

2.1 Exercise #2.1 - Explaining Concepts

I have referred to the [Simple Object Access Protocol Overview](#) and [Representational State Transfer](#) for this document.

Comparative Study of SOAP and REST

This article briefly describes the Simple Object Access Protocol (SOAP) and Representational State Transfer (REST). It also highlights their key differences and the advantages of using one over the other in different scenarios.

SOAP

SOAP is an XML-based protocol maintained by the World Wide Web Consortium (W3C). This protocol defines standard rules and messaging format for information exchange between web applications in a platform- and language-agnostic manner.

SOAP-compliant web applications expose SOAP APIs as the interfaces to rest of the web. These applications request information from or send information to other web applications as SOAP messages through the SOAP APIs. The SOAP APIs create and send standard SOAP messages understood by other applications' APIs, as well as interpret the received SOAP messages into a format understood by their underlying applications.

REST

REST is a set of architecture principles that help web applications exchange information in a platform- and language-agnostic manner, just like SOAP. However, unlike SOAP, REST is not standardized and has simple and flexible implementations.

Applications following the REST principles expose REST APIs as interfaces. Other applications can invoke these REST APIs to request or send information simply by using the APIs' Uniform Resource Identifiers (URIs) over the web. The APIs respond with messages that are lightweight and have flexible format, unlike the SOAP messages.

Key Differences

SOAP	REST
SOAP messages are XML-based, have rigid structure, and are heavier due to mandatory headers and footers. Therefore, SOAP messages require more resources for transmission.	REST messages are lightweight, have flexible structure, and can be plain text, HTML, XML, or JSON-based. These messages can be easily transmitted to and understood by machines and applications.

SOAP	REST
SOAP supports communication over multiple transfer protocols, including SMTP (email), FTP, TCP, apart from HTTP/HTTPS (web).	REST supports communication over HTTP and HTTPS protocols only, and therefore suitable only for web applications.
SOAP has in-built security compliance that is more comprehensive than SSL. Therefore, SOAP can ensure secure transmission of messages.	REST supports SSL and HTTPS, and is suitable for secure web applications. However, security not as robust or comprehensive as that from SOAP.
SOAP has in-built transaction compliance (atomicity, consistency, isolation, and durability, or ACID) to ensure reliable database transactions.	REST lacks transaction compliance.
SOAP has in-built success/retry mechanism to handle communication errors and failure, providing error-resilient and reliable messaging.	REST lacks success/retry mechanism. It instead enables API implementations to include error codes in responses. API-invoking applications receiving such responses are expected to have their own error-handling mechanism.
Requests to SOAP APIs are not cacheable. A fresh request must be sent to the API to get the information every time.	REST supports caching of data that can reduce the frequency of request to the REST API. Therefore, REST can ensure faster web applications and less load on servers.
<p>The attributes mentioned above make SOAP suitable for secure enterprise applications with no resource restrictions that may need to communicate over various protocols apart from HTTP.</p> <p>Further, ACID-compliance, security, and success-retry mechanism make SOAP ideal for applications requiring secure and reliable financial transactions, such as banking applications, telecom billing systems, stock trading applications, payment and money transfer systems, and ticketing solutions among others.</p>	<p>The attributes mentioned above make REST suitable for faster and interactive web applications providing real-time information with minimal security requirements. Some examples can be weather applications, stock market ticker apps, flight schedule information, sports score tickers, and so on.</p> <p>Further, with simple REST APIs and lightweight REST messages, REST is ideal for mobile web and apps, Internet of Things (IoT), and cloud applications.</p>