

# Was ist REST? Einsatz in der Industrie



Wenn es um die Kopplung von Systemen geht, dann spielt **REST (Representational State Transfer)** nahezu immer eine Schlüsselrolle. Gerade im Zusammenhang mit webfähigen Systemen ist REST für die Systemintegration einer der führenden Standards. Dabei ist REST keineswegs brandneu, im Gegenteil, das Konzept hinter REST ist altbewährt. Das **Verbinden von Systemen per REST** ist aufgrund seiner einfachen Gestaltung simpel und effektiv. Und durch die Zustandslosigkeit ist die einfache Skalierbarkeit gegeben. Im industriellen Bereich ist REST an vielen Stellen im Einsatz. Wir versorgen Sie hier mit dem notwendigen Basis-Wissen rund um REST. Zusätzlich finden Sie auch Informationen zum **praktischen Einsatz von REST**. Ein detailliertes Video zu diesem Thema finden Sie hier in unserem **Tutorial-Stream**.

# 1. Kernidee von REST

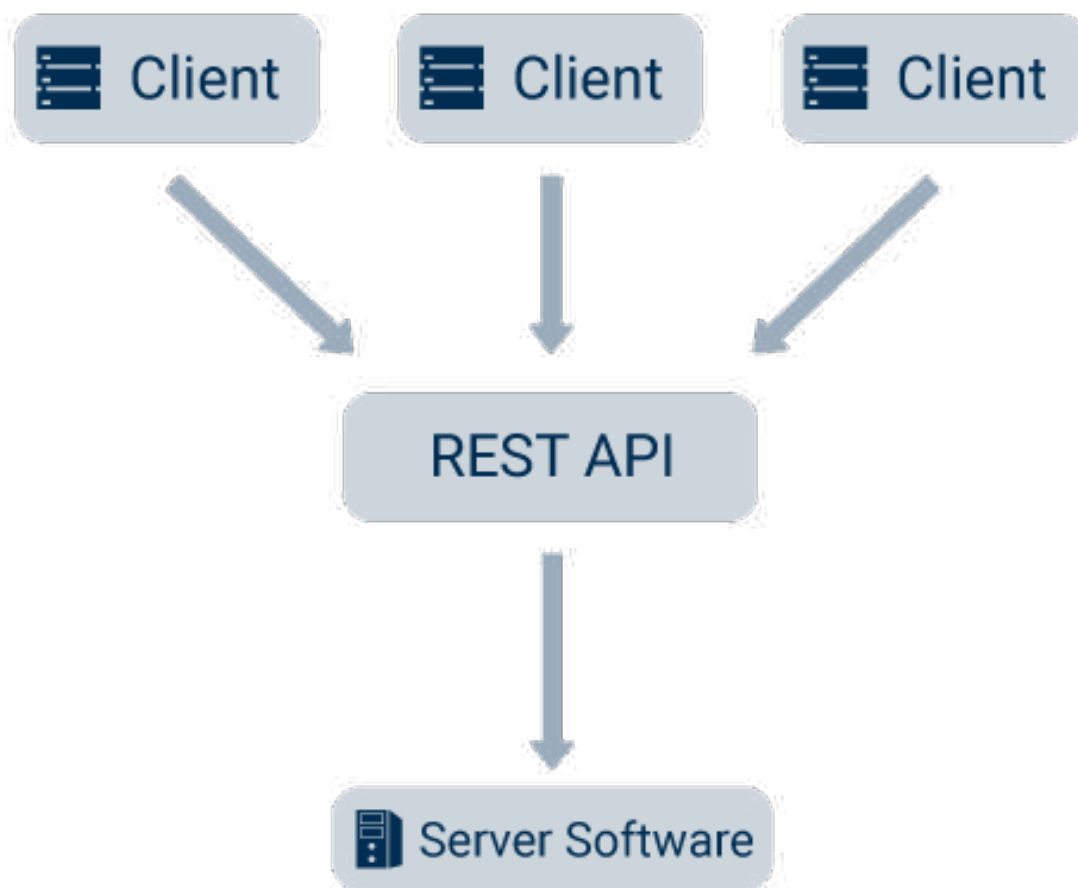


## REST

Die Idee des “Representational State Transfers” (REST) besteht in einer Softwarearchitektur für Datenaustausch zwischen Softwaresystemen (Client-Server). Das besondere gegenüber anderen Architekturen ist die sehr einfache Umsetzung, die Konzentration auf Ressourcen und die Zustandslosigkeit.

