

# REST API

REST (Representational State Transfer) is an architectural style for designing networked applications. A REST API is an application programming interface that follows REST principles and typically uses HTTP for communication.

REST was introduced by Roy Fielding in 2000 in his doctoral dissertation. It is not a strict protocol like JSON-RPC, but rather a set of architectural constraints.

REST is resource-oriented, which means the system is organized around resources instead of procedures. A resource can represent any entity, such as a user, product, order, or document. Each resource is identified by a unique URL.

For example:

```
GET /users/15
GET /products/42
```

In REST, the client does not call functions explicitly. Instead, it performs operations on resources using standard HTTP methods.

## Core REST Principles

### Resource Identification

Every resource must have a unique identifier, typically a URL. The URL represents the resource, not the action.

Bad example (RPC-style thinking):

```
GET /getUserById?id=10
```

### HTTP Methods (Verbs)

REST relies on standard HTTP methods to define operations:

Method	Meaning	Example
GET	Retrieve resource	GET /users/1
POST	Create new resource	POST /users
PUT	Replace resource	PUT /users/1
PATCH	Partial update	PATCH /users/1
DELETE	Remove resource	DELETE /users/1

The method defines the action, not the URL.

## Statelessness

Like JSON-RPC, REST is stateless. Each request must contain all necessary information. The server does not store client session state between requests (unless explicitly implemented via tokens or cookies).

This improves scalability and simplifies distributed deployments.

From:

<https://edu.iit.uni-miskolc.hu/> - **Institute of Information Science - University of Miskolc**

Permanent link:

[https://edu.iit.uni-miskolc.hu/tanszek:oktatas:iss\\_t:rest\\_api?rev=1772385416](https://edu.iit.uni-miskolc.hu/tanszek:oktatas:iss_t:rest_api?rev=1772385416)

Last update: **2026/03/01 17:16**

