Webservices

By
Jitendra Singh Tomar || Jeetu

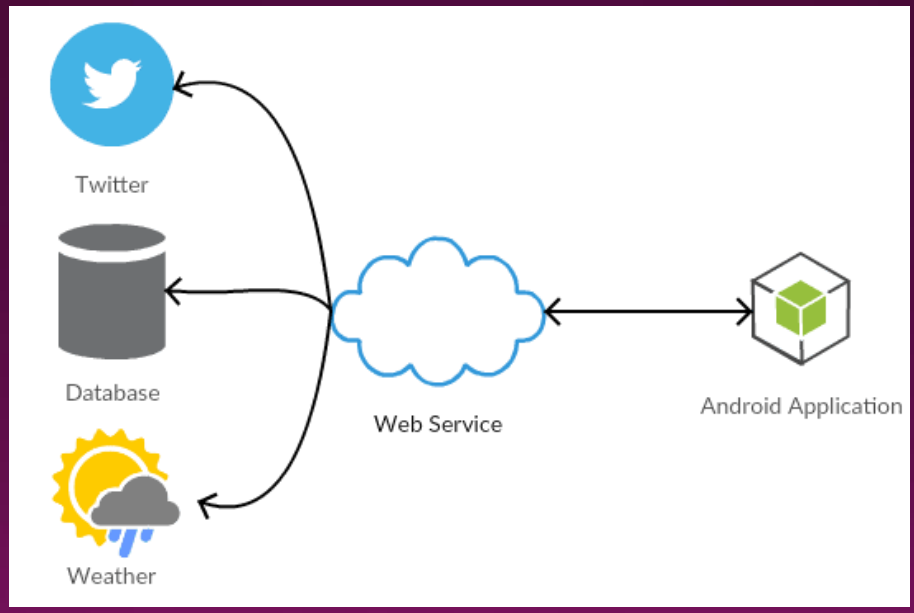
Topics to discuss

1. What are Web Services?
2. Types of Web Services
3. Introduction to APIs
4. Understanding RESTful Web Services
5. Getting Started with REST APIs
6. Using APIs in Real Life
7. Introduction to JSON
8. Building a Simple Web Service
9. Testing and Exploring APIs
10. Web servers

What are Web Services?

- Web services are a standardized way of integrating web-based applications using open standards over an internet protocol.

- Protocols like:
  - HTTP
  - SOAP
  - REST
  - XML
  - JSON
  - WSDL
  - Swagger / OpenAPI
  - GraphQL



- They allow different systems to communicate with each other, regardless of the platforms or languages they are built on, by exchanging data in a predefined format.

Basic concepts

Concept	Meaning
Service	A unit of functionality exposed over the network
Request and Response	The fundamental interaction in web services where a client sends a request to a service, and the service sends back a response.
Protocol	The set of rules defining how messages are formatted and transmitted between clients and services. Common protocols include HTTP, SOAP, and REST.
Endpoint	The specific URL or network address where a web service can be accessed.
Message Format	The structure in which data is exchanged between clients and services. Common formats include XML, JSON, and sometimes even plain text or binary.
WSDL	A standardized XML format used to describe the functionality offered by a web service.