Jegliche Kommunikation per REST stellt die Ressource in den Mittelpunkt und nicht die Aktion. Eine Ressource wird immer eindeutig adressiert und anschließend können die Grundaktionen Erstellen, Auslesen, Ändern oder Löschen ausgeführt werden. Zustandslos heißt in diesem Zusammenhang, dass jeweils alle beschreibenden Parameter zwischen Client und Server ausgetauscht werden. Der Vorteil liegt hier im Verzicht auf Sessions. Ohne Sessions könne Client- und Server-Systeme frei skaliert werden, weil jeder REST Aufruf alleinstehend abgeschlossen und verständlich ist, ohne die vorhergehenden oder folgenden.

## 2. REST API / Schnittstelle



### REST API Schnittstelle

Die API (Application Programming Interface) ist die jeweilige Implementierung der REST Architektur eines konkreten Systems. Es wird dann von einer RESTful API gesprochen. In der Schnittstelle werden die Ressourcen definiert, die per REST angesprochen werden können.

Zu den jeweiligen Ressourcen gibt es zugehörige Parameter, welche die Ressource beschreiben und die per REST modifiziert werden.

Für die RESTful API eines Systems stellt der Anbieter im Normalfall eine API Dokumentation bereit, in der alle **Ressourcen** und deren **Parameter** einsehbar sind.

Ein Beispiel für eine gute API Dokumentation ist die REST Schnittstelle unseres Ticketsystems Target Process, die hier eingesehen werden kann: [Target Process REST API](#).

In einer Produktionsumgebung könnte ein REST-fähiges System zum Beispiel folgende Ressourcen bereitstellen:

- Parameter: ID, Auftragsnummer, Menge, Produkt, Termin, ...
- /Produktion/Produktionsauftrag
- Parameter: ID, Auftragsnummer, Menge, Produkt, Termin, ...
- Parameter: ID, Materialnummer, Menge, Charge, ...
- /Produktion/Produktionsauftrag/{id}/Komponente
- Parameter: ID, Materialnummer, Menge, Charge, ...
- Parameter: ID, Status, aktueller Auftrag, nächste Wartung, Betriebsstunden, ...
- /Produktion/Anlage
- Parameter: ID, Status, aktueller Auftrag, nächste Wartung, Betriebsstunden, ...

### 3. REST und HTTP

Die Architektur-Idee des Representational State Transfers schreibt keine Technologie für die Umsetzung vor. In der Praxis wird REST allerdings immer mit dem **HTTP-Protokoll** realisiert. Die Ressourcen werden dabei mit einem eindeutigen **URI** (Uniform Ressource Identifier) adressiert, wie man es aus dem Webbrowser kennt. Die Parameter der Ressource werden als Query angehängt (auch als URL-Parameter bekannt).

Die Aktionen auf die Ressourcen werden in HTTP mit den Standard Aktionen GET, POST, PUT und DELETE umgesetzt. Dabei gilt folgender Standard:

- GET ruft eine oder mehrere Ressourcen ab
- POST erzeugt eine neue Instanz einer Ressource
- PUT schreibt Daten auf eine Ressource und ändert sie damit
- DELETE löscht eine Instanz einer Ressource

Ein REST Aufruf zum Abruf eines bestimmten Produktionsauftrags von einem REST-fähigen ERP System könnte über eine bestehende http-Verbindung dann z.B. so aussehen:

