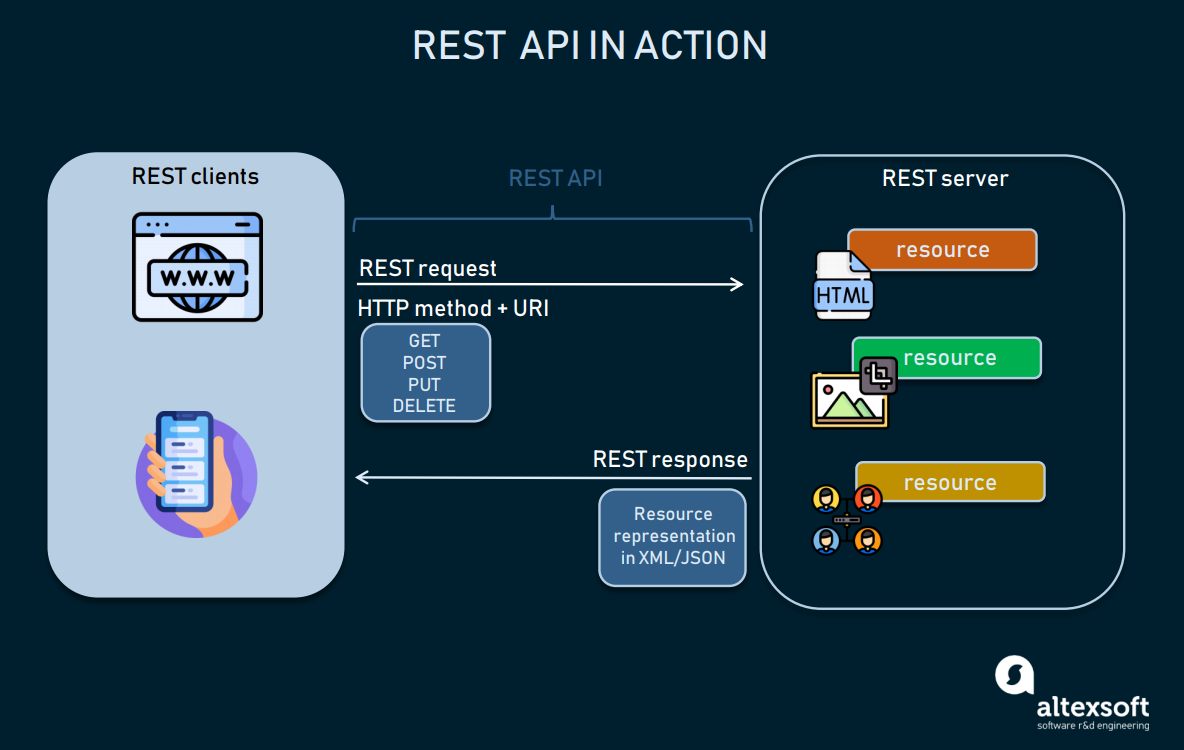
REST API - EXPLAINED

# WHAT IS REST API

* REST is short for **Representational State Transfer,**an architectural style for building web services that interact via an HTTP protocol. Its principles were formulated in 2000 by computer scientist Roy Fielding and gained popularity as a scalable and flexible alternative to older methods of machine-to-machine communication. It still remains the gold standard for public APIs.

# REST API IN ACTION



*A REST client can interact with each resource by sending an HTTP request.*

* The key elements of the REST API paradigm are
* a client or software that runs on a user’s computer or smartphone and initiates communication;
* a server that offers an API as a means of access to its data or features; and
* a resource, which is any piece of content that the server can provide to the client (for example, a video or a text file).
* To get access to a resource, the client sends an HTTP request. In return, the server generates an HTTP response with encoded data on the resource. Both types of REST messages are self-descriptive, meaning they contain information on how to interpret and process them.

# REST API – REQUEST structure

Any REST request includes four essential parts: an HTTP method, an endpoint, headers, and a body.

