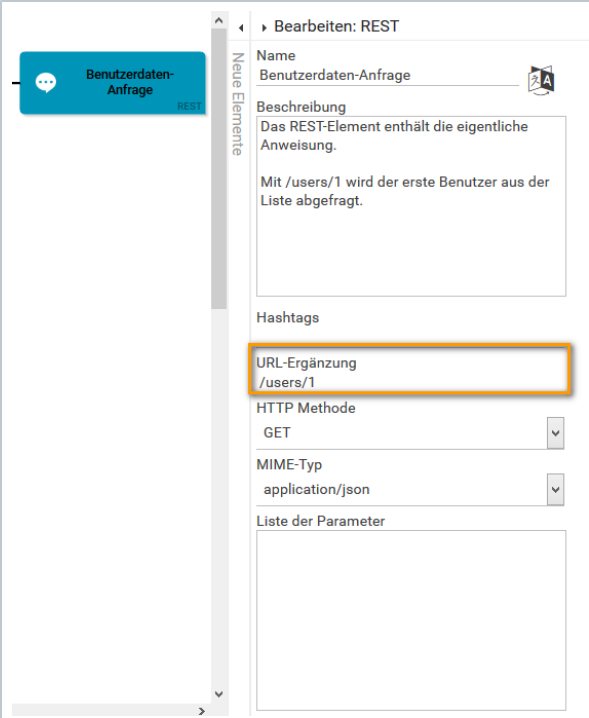
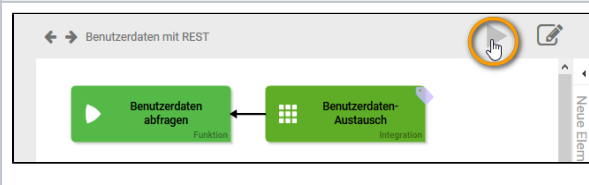
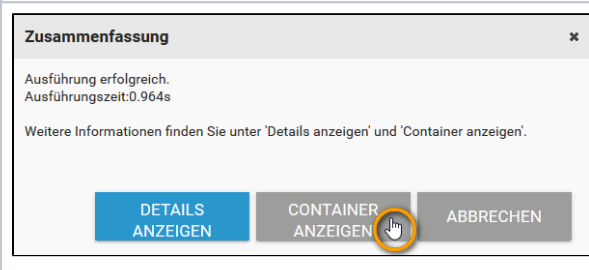
URL

http://jsonplaceholder.typicode.com

URL (Intern)

Im System wird im Feld **URL** die Adresse des Fremdsystems eingetragen, das angesprochen werden soll.

Im Beispiel wird <http://jsonplaceholder.typicode.com> verwendet. Dabei handelt es sich um eine kostenlose, frei zugängliche Attrappe, die genutzt werden kann, um zu Testzwecken REST-Anfragen zu senden.

	<p>Das REST-Element enthält die eigentliche Anweisung. Im Feld <b>URL-Ergänzung</b> wird die Angabe zur Ressource eingetragen, die man abfragen möchte.</p> <p>Im Beispiel wird mit <code>/users/1</code> die Ressource Nummer 1 aus dem Thema Users geholt - also der erste Benutzereint rag.</p> <div data-bbox="980 783 1094 1289" style="border: 1px solid green; padding: 5px; margin-top: 10px;"> <p><b>Expert en-Tipp</b></p> <p>Der Eintrag im Feld <b>URL-Ergänzung</b> kann auch einen zu evaluierenden Wert enthalten.</p> </div>
	<p>Über den Play-Button kann die Integration nun getestet werden.</p>
	<p>Nach der Ausführung kann man über den Button <b>Container anzeigen</b> die Werte im Container ansehen.</p>

Inhalt des Containers

```
},
"statusCode": 200,
"body": {
  "id": 1,
  "name": "Leanne Graham",
  "username": "Bret",
  "email": "Sincere@april.biz",
  "address": {
    "street": "Kulas Light",
    "suite": "Apt. 556",
    "city": "Gwenborough",
    "zipcode": "92998-3874",
    "geo": {
      "lat": "-37.3159",
      "lng": "81.1496"
    }
  },
  "phone": "1-770-736-8031 x56442",
  "website": "hildegard.org",
  "company": {
    "name": "Romaguera-Crona",
    "catchPhrase": "Multi-layered client-server neural-net",
    "bs": "harness real-time e-markets"
  }
},
}
meta: {
  "finished": true
}
```

ABBRECHEN

Wenn man zum Ende des Containers scrollt, wird dort das Resultat der Integration angezeigt.