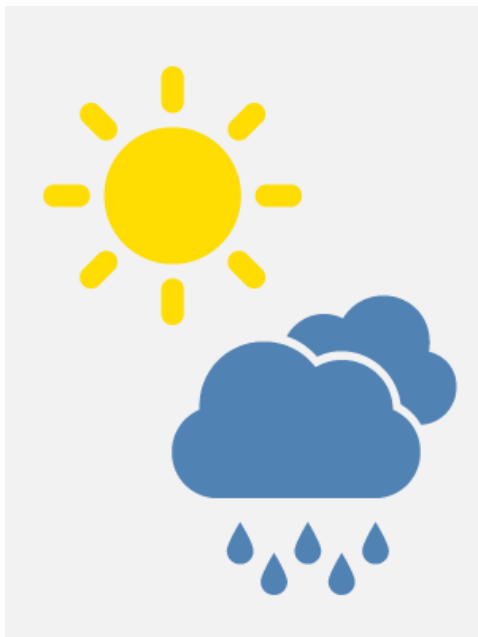
```
GET http://ERPSYSTEM/Produktionsauftrag/4711
Accept: application/json
```

Die zweite Zeile gibt dem ERP-System dabei vor, in welchem Format die Antwort erfolgen soll.

Eine mögliche Antwort im JSON-Format wäre:

```
[  
{  
  „id“: „4711“,  
  „Artikel“: „Produkt A“,  
  „Menge“: „10500“,  
  „Freigabe“: false,  
  „Lieferdatum“: „21.7.2020“  
}  
]
```

## 4. Beispiel



REST Open Weather mit dem OPC Router

Ein reales Beispiel für einen REST-Aufruf lässt sich zum Beispiel auf der Seite des Wetter-Vorhersage-Dienstes OpenWeatherMap ausprobieren.

Wird der Link mit dem Webbrowser aufgerufen (per Click), wird eine **GET** Anfrage an den Dienst gesendet. Als Ergebnis antwortet der Dienst mit der Wetter-Vorhersage für den Ort London.

Der **Endpunkt** ist in diesem Fall: `api.openweathermap.org/data/2.5/forecast`

Die Ressource ist der "forecast" mit den zugehörigen Parametern "q" für den Ort und "mode" für das Format. Ein ausführliches Beispiel für den Aufruf dieser API finden Sie in unserem [Anleitungsartikel für den Abruf von Wetterdaten von OpenWathermap mit dem OPC Router](#).

## 5. Datenformate

Das Datenformat der Antwort eines REST-Endpunkts ist nicht vorgeschrieben und kann daher im Prinzip beliebig sein. In der Praxis richtet sich das Format nach der Funktion des Endpunkts. In den meisten Fällen werden die Daten als [JSON](#)- oder als [XML](#)-Dokument geliefert. Wird eine normale Webseite per REST-Aufruf abgefragt, erfolgt die Antwort in HTML.

Die Struktur von XML/JSON Antworten muss vom Betreiber der REST Schnittstelle dokumentiert werden, da es hier keine Möglichkeit der Selbstbeschreibung gibt.

## Lassen Sie Ihre Systeme per REST miteinander sprechen

Testen Sie jetzt den OPC Router mit dem REST Plug-in kostenfrei und unverbindlich.

## 6. Service / Webservice

Ein Webservice bzw. Service bezeichnet in Bezug auf Software einen Dienst, der den Datenaustausch und die Zusammenarbeit von verschiedenen Softwaresystemen erlaubt. Der Webservice ermöglicht dies dabei im Web durch Standard-Webtechnologien. Neben vielen weiteren ist REST eine konkrete Ausprägung einer Webservice-Umsetzung. Andere sind zum Beispiel [SOAP](#) oder RPC (Remote Procedure Call).

## 7. Sensoren mit REST

Viele Feldgeräte mit eingebauten Kleinrechnern bieten Ihre Daten auch über REST an. So können diese Feldebene-Geräte einfach in andere Anwendungen integriert werden.

Beispiel hierfür sind I/O-Baugruppen (z.B. von [WuT](#)) oder Energiemessgeräte (z.B. von Phoenix Contact).

Für Steuerungen (SPS) und andere IT-fernere Datenquellen bietet der [Kepware OPC Server](#) auch eine Möglichkeit Daten per REST Server anzubieten.