Basic concepts

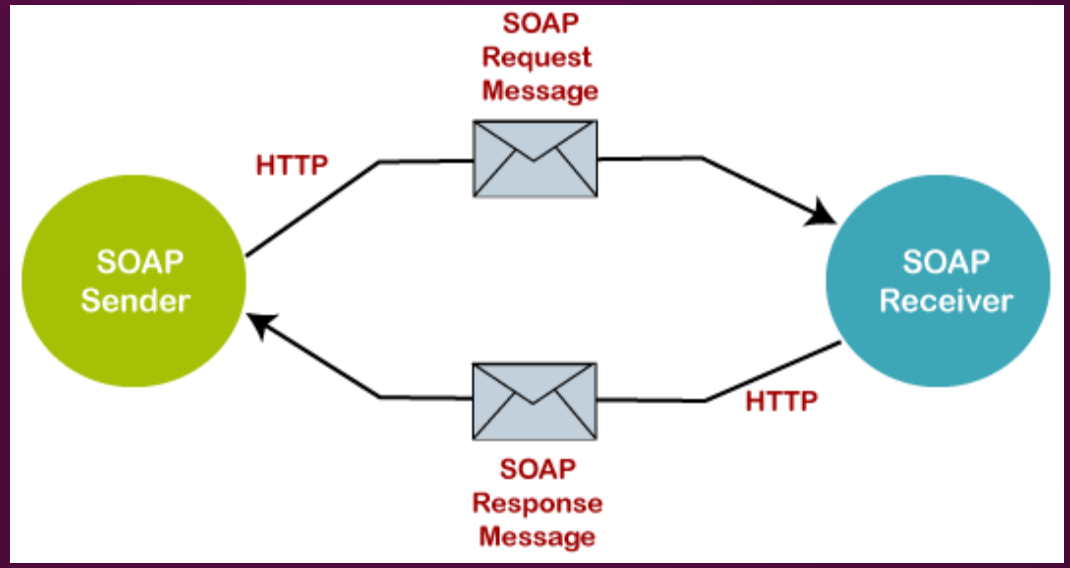
Concept	Meaning
SOAP	A protocol that relies on XML & for exchanging structured information in the implementation of web services.
REST	An architectural style for designing networked applications. RESTful services use standard HTTP methods (GET, POST, PUT, DELETE) to perform CRUD operations on resources identified by URLs.
API	The interface provided by a web service that defines how clients can interact with it.
Security	Techniques and protocols used to ensure the confidentiality, integrity, and authenticity of data exchanged between clients and services. Common security mechanisms include HTTPS, OAuth, JWT, etc.
Versioning	The practice of managing changes to a web service's interface over time to ensure backward compatibility and smooth upgrades for clients.
Documentation	Comprehensive and accurate documentation is essential for understanding how to use a web service effectively.

