TREXIN INSIGHT PAPER

September 22, 2020

SIMPLIFY WEB SERVICES WITH RESTFUL API'S

Leveraging the principles of REST to develop scalable and resilient services for the Cloud.

REST (Representational State Transfer) is an alternative to SOAP (Simple Object Access Protocol) that provides updated standards and design to simplify communication between Clients and Web Services.

Representational State Transfer (REST) is a stateless architecture methodology for web services, presented by Roy Fielding in 2000. REST enables clients and services to communicate requests and responses in a standard format using REST statements. When a client sends a request to a REST service, it instructs the service on what the client wants and where to locate it. The client accomplishes this by using HTTP request methods to specify how it wants to interact with the resource. These interactions can be creating, obtaining, or modifying resources like data, images, videos, or audio clips. Leveraging RESTful architecture enables companies to build services with improved scalability and reliability.

Driving Value from Implementing REST Principles

- Increases sustainability of an IT ecosystem and reduces development time as resource manipulation does not have to be coded into each service
  - REST design creates a single shared service by decoupling the logic of resource manipulation from the client
- Developers can quickly triage defects or issues in client/service interactions
  - Because HTTP protocol is used, this enables the service to send HTTP codes in responses that explain what type of error occurred
- Increases reliability in IT systems and minimizes data loss from errors in client service interactions
  - Failures in a request or a response is constrained between the service and the client in that one instance for whatever action was occurring
- Provides flexibility for services to offer multiple data formats
  - REST provides flexibility in how data is structured compared to SOAP, which focuses on structuring data using XML and schema
- Implementing REST principles is easy for organizations as it does not require a significant amount of knowledge
  - REST uses standardized data formats and HTTP protocol is widely used and accepted in the industry
- Microservices and services deployed on Kubernetes, benefit from REST design
  - REST is independent, stateless, and scalable in nature

Understanding How REST is Applied

RESTful architecture is stateless, meaning the service does not need to know anything about the client or vice versa and the service does not store any data. REST architectures enable clients and services to implement independent code changes because it breaks out the logic of calling, manipulating, and deleting data into its own standalone service. Because REST architecture creates a standalone service, this enables multiple clients to leverage the same REST endpoints to perform the same action and get the same response through the REST Application Programming Interface (API).

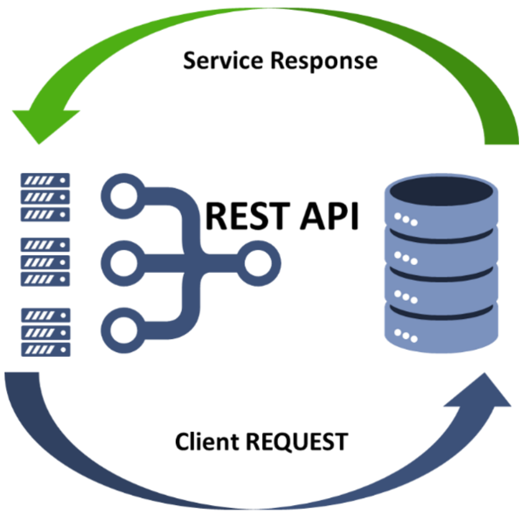
What is a REST API?

A web service that conforms to the REST API specification, is considered a RESTful API. An API contains the information and procedures needed to integrate applications. The REST API denotes what types of request the client can make to the service and the data formats the service is able to understand. If a change made to the service breaks the established convention, then the REST API specification needs to be updated and absorbed by all clients sending REST requests to that endpoint.

The REST Request and Response Process

REST architecture leverages HTTP protocol to create and communicate REST requests. When a client sends a request in a RESTful Architecture it sends one of four HTTP commands. These commands allow the client to create, retrieve, modify, or delete a resource. Resource is defined as a piece of information or data. See below for examples of the four commands:

- GET - Retrieves a resource
  Example: Retrieve bank account information
- POST - Creates a new resource
  Example: Create a new bank account
- PUT - Updates an existing resource
  Example: Change the address on someone's bank account
- DELETE – Deletes a resource
  Example: Delete an address from a bank account



Because REST uses HTTP protocol, the service can send the response with different HTTP codes back to the client denoting if the request was successful, not found, and many more. For example, a client sent a request to create an online account, but the service found a match and returned an error saying the account exists.

REST Simplifies Interactions between Client and Web Service

Because standard HTTP protocol is used, REST design is easier for IT organizations to implement. Development time is reduced since it creates a shared service by decoupling the logic of resource manipulation from the client. Microservices
and Cloud-based technologies like Kubernetes benefit from REST design due to it being stateless and scalable. By simplifying client and webservice interactions, REST increases the sustainability and nimbleness of IT systems.

Resources:
https://restfulapi.net
https://restfulapi.net/soap-vs-rest-apis/
https://www.codecademy.com/articles/what-is-rest
https://www.redhat.com/en/topics/api/what-are-application-programming-interfaces

This TIP was written by Trexin Associate, Benjamin Stensrud. Ben welcomes comments and discussion on this topic and can be reached at benjamin.stensrud@trexin.com.