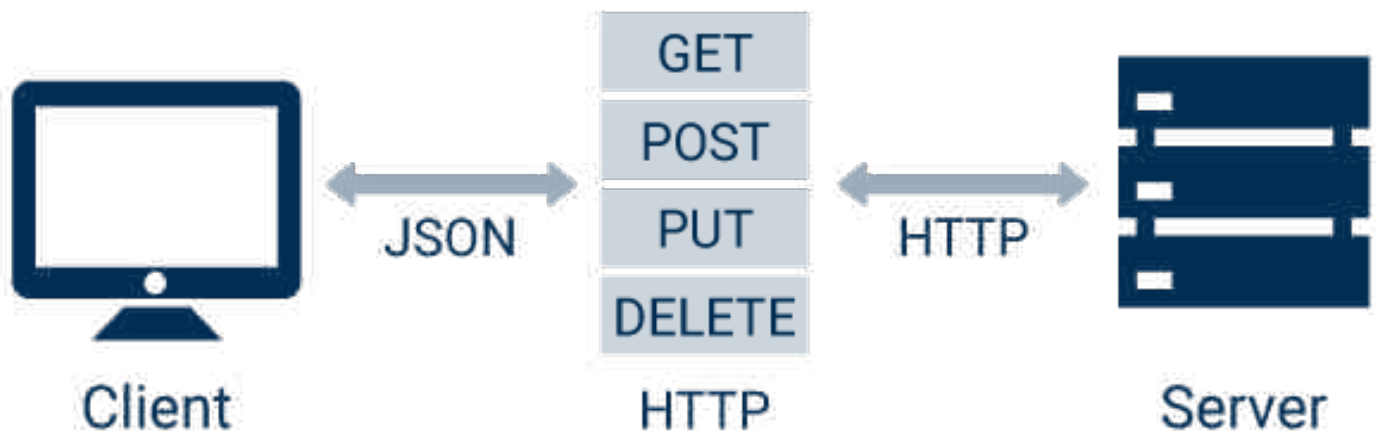
## 8. Softwaresysteme mit REST

Eine große Verbreitung der REST-Schnittstelle findet sich bei Softwaresystemen (z.B. ERP, MES, PLS) und vor allem im Web bei Cloud-basierten Systemen. Bei einer gut ausgestatteten API kann nahezu der gesamte Funktionsumfang des Softwaresystems über die Schnittstelle angesprochen werden oder es können alternativ benutzerspezifische REST Endpunkte geschaffen werden.

Beispiele hierfür sind neben vielen anderen [SAP mit REST über PI](#), Microsoft mit der [REST API für Dynamics 365](#) oder die [Siemens Mindsphere REST API](#).

Letztendlich lohnt aber bei jedem einzusetzten Softwaresystem die Frage an den Hersteller nach einer REST API.

## 9. Der Endpunkt



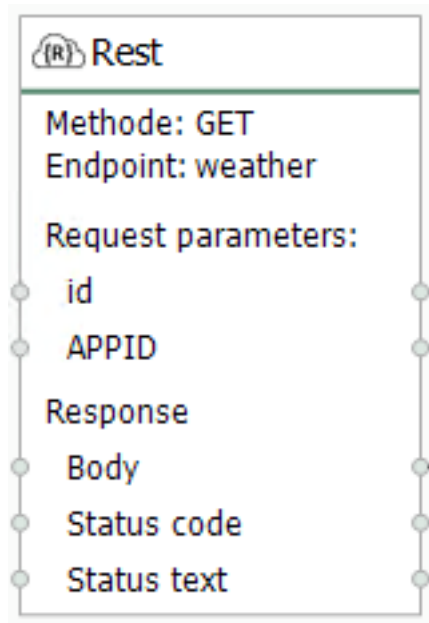
REST Endpunkt

Endpunkte sind die vollen URIs, aus denen sich die jeweilige REST API zusammensetzt. Die Summe aller Endpunkte ist die API und der einzelne Endpunkt spricht genau eine Ressource an. Für das Beispiel des Produktionsauftrags könnte ein Endpunkt zur Abfrage einer Komponentenmenge wie folgt aussehen:

`/Produktionsauftrag/{id}/Komponente/{id}/Menge`

Wobei in diesem Endpunkt die Nummern für Auftrag und Komponente als ID übergeben werden.

## 10. OPC Router und REST



#### OPC Router und REST

Der OPC Router dient als Client beim Aufruf einer REST API (s. REST Plug-in ). Im OPC Router wird der Endpunkt für den Aufruf konfiguriert und dann in einer Verbindung aufgerufen. Im Transferobjekt sind dann die Methode, der Endpunkt und die Parameter ersichtlich.

