



Sreenidhi Rajakrishnan 

Automation Architect
@sreenidhirajakrishnan

REST vs SOAP

ASPECT	REST	SOAP
Full form	Representational State Transfer	Simple Object Access Protocol
What it is	Architectural style, conventions	Protocol, strict W3C spec
Message format	JSON, XML, text, HTML	XML only (envelope)
Transport	HTTP / HTTPS only	HTTP, SMTP, TCP, JMS
Contract	OpenAPI / Swagger (optional)	WSDL (mandatory)
Payload weight	Lightweight, low bandwidth	Heavy, verbose envelope
Performance	Faster, smaller payloads	Slower, XML parsing cost
Caching	Native via HTTP GET	Not supported natively
State	Stateless by principle	Stateless or stateful
Security	HTTPS, OAuth 2.0, JWT, API key	WS-Security + SSL, message-level
Transactions	No built-in support	ACID via WS-AtomicTransaction
Reliability	Client handles retries	WS-ReliableMessaging built in
Error handling	HTTP status codes (4xx / 5xx)	SOAP Fault XML block
Endpoint style	Resource URIs, /users/1	One endpoint + SOAPAction
Coupling	Loosely coupled	Tightly coupled to contract
Schema validation	JSON Schema (optional)	XSD inside WSDL (enforced)
Versioning	URI or header based, /v2	New WSDL / namespace
Browser + mobile	First-class native support	Poor, needs a SOAP client
Learning curve	Easy, reads like the web	Steep, specs and namespaces
Automation tools	REST Assured, Postman, Karate	SoapUI, JMeter, Postman
Assertion focus	Status code, JSON path, headers	XML node, XPath, fault code
Mocking	WireMock, Postman mock server	SoapUI mock service
Best fit	Public APIs, microservices, mobile	Banking, telecom, legacy ESB
Biggest risk in test	Undocumented contract drift	Brittle XML / namespace changes
Interview question	"Why REST over SOAP?"	"When is SOAP still required?"

Verdict

REST for anything new, web, mobile, microservices. SOAP where a formal contract, transactions and WS-Security are non-negotiable: banking, payments, telecom.

Say this in the interview

"REST is an architectural style over HTTP exchanging lightweight JSON around resources. SOAP is a strict XML protocol with a WSDL contract plus built-in security and transaction standards. I pick REST for speed and reach, SOAP when the client demands a guaranteed contract, and I test REST with REST Assured, SOAP with SoapUI."

Don't say

"SOAP is dead." It still runs core banking and telecom. Say legacy-heavy, not obsolete.



Sreenidhi Rajakrishnan Automation Architect @sreenidhirajakrishnan



Follow for More on AI & Automation