REST vs. SOAP

REST and SOAP are both solutions to the same overall problem: how an application should access web services. Both methods have their own strengths and weaknesses, so it's important to consider what your application is trying to achieve when choosing the right solution.

What is REST?

REST stands for REpresentational State Transfer and is an architectural style that establishes standards for computer systems on the web and makes it easier for systems to communicate with each other. A REST-compliant system, or RESTful system, must have several unique characteristics:

- A uniform interface between components
- A client-server architecture
- Stateless client-server communication
- Cacheable data
- A layered system constraint

REST was specifically designed to be used with lightweight web or mobile applications and utilize normal HTTP protocol to transfer data. Since REST is a set of guidelines, it allows developers a lot of flexibility in determining how best to use them.

What is SOAP?

SOAP stands for Simple Object Access Protocol. The main idea behind SOAP is to ensure programs built on different platforms and languages could exchange data and communicate with each other. SOAP messages are XML documents defined by built-in rules.

When a request is made to a SOAP API it can be handled through any of the application protocols such as HTTP, TCP, SMTP, and others. However, once the request is received SOAP will always return a response as an XML document. All SOAP requests must be sent to the server each time as they cannot be cached by the browser.

SOAP is often preferred in enterprise-level solutions due to its guaranteed level of reliability and security which make it well-suited to large scale applications that require reliable database transactions.

REST vs. SOAP

Now that you have a better understanding of the basics of REST and SOAP it's time to look at how they compare to one another.

- REST is an architectural pattern and SOAP is a protocol.
- SOAP has specific requirements like XML messaging, whereas REST offers flexible implementation because it is a set of guidelines.
- REST APIs are lightweight and ideal for web-based and mobile applications. SOAP web services feature built-in security and transaction compliance, but this also makes SOAP web services heavier.
- SOAP exposes its functionality by using service interfaces while REST uses Uniform Service locators to access resources.
- SOAP cannot use REST, but REST can make use of SOAP.

When to use REST

You should consider the following factors when deciding whether to use REST:

- Your application has limited resources and bandwidth. If your application has limited resources, REST is the preferred option because it is a more lightweight solution than SOAP.
- Caching is a requirement. RESTful systems include caching as a basic feature.
- The application can perform tasks without needing to maintain its state. Statelessness is a key feature of RESTful systems.

When to use SOAP

The following should be considered when determining if SOAP is a good fit for your application:

- You need a guaranteed level of security and reliability. As a protocol, SOAP has built-in rules and constraints that must meet established standards.
- The application requires stateful operations. If you need to maintain information about the state during different processes, SOAP has the required structure to support it.
- You need to define specific requirements for communication. SOAP is a strong solution if you need to establish rigid specifications for how data is transferred and what information is included.