Ein detailliertes Beispiel ist hier zu finden: [Anleitungsartikel für den Abruf von Wetterdaten von OpenWathermap mit dem OPC Router](#)

Neben dem Aufruf von REST Endpunkten kann der OPC Router auch als Server dienen und REST Endpunkte definieren. Dafür stellt der OPC Router sogenannte REST Trigger bereit. Auf diese Weise kann mit den OPC Router eine eigene REST API aufgebaut werden.

## 11. Postman – Test API Client





Zum schnellen und einfachen Testen von REST Schnittstellen gibt es verschiedene Tools. Eines der bekanntesten ist der Postman. Ein sehr mächtiges Programm zum Aufrufen und Erstellen von APIs. Postman kann hier heruntergeladen werden: [Postman REST API Client](#).

## 12. Swagger – Framework für API Beschreibung



Swagger

Das [Swagger Framework](#) dient zur Erstellung, Nutzung und Dokumentation von Webservices. Für REST Clients ist dies besonders hilfreich, da eine per Swagger dokumentierte REST API eine standardisierte Dokumentation bereitstellt, so dass im Client ein Browsing der Schnittstelle möglich ist. Diese Browsing-Funktionalität ist nicht in der Kernidee von REST enthalten und wird hiermit ergänzt

## 13. Geschichte von REST

Webservices waren von Beginn an ein wichtiges Thema in verteilten Softwaresystemen. Viele Technologien wurden schnell sehr mächtig und stellten viele Funktionen bereit. Für kleinere Anwendungen waren diese dann oftmals bereits zu komplex und zu rechenintensiv.

Im Jahr 2000 nahm das der amerikanische Informatiker [Roy Fielding](#) zum Anlass eine neue einfache Architektur zu entwerfen. Ihm wurde dafür der Doktor-Titel verliehen. Seitdem hat REST sich stark verbreitet und dient als einer der wichtigsten Systemkopplungs-Techniken im Internet.

## 14. Zukunft von REST – GraphQL

Das Architekturmodell von REST ist fix und wird so wie es ist betrieben. Die fehlenden Strukturen in REST Abfragen haben dazu geführt, dass es neue Entwicklungen gibt, welche die Idee der REST Architektur aufnehmen und weiterentwickeln. Facebook hat 2012 die GraphQL Architektur entwickelt und 2015 veröffentlicht. Genau wie REST ist GraphQL eine zustandslose Abfragesprache. Neben einigen anderen Verbesserungen werden in GraphQL Schemas definiert. Damit ist festgelegt, wie ein Endpunkt antwortet und wie die Struktur der Daten in der Antwort sind. GraphQL verbessert die Architektur damit bei einem der größten Schwachpunkte von REST.

## Jetzt zum Newsletter anmelden

Bleiben Sie informiert und melden Sie sich zu unserem OPC Router Newsletter an.

