



VS



Herbert G. Sri Harsha

Date 7/10/2020

Table of Contents

Explaining Concepts.....2

How to Play Tic-Tac-Toe.....4

Explaining Concepts

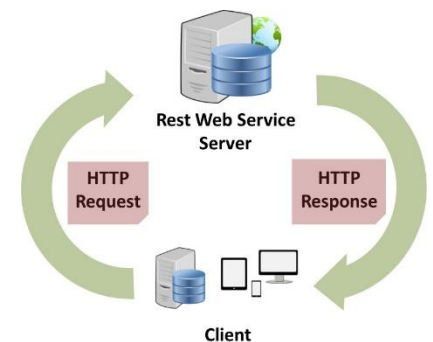
Short overview that helps an engineer choose **when to use REST or SOAP web services**.



REST is a lightweight, highly scalable, and maintainable architecture. It is commonly used to create APIs for web-based applications that specifically works with media components, files, or even objects on a hardware device. It deploys multiple standards, so it takes fewer resources and less bandwidth. REST uses four different HTTP 1.1 verbs (GET, POST, PUT, and DELETE) to perform tasks.

Functionalities of REST

- It's a **Data-driven** approach.
- Supports SSL and HTTPS security.
- Supports XML, JSON, Plain Text, HTML, and other messaging formats.
- Unsuitable for distributed environments, less security.
- High Performance, Scalability, Flexibility, and browser friendliness.



RECOMMENDED



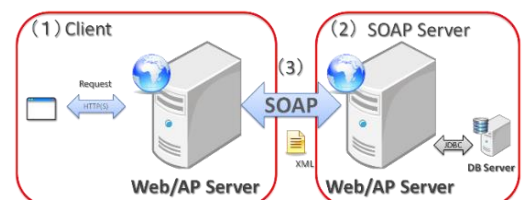
Public web services, Social Networks, and Mobile Services.



SOAP is platform-independent and operating system independent. It works on HTTP protocol because HTTP is supported by all Internet browsers and servers. It communicates between applications running on different operating systems, with different technologies and programming languages.

Functionalities of SOAP

- It's a standard protocol with predefined rules to follow.
- It's a **Function Driven** approach.
- Provides WS-Security with SSL support.
- Requires more resources and high bandwidth.
- Supports only **XML**.
- Built-in error handling.
- It is more complex, slow performance, and less flexible.



Payment Gateways, Healthcare, Financial and Telecom services, and so on.

When to choose REST Web Services



Choose REST web services in the following instances:

- ✓ **REST works with limited resources and less bandwidth.**
- ✓ **Statelessness:** Requires no state of information from one request to another request.
- ✓ **Caching:** Need to cache a lot of requests REST is the perfect solution.
- ✓ **Ease of coding:** Coding and implementation are easier in REST.
- ✓ **Public Webservices:** All public web services use REST.

When to choose SOAP Web Services



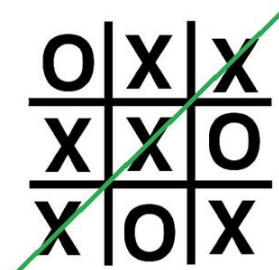
Choose SOAP web services in the following instances:

- ✓ **Continuous processing of secured requests:** if there is a requirement of continuous requests with secured processing SOAP is the only choice.
- ✓ **Firm Agreement:** For online purchasing and banking SOAP is reliable.
- ✓ **Stateful Operations:** Applications that require state needs to be maintained from one request to another SOAP supports it.
- ✓ **Enterprise Webservice:** All enterprises still use SOAP because of its secured transactions.

How to Play Tic-Tac-Toe

Write a short document that describes how to play tic-tac-toe.

Tic-tac-toe is a fun game that comes under the category of board games; it is played between two players; its origin is traced back to ancient Egypt. An early version of tic-tac-toe was played in the Roman Empire, around the first century BC.



Prerequisites to play Tic Tac Toe:

1. No age limit to play.
2. It is not gender-biased.
3. We need white paper, a pen, and an opponent.
4. Simple common sense is required to play.

Procedure to play Tic Tac Toe:

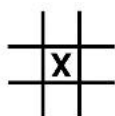
1. **Draw the board** with the combination of 2 rows and 2 columns combined, that should exactly look like 3 x 3 grid of squares.
2. **Player 1** starts the game by marking any of the symbols (X/Y/O/1/) in any of the squares.
3. **Player 2** starts by marking in the rest of the empty squares (Player 2 should choose a different symbol from the player 1).
4. Both the players should mark the squares in turns.
5. The **winner** of the game is decided based upon the 3 consecutive same symbols marked vertically, horizontally, or diagonally (as shown in the image).
6. The game is set to **draw** if none of the players mark 3 consecutively same symbols.
7. When all 9 squares are filled the game is **completed**.

Let's illustrate with an example:

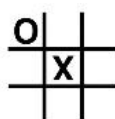
Step 1. Draw a 3x3 grid of squares.



Step 2. Player 1 chooses the symbol **X** and marks anywhere in the empty grid.



Step 3. Player 2 chooses **O** and takes his turn to mark in the empty squares of the grid.



Step 4. Player 1 takes his turn by marking in the rest of the empty squares, keeping in mind to make 3 consecutive same symbols horizontally/vertically/diagonally to win.

o		x
	x	

Step 5. Player 2 takes his turn by blocking player 1 game to win by marking his symbol diagonally (in this case).

o		x
	x	
o		

Step 6. Player 1 blocks player 2 by marking vertically.

o		x
x	x	
o		

Step 7. Player 2 blocks player 1 by marking horizontally.

o		x
x	x	o
o		

Step 8. Player 1 has a chance to win by marking a new entry at the bottom of the square.

o		x
x	x	o
o	x	

Step 9. This is the final chance for player 2 to block player 1 by marking the top middle square.

o	o	x
x	x	o
o	x	

Step 10. Player 1 finishes the game as a **draw** by marking the bottom right square.

o	o	x
x	x	o
o	x	x

Step 11. Suppose if the player 2 fails to mark at the top middle square, Player 1 wins my marking in that middle square.

o	x	x
x	x	o
o	x	

Note: The same scenario applies in all the levels of completion to win whether horizontally/vertically/diagonally.