18. Caching Web Services:
Caching in web services refers to the practice of storing copies of frequently accessed or computed data to improve response time and reduce the load on the server. It involves temporarily storing the results of repetitive operations so that subsequent requests for the same data can be served faster.

19. Differences Between REST APIs and SOAP APIs:
REST (Representational State Transfer):

- **Communication Style:** REST is based on a stateless client-server architecture.
- **Protocol:** Typically uses HTTP as the communication protocol.
- **Data Format:** Commonly uses lightweight formats such as JSON or XML.
- **State:** Stateless, meaning each request from a client contains all the information needed to understand and process the request
- **Flexibility:** Generally considered more flexible due to its simplicity.

SOAP (Simple Object Access Protocol):

- **Communication Style:** Uses a more rigid and formal contract-based approach.
- **Protocol:** Can use various protocols, including HTTP, SMTP, or more.
- **Data Format:** Typically relies on XML for message format.
- **State:** Can maintain stateful communication through the use of sessions.
- **Complexity:** Generally considered more complex and heavyweight.

Example of REST API

GET /api/users/123

Content-Type: application/json

```
{
  "id": 123,
  "name": "John Doe",
  "age": 25
}
```

Example of SOAP API

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">

  <SOAP-ENV:Body>
    <m:GetUserDetails xmlns:m="http://example.com/api">
      <m:UserID>123</m:UserID>
    </m:GetUserDetails>
  </SOAP-ENV:Body>
```

</SOAP-ENV:Envelope>

20. HTTP Methods Supported by REST:

RESTful APIs commonly use the following HTTP methods:

- **GET:** Retrieve data from the server. It should not have any side effects on the server.
- **POST:** Submit data to be processed to a specified resource. It can result in the creation of a new resource or the updating of an existing one.
- **PUT:** Update a resource or create it if it doesn't exist at the specified URI.
- **DELETE:** Request the removal of a resource at a specified URI.
- **PATCH:** Apply partial modifications to a resource.
- **OPTIONS:** Retrieve information about the communication options available for the target resource.
- **HEAD:** Retrieve the headers of a resource without the actual data.