**Introduction to Web Services**

**Web Services** are software modules that allow applications to communicate over a network—typically the internet. They enable **interoperability** between different platforms, programming languages, and systems.

**Key Benefits:**

* Platform-independent communication
* Language-neutral data exchange
* Reusable components for distributed systems

**2. What Are APIs?**

An **API (Application Programming Interface)** is a set of rules and protocols that allow two software systems to interact.

**Web APIs:**

* Expose application functionality over the web
* Use HTTP as the transport protocol
* Enable integration with third-party services

**Example**:  
Google Maps API allows developers to embed maps and location services into their apps.

**3. HTTP Methods in REST APIs**

RESTful APIs use standard HTTP methods to perform operations on resources:

|  |  |  |
| --- | --- | --- |
| **Method** | **Purpose** | **Example Use Case** |
| GET | Retrieve data | Fetch user profile |
| POST | Create new data | Add a new user |
| PUT | Fully update existing data | Replace user details |
| PATCH | Partially update data | Update user email only |
| DELETE | Remove data | Delete a user account |

**4. SOAP vs REST**

|  |  |  |
| --- | --- | --- |
| **Feature** | **SOAP** | **REST** |
| Type | Protocol-based | Architectural style |
| Data Format | XML only | JSON, XML, plain text, HTML |
| Complexity | Heavyweight | Lightweight |
| Security | Built-in WS-Security | Relies on HTTPS, OAuth, etc. |
| Speed | Slower due to XML overhead | Faster and more efficient |
| Use Case | Enterprise systems, banking, telecom | Web/mobile apps, microservices |
| Standards | Strict (WSDL, XSD) | Flexible (URI-based resources) |

SOAP is ideal for **transaction-heavy, secure environments**, while REST is preferred for **scalable, fast, and flexible web services**.

**5. JSON vs XML Payloads**

**JSON (JavaScript Object Notation)**

* Lightweight and easy to read
* Native to JavaScript
* Widely used in REST APIs

**Example**:

{ "id": 1, "name": "Ruchitha" }

**XML (eXtensible Markup Language)**

* Verbose and tag-based
* Supports attributes and nested elements
* Common in SOAP services

**Example**:

<user id="1">

<name>Ruchitha</name>

</user>

**6. Common HTTP Status Codes**

|  |  |  |
| --- | --- | --- |
| **Code** | **Meaning** | **Description** |
| 200 | OK | Request succeeded |
| 201 | Created | Resource successfully created |
| 400 | Bad Request | Malformed or invalid request |
| 401 | Unauthorized | Authentication required |
| 403 | Forbidden | Access denied |
| 404 | Not Found | Resource not found |
| 500 | Internal Server Error | Server-side failure |

These codes help clients understand the outcome of their requests and handle errors gracefully.