Im Beispiel sind die abgefragten Benutzerdaten von User 1 zu sehen.

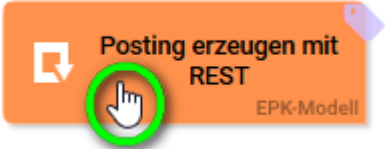
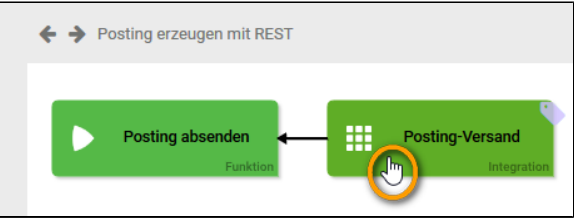
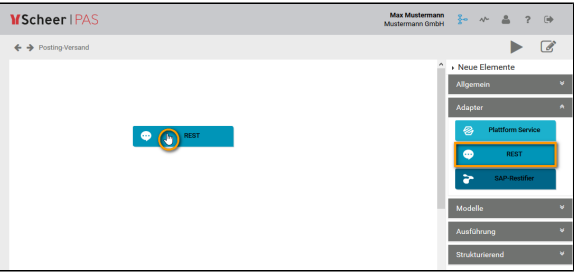


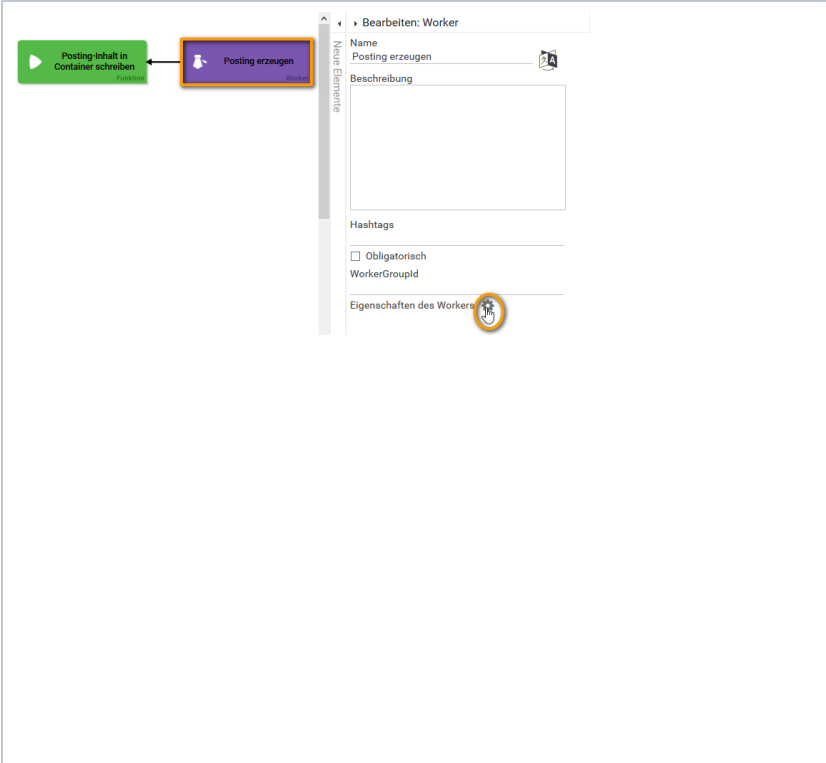
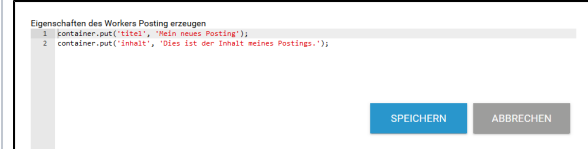
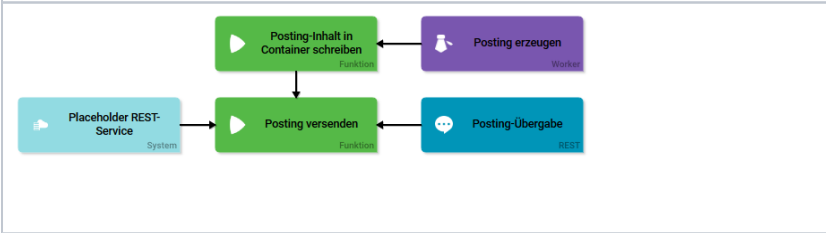
Ohne weitere Modelle der Lieferungen sind die Daten für die Erstellung der Container nicht möglich, das heißt es werden in dem Container nur hinterlegt,



## Schreibende Anfrage mit REST

Mit REST können Sie nicht nur Daten abrufen, sondern auch Daten in das System übergeben, das sich hinter der REST-Schnittstelle befindet.

