REST determines how the **API** looks like. It stands for “Representational State Transfer”. It is a set of rules that developers follow when they create their **API**. One of these rules states that you should be able to get a piece of data (called a resource) when you link to a specific URL..

Representational state transfer is a software architectural style that defines a set of constraints to be used for creating Web services. Web services that conform to the REST architectural style, called RESTful Web services, provide interoperability between computer systems on the Internet.

What is SOAP and REST API?

SOAP and REST are two **API** styles that approach the question of data transmission from a different point of view. **SOAP** is a standardized protocol that sends messages using other protocols such as HTTP and SMTP. ... It allows different messaging formats, such as HTML, JSON, XML, and plain text, while **SOAP** only allows XML.

What is REST API example?

A **REST API** defines a set of functions which developers can perform requests and receive responses via HTTP protocol such as GET and POST. ... The World Wide Web (WWW) is an **example** of a distributed system that uses **REST** protocol architecture to provide a hypermedia driven interface for websites.

Why is REST API used?

REST or RESTful **APIs** were designed to take advantage of existing protocols. While **REST** - or Representational State Transfer - can be **used** over nearly any protocol, when **used** for web **APIs** it typically takes advantage of HTTP. ... One of the key advantages of **REST APIs** is that they provide a great deal of flexibility.

How does REST API work?

It is a set of rules that allow programs to talk to each other. The developer creates the **API** on the server and allows the client to talk to it. **REST** determines how the **API** looks like. It stands for “Representational State Transfer”.

What are REST principles?

REST is stateless. That means the communication between the client and the server always contains all the information needed to perform the request. ... The client, the server and any intermediary components can all cache resources in order to improve performance. **REST** provides a uniform interface between components.