

Context for Using REST vs. SOAP Web Services Overview

Background

- **Audience:** Engineers needing to determine when to use REST or SOAP web services.
- **Percentage of I wrote:** 100%.
- **Original content or revised content:** The document is my original writing.
- **Resources for drafting content:** The content derived from my research of SOAP and REST web services. Resources consist of 7 websites. I wasn't sure if this question was requesting me to list references, but they are available upon request.
- **Content editing:** I edited the document myself as someone confident in my editing abilities because of my background. The document underwent moderate editing.
- **Style guide usage:** A company style guide was not used to write this document. I used a general document template that I created. The template is a good template for content that will be published as online content (transferred from a Word document to a wiki or website), which factored into why the headings are not numbered. When a style guide is not applicable, I create a style sheet when needed to ensure consistency within a document.
- **Additional useful context, such as deadlines, achievements, etc.:** To provide samples for a job opportunity, I could provide samples I had already written or choose from topics provided by the company. Because some of my samples did not meet the sample requirements, I chose one of the topics provided by the company. My deadline was a week after I received the sample instructions and there was a 2-page limit. An achievement for me is creating a sample quickly about topics I wasn't familiar with prior to my research.
- **Changes after publication and why:** The document was not published.

How I Went Above and Beyond

I created a template with headings to enhance readability and help users scan the document and find sections of interest easily.

How to Determine when to Use SOAP or Rest Web Service for API Design

Introduction

When needing to send data over the web from system to system, whether it is communication between applications, data, or devices, this communication occurs through application programming interfaces (APIs). Because you can build applications using many different programming languages, architectures, and platforms, the difficulty of exchanging data between varied technologies is solved through using web services to build APIs. Two web services used for internet data exchange are Simple Object Access Protocol (SOAP) and Representational State Transfer (REST). The content in this document is specific to SOAP and REST and determining which to use in any given case.

Overview of SOAP and REST

SOAP

SOAP is a communication protocol that defines a set of rules for message or data exchange between systems, using XML for message services. Following predefined rules, SOAP functionality allows clients to create, retrieve, update, and delete records from a server. Because of the security and strict standards, SOAP is often preferred for enterprise-level applications where features like tight security, compliance, and complexity satisfy requirements in enterprise environments.

REST

REST is an architectural style that uses standard HTTP principles for designing APIs and easily integrates with existing websites. Four commands are used to perform create, retrieve, update, and delete operations on data or services: GET, POST, PUT, and DELETE. Because of simplicity, scalability, flexibility, and compatibility with the HTTP protocol, REST is often used in modern web development and power applications.

Factors to Consider when Choosing SOAP or REST

SOAP and REST are not created equal, so knowing which to choose for your use case is important in project success. When choosing the best web service for your use case, consider the following:

- Programming language being used.
- Application environment or existing infrastructure.
- Requirements of the service (e.g. complexity of the request or needed capabilities).
- Resources and bandwidth of resources.

- Ease of use for the use case (e.g. simplicity, flexibility, standards compliance, etc.).

SOAP and REST Comparison

SOAP	REST
General	
Shares data by using an API that defines communication rules	Shares data by using an API that defines communication rules
A protocol for communication between systems	An architectural style for designing communication interfaces
Language, platform, and transport independent	Language, platform, and transport independent
Messaging/Data Exchange	
Less flexible - limited to XML for data exchange and is very structured or strict, not tolerating errors and controlling every aspect of data exchange	More flexible - allows data exchange in various formats (e.g. plain text, XML, YAML, JSON, and more) with JSON being more common
Supports stateless (not storing messages exchanged with a client) by default but can be stateful (storing messages exchanged with a client). Hard to scale when stateful	Stateless (not storing messages exchanged with a client)
Security and Use Cases	
Supports multiple transfer protocols (e.g. HTTP, SMTP, and UDP, and more)	Limited to an HTTP-based infrastructure or transfer protocol
Secure through WS with WS-Security, supporting message-level security and providing end-to-end security. Also supports SSL/TLS	Secure through HTTPS and SSL/TLS for transport-level security. Message-level encryption is not used
Commonly used in enterprise-level applications requiring strict standards and security and legacy systems	Often preferred for mobile and web applications and other modern use cases
Performance	
Considered more reliable because of built-in error handling.	Considered less reliable
Strictness can inhibit speed	Freedom in design and therefore faster because of no strict specifications

Takeaways

No two use cases are the same, so doing a deep dive into needs and desires is key to choosing the right web service. SOAP and REST are two capable methods that can make a difference in productivity, profitability, and reach. More research may be needed to further understand the two web services. However, this overview is a good starting point for understanding some key differences and determining which web service fits the needs.