	<p>Ausgangspunkt ist wieder ein EPK-Modell. Diesmal sollen keine Daten abgefragt werden. Stattdessen soll ein Posting an das Fremdsystem übergeben werden.</p>
	<p>Im EPK-Modell Posting erzeugen mit REST werden die Funktion Posting absenden und die Integration Posting-Versand angelegt.</p> <p>Um das REST-Element verwenden zu können, muss die Integration geöffnet werden.</p>
	<p>Nur der Editor innerhalb der Integration enthält das Element REST, das nun auf der Arbeitsfläche angelegt werden kann.</p>

	<p>Um einen Inhalt übergeben zu können, muss der entsprechende Inhalt zunächst im Container erzeugt werden. Im Beispiel werden Titel und Inhalt des Postings über einen Worker erstellt.</p> <p>Die EPK besteht aus dem Worker Posting erzeugen, der verbunden ist mit der Funktion Posting-Inhalt in Container schreiben.</p>
<p>Im ersten Schritt wird also über den Worker Inhalt generiert:</p> <pre>container.put('titel', 'Mein neues Posting'); container.put('inhalt', 'Dies ist der Inhalt meines Postings.');</pre> <p>Der Code wird unter <b>Eigenschaften des Workers</b> gespeichert:</p> 	
	<p>Im zweiten Schritt Posting versenden werden System- und REST-Element angebunden.</p>

Placeholder REST-Service

System

Bearbeiten: System

Name  
Placeholder REST-Service

Beschreibung  
jsonplaceholder.typicode.com ist eine kostenlose, frei zugängliche Attrappe, die genutzt werden kann, um zu Testzwecken REST-Anfragen zu senden.

Hashtags

URL  
http://jsonplaceholder.typicode.com

URL (Intern)

Im Feld **URL** des System-Elements wird die Adresse des Fremdsystems eingetragen, das angesprochen werden soll.

Im Beispiel wird <http://jsonplaceholder.typicode.com> verwendet. Dabei handelt es sich um eine kostenlose, frei zugängliche Attrappe, die genutzt werden kann, um zu Testzwecken REST-Anfragen zu senden.

Posting-Übergabe

REST

Bearbeiten: REST

Name  
Posting-Übergabe

Beschreibung  
Das ReST-Element enthält die eigentliche Anweisung.  
  
Mit dem Worker "Posting erzeugen" werden Titel und Inhalt des Postings im Container angelegt.  
  
Die Inhalte der neu anzulegenden Ressource werden über POST gesendet.

Hashtags

URL-Ergänzung  
/posts

HTTP Methode  
POST

MIME-Typ  
application/json

Liste der Parameter  
{  
  "title" : container.get('titel'),  
  "body" : container.get('inhalt'),  
  "userld" : 1  
}

Im REST-Element wird die eigentliche Anweisung definiert. Die Angabe im Feld **URL-Ergänzung** definiert das Thema, für das man einen neuen Eintrag übermitteln möchte.

Im Beispiel wird für das Thema `/posts` ein neuer Eintrag übergeben. Entsprechend wird als **HTTP Methode** POST ausgewählt.

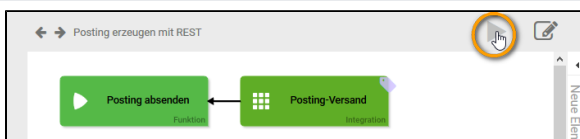


### Expert en- Tipp

Der  
MIME-  
Typ  
muss  
an  
dieser  
Stelle  
an die  
Spezifik  
ationen  
der  
REST-  
Schnitt  
stelle  
angepa  
sst  
werden.

Da es sich  
um eine  
schreibende  
Anfrage  
handelt,  
enthält die **L  
iste der  
Parameter**  
die  
Angaben  
zum Inhalt:

```
{  
  
  "title"  
  :  
  contain  
er.get  
( 'titel  
' ),  
  
  "body"  
  :  
  contain  
er.get  
( 'inhal  
t' ),  
  
  "userId  
" : 1  
}
```



Über den  
Play-Button  
kann die  
Integration  
nun  
getestet  
werden.