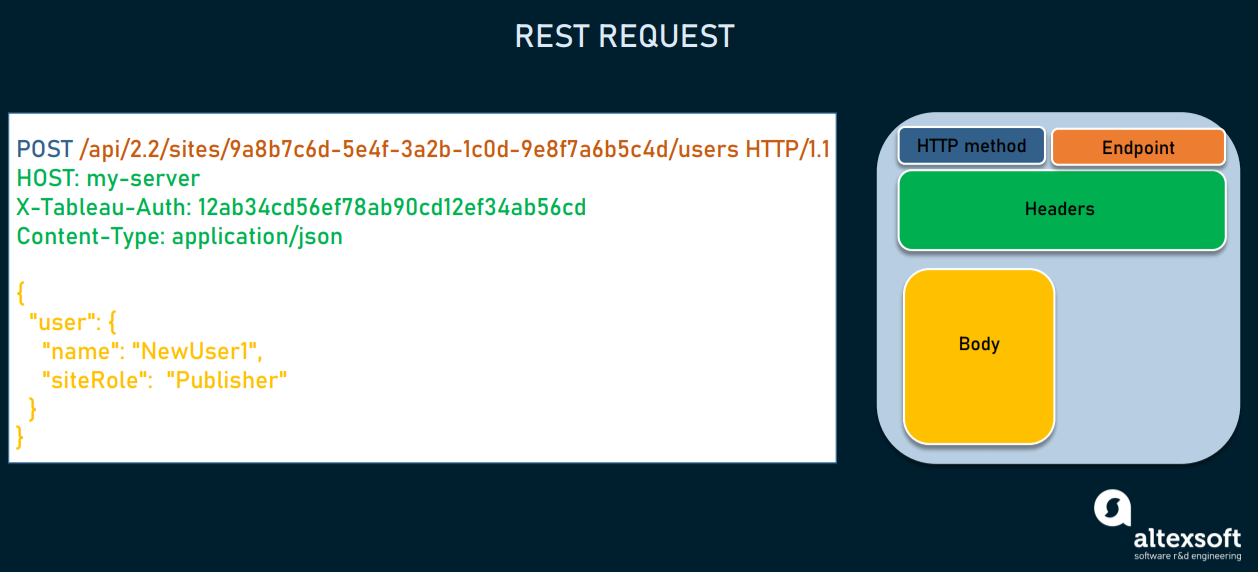
An HTTP method describes what is to be done with a resource. There are four basic methods also named CRUD operations:

* POST to Create a resource,
* GET to Retrieve a resource,
* PUT to Update a resource, and
* DELETE to Delete a resource.

An endpoint contains a Uniform Resource Identifier (URI) indicating where and how to find the resource on the Internet. The most common type of URI is a Unique Resource Location (URL), serving as a complete web address.

Headers store information relevant to both the client and server. Mainly, headers provide authentication data — such as an API key, the name or IP address of the computer where the server is installed, and the information about the response format.

A body is used to convey additional information to the server. For instance, it may be a piece of data you want to add or replace.

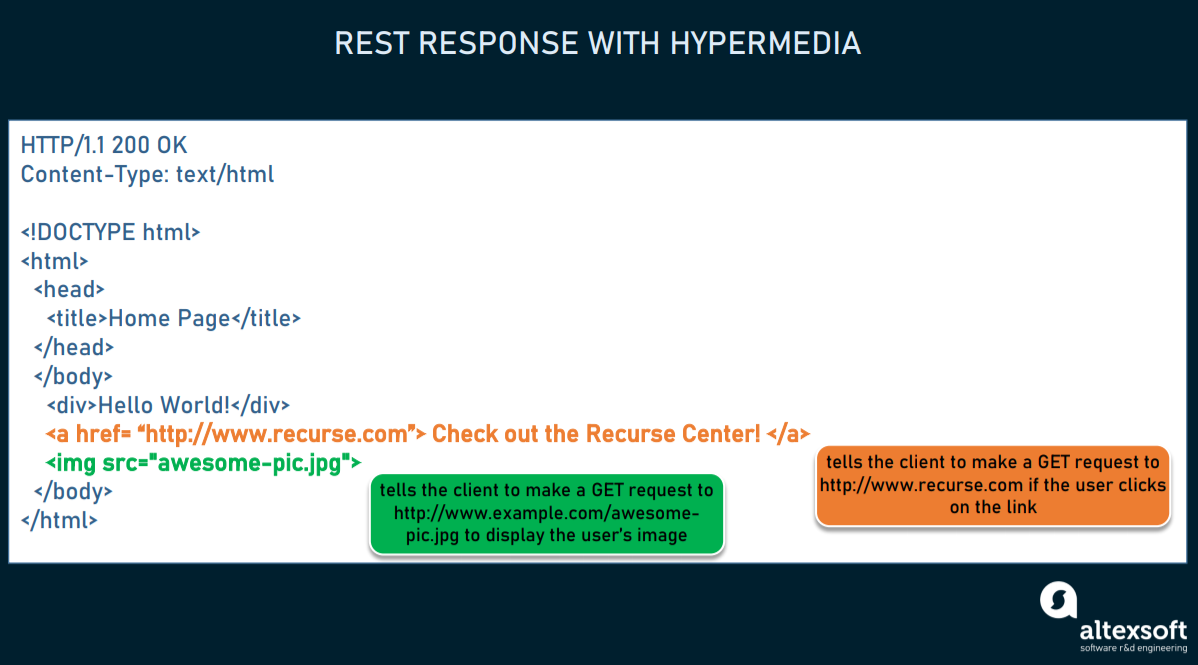


*REST request for creating a new user where the response will return the ID of the created resource.*

# REST API - RESPONSE structure

In response, the server sends not the sought-for resource itself, but its **representation**— a machine-readable description of its current state. The same resource can be represented in different formats, but the most popular ones are XML and JSON.

Whenever relevant, a server includes in the response hyperlinks or **hypermedia** that links to other related resources. This way, the server gives instructions on what the client can do next and what further requests it can make.

*An example of a self-descriptive server response with hypermedia.*