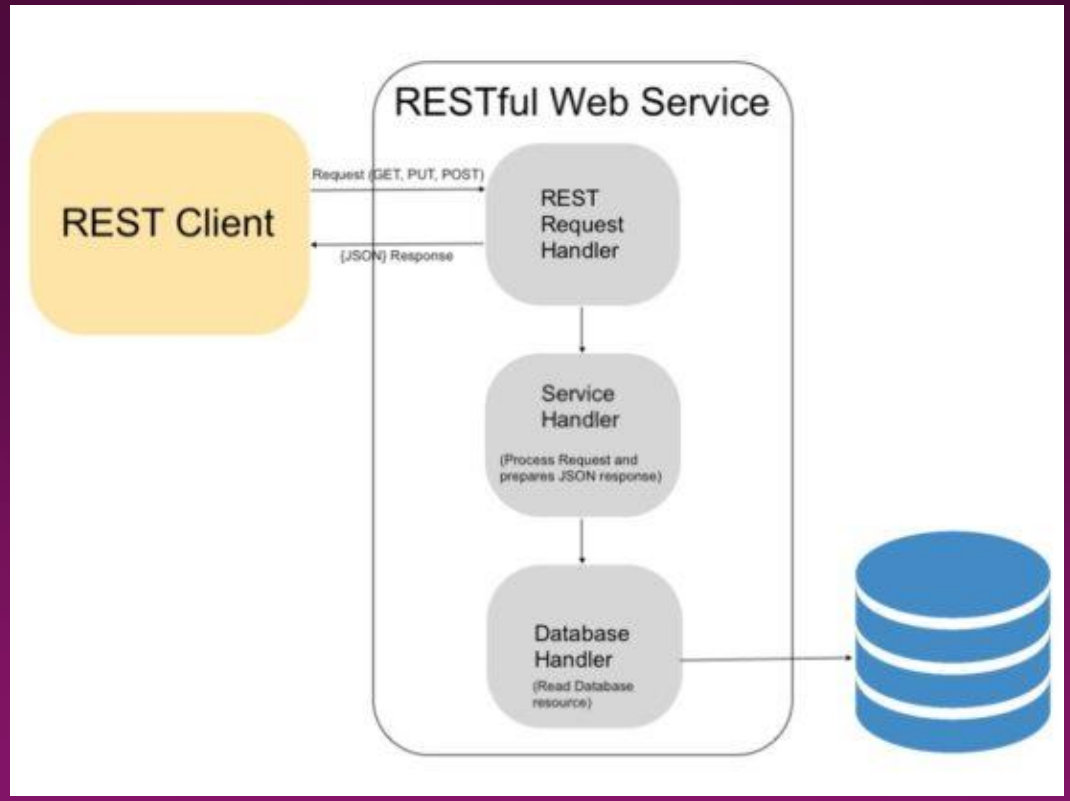
Examples of Everyday Web Services

- Weather API
- Social Media APIs
- Payment Gateways
- Mapping and Geolocation APIs
- E-commerce Platforms
- Messaging and Communication APIs
- File Storage and Sharing APIs
- Travel and Booking APIs
- Health and Fitness APIs
- News and Content APIs

Types of Web Services

- SOAP Web Services
- RESTful Web Services
- JSON-RPC and XML-RPC
- GraphQL
- WebSocket Services





Overview of SOAP

- Stands for “Simple Object Access Protocol”.
- SOAP is a protocol for exchanging structured information in the implementation of web services.
- SOAP is a message protocol that enables the distributed elements of an application to communicate.
- SOAP can be carried over a variety of standard protocols, including the web-related Hypertext Transfer Protocol (HTTP).
- SOAP is a lightweight protocol used to create web APIs, usually with Extensible Markup Language (XML).
- It supports a wide range of communication protocols across the internet, HTTP, SMTP, TCP.

Key characteristics of SOAP

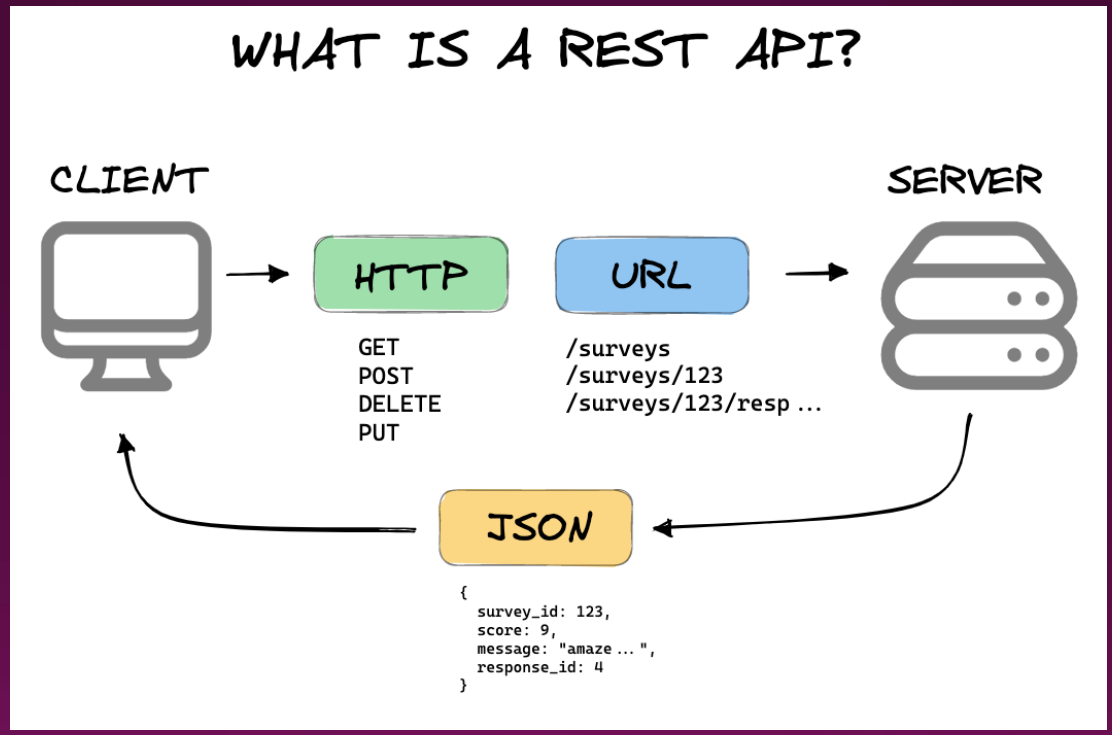
- Protocol
- XML-Based
- Transport Agnostic
- WSDL (Web Services Description Language)
- Strong Typing
- Error Handling
- Complexity