<div> <div>Zusammenfassung</div> <div> <p>Ausführung erfolgreich.</p> <p>Ausführungszeit:0.839s</p> <p>Weitere Informationen finden Sie unter 'Details anzeigen' und 'Container anzeigen'.</p> <div> <div>DETAILS ANZEIGEN</div> <div>CONTAINER ANZEIGEN</div> <div>ABBRECHEN</div> </div> </div> </div>	<p>Nach der Ausführung öffnet man über den Button <b>Container anzeigen</b> den Container.</p>
<div> <div>Inhalt des Containers</div> <div> <pre> {   },   "executeCurrentStep": true }, "execution": {} } useDefaultDecider: true titel: Mein neues Posting inhalt: Dies ist der Inhalt meines Postings. integration: {   "result": {     "headers": {       "date": "Mon, 22 Oct 2018 10:00:40 GMT",       "content-type": "application/json; charset=utf-8",       "content-length": "113",       "set-cookie": [         "__cfduid=dfa93266f2d15440062756b714d9aa1151"       ],       expires="Tue, 22-Oct-19 10:00:39 GMT; path=/; domain=.typicode.com; HttpOnly"     },     "x-powered-by": "Express",     "vary": "Origin, X-HTTP-Method-Override, Accept-Encoding",     "access-control-allow-credentials": "true",     "cache-control": "no-cache",     "pragma": "no-cache",     "expires": "-1",     "access-control-expose-headers": "Location",     "location": "http://isomplaceholder.tyicode.com"   } } </pre> <div>ABBRECHEN</div> </div> </div>	<p>Im Fenster <b>Inhalt des Containers</b> bekommt man zum einen die Daten angezeigt, die man an das Fremdsystem übergeben hat...</p>

## Inhalt des Containers

```
Encoding",
  "access-control-allow-credentials": "true",
  "cache-control": "no-cache",
  "pragma": "no-cache",
  "expires": "-1",
  "access-control-expose-headers": "Location",
  "location": "http://jsonplaceholder.typicode.com
/posts/102",
  "x-content-type-options": "nosniff",
  "etag": "W/\"71-3yJKTy9sbQfjwycABXOGC0KYSI0\"",
  "server": "cloudflare",
  "cf-ray": "46db1e40679297ce-FRA",
  "x-cache": "MISS from opspsrv032",
  "x-cache-lookup": "MISS from opspsrv032:3128",
  "via": "1.1 vegur, 1.1 opspsrv032 (squid/3.3.14)",
  "connection": "close"
},
"statusCode": 201
"body": {
  "title": "Mein neues Posting",
  "body": "Dies ist der Inhalt meines Postings.",
  "userId": 1,
  "id": 102
}
}
}
meta: {
  "finished": true
}
```

ABBRECHEN

...zum  
anderen  
das  
Resultat der  
Integration:  
Die Antwort  
des  
fremden  
Servers mit  
der  
vollständige  
n  
Ressource.  
Diese  
umfasst die  
gesendeten  
Eingaben  
sowie  
zusätzliche  
Felder, die  
von dem  
fremden  
Server  
hinzugefügt  
wurden.

Im Beispiel  
besteht die  
vollständige  
Ressource  
aus den  
Werten

- **title**  
(mit  
dem  
Inhalt "  
Mein  
neues  
Posting"  
)
- **body**  
(mit  
dem  
Inhalt "  
Dies  
ist der  
Inhalt  
meines  
Posting  
s.")
- **userId**  
(mit  
dem  
Inhalt 1)
- **id** (mit  
dem  
Inhalt 1  
02)

Während **title**  
, **body** und  
**userId** die  
Werte  
enthält, die  
mitgegeben  
wurden,  
wurde der  
Inhalt von **id**  
vom  
Fremdsyste  
m  
hinzugefügt.



O  
h  
n

weiterem Modellierungsinstrumente Datenflüchtigkeitsdashboards werden in Containern hinterlegt, nicht gespeichert. Weiterführende

e  
In  
fo  
r  
m  
at  
io  
n  
e  
n  
z  
u  
m  
T  
h  
e  
m  
a  
fi  
n  
d  
e  
n  
Si  
e  
a  
uf  
d  
er  
S  
ei  
te  
[Fl  
ü  
c  
ht  
ig  
e  
W  
er  
te  
in  
di  
e  
D  
at  
e  
n  
b  
a  
n  
k  
s  
p  
ei  
c  
h  
ern](#)  
.



Die Beispiel-EPKs Benutzerdaten mit REST und Posting erzeugen mit REST können auf der Seite [Downloads](#) heruntergeladen werden.