

Background information:

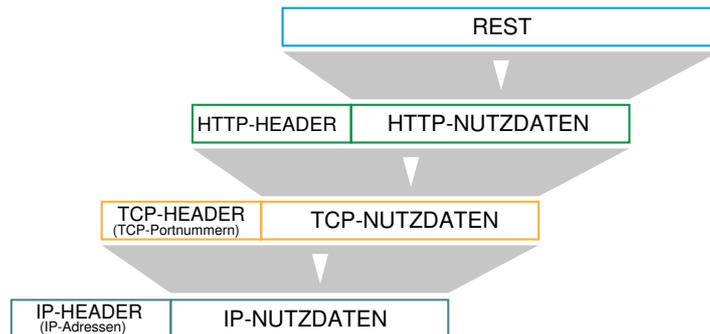
REST - REpresentational State Transfer

Industrial communication based on standardized HTTP requests

REST does not describe an actual protocol structure. Rather REST prescribes properties which should be met for data communication in industrial environments.

REST - Transmission on the network level

It should be said in advance that REST uses HTTP or HTTPS as the higher level protocol.



The advantage of HTTP(S) as a communication basis is that most networks, even those protected by firewalls etc., are fully usable for HTTP(S). In addition, HTTP(S) has the characteristics that REST requires.

As is common in the case of HTTP and HTTPS, TCP ports 80 and 443 are used.

REST - Properties and basic elements

Client/Server Model

REST is based on the Client/Server model, where an action is always initiated by the client. The server provides resources (data, content, functions). The client sends a request to access selected resources. The server replies with the requested data or confirms the desired action.

Stateless condition

In many Client/Server applications when the server opens a connection a particular status (authorization, special task, specific purpose, ...) is assigned which remains in effect for the duration of the connection. REST at first treats everything incoming from the server the same, and only the content of the data transfer determines what proceeds on the server side and how the request is classified. The status of an application thus lies in the hands of the client.

Layered system

REST provides for clear separation of responsibilities. This separation applies in particular to how communication is handled and the processing of the transported contents.

This makes it possible to use proxies, gateways or similar intermediate stations along the transmission path.

In addition, HTTP or encrypted transmission via HTTPS can be selected depending on the security requirements. Which is used has no effect on the functionality of the REST contents.

Addressability of resources

All resources available on a server are accessible using a unique address (URI - Uniform Resource Identifier). The URI is structured like the URL which is used for addressing in the browser:

Protokoll://<Host>:<Port>/<Pfad>/<Ressource>?<Parameter>&<Parameter>

Request methods

Standardized HTTP requests can be used:

- **GET**
requests resources from the server and is then used read-only
- **POST**

posts new resources or changes existing ones

- **PUT**
changes existing resources
- **DELETE**
deletes existing resources
- **HEAD**
requests only the HTTP head, for example to check the availability of the requested resource
- **OPTIONS**
requests information about the communication options

The data requested are referred to as representations of one or more resources. Hence the term REST for REpresentational State Transfer.

Representations are basically nothing more than an image of parts of the process data. Transmission can be in any agreed upon form. Common are JSON, XML or raw text. SVG, MP3 or other formats can however also be used depending on the application. Representations may also contain hyperlinks to additional resources.

The client sends:

```
http://192.168.1.19/rest/json/iostate/
```

And the Web Thermo-Hygrobarometer sends in reply:

```
{
  "iostate": {
    "sensor": [
      {
        "name": "Temperatur",
        "number": 0,
        "unit": "°C",
        "value": 25.9
      },
      {
        "name": "rel. Feuchte",
        "number": 1,
        "unit": "%",
        "value": 43.2
      },
      {
        "name": "Luftdruck",
        "number": 2,
        "unit": "hPa",
        "value": 994.7
      }
    ]
  }
}
```

Code on Demand

For websites it is usual today to also include JavaScript for example. REST also allows you to later include source code, program sections or function blocks. This enables the client function to be expanded or modified during runtime.

Cache

REST provides for responding to repeated client generated requests from a cache in order to reduce the data load on the transmission channels. The server uses corresponding entries in the HTTP head to determine whether the cache is allowed to be used for the requested representation or not.

Advantages of REST

Simple implementation, since HTTP mechanisms are used for the most part.

Disadvantages of REST

The request/reply methodology means that only polling - i.e. targeted calling of data - is possible, but not event-driven communication. The relevant data must be continuously polled in order to detect changes.

Web-IO - Automation using standard protocols
Integrating signals with Modbus-TCP, OPC UA/DA, REST or MQTT

Data formats and protocols
From traditional data exchange to IoT

Modbus-TCP
Open standard for industrial communication

MQTT - Message Query Telemetry Protocol
Data exchange without direct connection

OPC UA - OPC Unified Architecture
OPC support out of the box

OPC DA - OPC Data Access
The process data interpreter

Products for industrial applications with standard protocols



Web-IO 4.0 Digital
2xIn, 2xOut

Power via PoE also when needed



Web-IO 4.0 Digital
12xIn, 12xOut

12x outputs (6-30V),
12x inputs (8-30V)



Other Web-IOs

All W&T Web-IO Digital 24V

W&T

www.wut.de

We are available to you in person:

Wiesemann & Theis GmbH
Porschestr. 12
42279 Wuppertal
Phone: +49 202/2680-110 (Mon.-Fri. 8 a.m. to 5 p.m.)
Fax: +49 202/2680-265
info@wut.de

© Wiesemann & Theis GmbH, subject to mistakes and changes: Since we can make mistakes, none of our statements should be applied without verification. Please let us know of any errors or misunderstandings you find so that we can become aware of and eliminate them.

[Data Privacy](#)