## 15. Sicherheit in REST

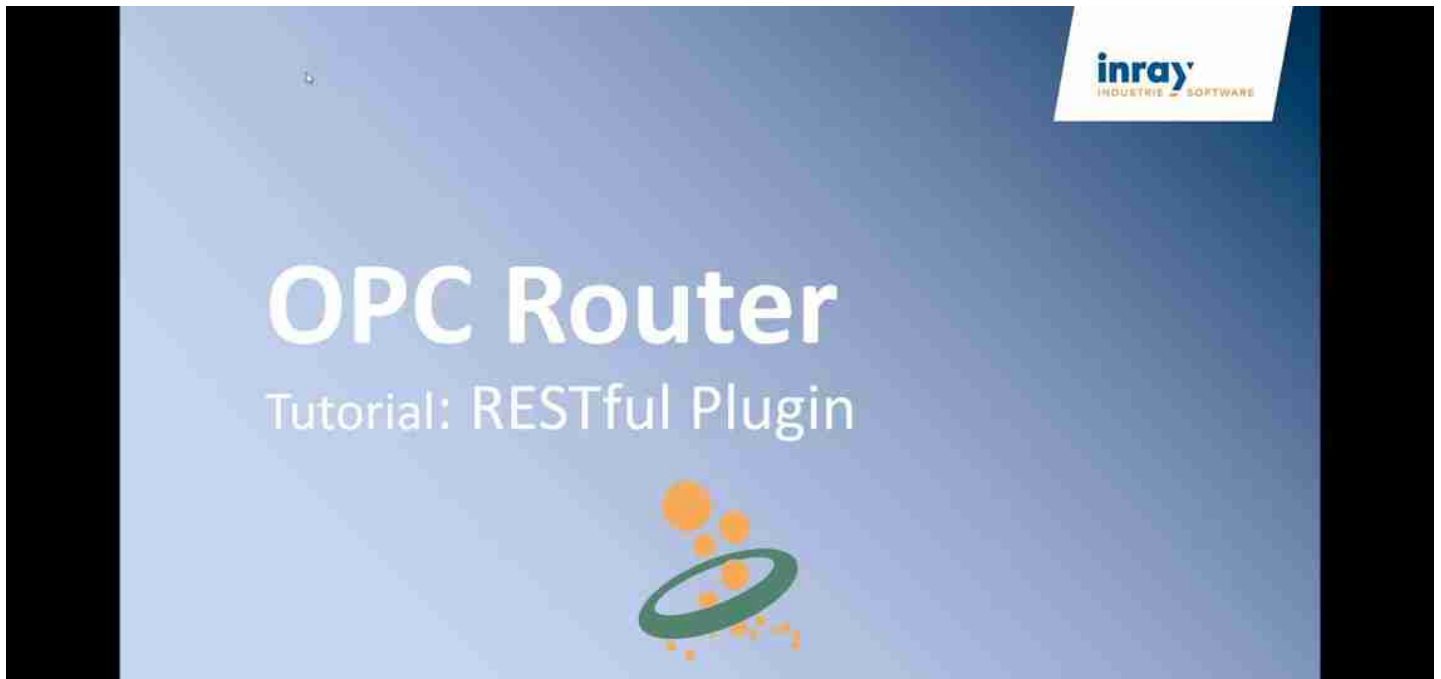
Da für den Aufruf von REST Endpunkten HTTP verwendet wird, können auch die im Standard verfügbaren Authentifizierungen genutzt werden, wie z.B.: HttpBasic, Jwt, Ntml, OAuth1, OAuth2.

Zusätzlich wird statt http natürlich https genutzt, um eine abhörsichere Verbindung zu nutzen.

Neben den Standard Authentifizierungsmöglichkeiten wird oft auch ein sogenannter AppKey ausgetauscht. Dieser Key ist ein für den Client erstellter, geheimer Code, der bei jedem Aufruf mit übergeben wird, um so die Berechtigung für den Aufruf zu bekommen. Auch der sogenannte Bearer Token wird vom OPC Router unterstützt. REST ist durch die Nutzung weit verbreiteter Methoden als sicher anzusehen.

## 16. Video: Tutorial REST Plug-in

In diesem Video erklären wir Ihnen, wie Sie den OPC Router in Ihre REST-Kommunikation einbinden. Im ersten Beispiel fragen wir Wetterdaten bei OpenWeather ab. Im zweiten Beispiel zeigen wir Ihnen, wie Sie am OPC Router eine REST-ful Schnittstelle zum Auslesen von Auftragsdaten aus einer Datenbank konfigurieren und testen können.



## Einfache REST-Kommunikation in der Praxis

REST ist einer der führenden Standards wenn es um die Kopplung webfähiger Systeme geht. Über die REST Schnittstelle (REST API) können Client und Server ganz einfach und effektiv miteinander kommunizieren. Für die Systemintegration mit REST ist kein umfangreicher Code notwendig, meist genügt eine einfache URL. Mit den vier einfachen http-Aktionen GET, POST, PUT und DELETE kann REST nahezu alle Aufgaben ausführen.

Für eine [Systemintegration per REST](#) in der Praxis wird zunächst eine Software benötigt, die REST Webservices ansprechen und abfragen kann.