Breaking down SOAP

- Communication Protocol
- Structured Messages
- Transport Options
- Describing Services
- Error Handling

Overview of REST

- REST means Representational State Transfer.
- Key characteristics of REST are:
  - Architecture Style
  - HTTP-Based
  - Resource-Oriented
  - Representations
  - Statelessness
  - Hypermedia
  - Simplicity



Breaking down REST

- Architecture Style
- Resource-Based
- HTTP Methods
- Statelessness
- Representation
- Uniform Interface
- State Transfer
- HATEOAS (Hypermedia as the Engine of Application State)

SOAP vs REST

Feature	SOAP	REST
Type	Protocol	Architectural Style
Data format	XML only	XML, JSON, Plain Text, etc.
Performance	Slower due to larger messages	Faster due to smaller messages
Scalability	More complex to scale due to statefulness	Easier to scale due to statelessness
Security	Requires additional security layers	Inherits security from underlying protocol (HTTPS)

Comparison with Real-life Analogies

Sending Mail:

- SOAP is like sending a registered letter through the postal service.
  - It's formal, with a predefined structure and clear instructions.
  - Each letter contains detailed information, including the sender, recipient, and contents, and follows a specific delivery protocol.
- REST is like sending a postcard.
  - It's lightweight and flexible, with less strict formatting requirements.
  - You write down the essential information (e.g., recipient's address) and drop it in the mailbox.
  - The postcard's format is simpler, and there's no need for formalities like registration or signatures.

Introduction to APIs

- APIs, or Application Programming Interfaces, serve as the bridge between different software applications, allowing them to communicate and interact with each other.
- An API is a set of rules and protocols that define how different software components should interact with each other. It specifies the methods, parameters, and data formats that applications can use to communicate.
- APIs enable developers to build upon existing software components without needing to understand their internal workings.

APIs

- Types of APIs:
  - Web APIs
  - Library APIs
  - Operating System APIs
  - Hardware APIs
- Key Concepts:
  - Endpoints
  - Requests and Responses
  - Authentication and Authorization
  - Rate Limiting

Communication Between Different Systems

- Standardized Interface
- Abstraction of Complexity
- Encapsulation of Functionality
- Interoperability
- Data Exchange
- Integration
- Service Composition
- Scalability and Flexibility

Popular APIs

- Social Media APIs
- Mapping and Geolocation APIs
- Payment Processing APIs
- E-commerce APIs
- Messaging and Communication APIs
- Cloud Services APIs
- Weather APIs
- News and Content APIs

RESTful Web Services

- Resource-Oriented
- HTTP Methods
- Statelessness
- Uniform Interface
- Data Formats
- State Transfer

Basic Components of a RESTful API

- Resources
- Uniform Resource Identifiers (URIs)
- HTTP Methods
- Representations
- Headers
- Status Codes
- Hypermedia Controls (HATEOAS
- Authentication and Authorization

What is JSON ?

- JSON (JavaScript Object Notation) is a lightweight data interchange format that is easy for humans to read and write, and easy for machines to parse and generate.
- It is based on a subset of the JavaScript programming language, although it is language-independent, meaning it can be used with any programming language.
- JSON is commonly used for transmitting data between a server and a web application, as well as for storing configuration data and transmitting structured data over the internet. It is often used as an alternative to XML due to its simplicity and ease of use.

JSON

- JSON is represented in Key-Value pair.

- Example:

```
{
  "name": "John Wick",
  "age": 30,
  "isStudent": false,
  "address": {
    "street": "123 Main St",
    "city": "New York",
    "zipCode": "10001"
  },
  "languages": ["English", "Spanish", "French"]
}
```