Software-Produkte wie z.B. der [OPC Router](#) bieten hierfür sogar eine grafische Benutzeroberfläche, in der Verbindungen mit Maschinen, Steuerungen, Datenbanken, Cloud-Systemen und anderen Systembestandteilen ganz einfach per Drag & Drop erstellt werden können. Die Software sollte zudem einen individuellen und anwendungsbezogenen Datenaustausch ermöglichen. Dies funktioniert mit sogenannten Plug-ins beispielsweise für [SAP-Systeme](#) oder eine [SQL-Datenbank](#) genauso wie für Cloud-Systeme oder IoT-Plattformen.

Besonders in der Industrie kann REST somit sehr vielfältig eingesetzt werden und ermöglicht eine effizientere Kommunikation.

Das könnte Sie auch interessieren



Was ist REST?

## Details zum REST Plug-in

Für die Systemintegration ist das Architekturmodell von REST ein führender Standard. Er basiert auf reiner Webtechnologie und einer zustandslosen Logik zum Abfragen und Editieren von Daten in Systemen. So ist eine einfache Skalierbarkeit möglich, weshalb REST in der Industrie vielfältig im Einsatz ist. Auf unserer Detail-Seite zur [REST Schnittstelle](#) erfahren Sie alle Details zur Anbindung per REST sowie praktische Anwendungs-Beispiele.



Ewon HMS Flexy 205 per REST über Talk2M-Cloud ansprechen

## Dezentrale per Cloud verfügbare Industrieanlagen per REST anbinden

Dezentrale, per Cloud verfügbare Industrieanlagen können durch REST perfekt in Systemstrukturen integriert werden. Für Wartungseingriffe oder bei Störungen muss dann niemand mehr an der Anlage präsent sein. Technisch umsetzbar ist diese Anbindung mit dem Ewon Flexy 205 Fernwartungsrouter, der zusätzlich zur reinen Remote-Vernetzung auch die Anbindung an die Talk2M-Cloud ermöglicht. Für weitere Systeme bietet die Cloud eine modern REST Schnittstelle an. Wie genau die Einrichtung mit dem OPC Router funktioniert, erklären wir in unserer Schritt für Schritt-Anleitung „[Talk2M-Anbindung](#)„.



SMS- und WhatsApp-Versand mit Twilio

## Mobile Benachrichtigungen für mobile Nutzer per SMS und als WhatsApp versenden

In der Industrie 4.0 ist der Zugriff auf Informationen essenziell, um rechtzeitig die richtigen Entscheidungen treffen zu können. Am besten ist es, wenn die Prozesse sich aktiv melden – und das auch noch mobil. Bekannte Benachrichtigungsdienste wie WhatsApp oder die klassische SMS ermöglichen das in Kombination mit dem OPC Router, dem REST Plug-in und dem Twilio Dienst. Unsere Schritt für Schritt-Anleitung „[Benachrichtigungen für mobile Nutzer per SMS und als WhatsApp-Nachricht versenden](#)“ zeigt Ihnen, wie Sie Ihre ganz individuellen Benachrichtigungen versenden können.



ThingWorx step-by-step via REST Plug-in verbinden

## IoT-Plattform ThingWorx via REST Plug-in verbinden

Die IoT-Plattform ThingWorx von ptc stellt virtuelle Abbildung von Geräten, Maschinen und Sensoren mit dem Konzept des „digitalen Zwillings“ für Visualisierungen, Analytics oder auch Augmented Reality zur Verfügung. Die Abbildungen dieser digitalen Zwillinge (Things) können per REST angebunden und Daten so übergeben oder abgerufen werden. Unsere Anleitung „[ThingWorx via REST Plug-in verbinden](#)“ zeigt, wie Sie Eigenschaften und Methoden der digitalen Zwillinge aus der IoT Plattform ThingWorx mit Hilfe des OPC Router REST Plug-ins anbinden.

# Lassen Sie Ihre Systeme per REST miteinander sprechen

Testen Sie jetzt den OPC Router mit dem REST Plug-in kostenfrei und unverbindlich oder melden Sie sich zu unserem Newsletter an und bleiben Sie so stets über Änderungen und Aktuelles informiert.

Teilen Sie unseren Beitrag!

[Benjamin Krause](#)2022-07-18T11:42:01+02:00