

Glossary

 support.smartbear.com/algo.html

A

Agent

Agents are software packages that run a set of tests from different locations, often as part of a cloud deployed solution. They are also known as “remote agents” or “load agents”.

Alert

A notification about failed tests in [Secure](#) containing additional information about the response and possible reasons for failure.

API

Application Programming Interface (API) is the way a computer program interacts with another computer program. It is possible and very common that one computer program uses the services of another computer program, and that they are connected through a network. This is often referred to as Service Oriented Architecture (SOA).

A computer that offers an API is a server, and a program that uses the service is a client. A supplier does not have to know anything about the client. All it needs to do is to answer specific questions that the clients ask.

These services will be released independently of each other. They can be developed using different tools, different languages, and by different teams. There are many differences, but a common denominator is the way they communicate. They offer an API that others may or may not use through a network. Also, a common property is that they do not have a user interface.

A SOAP API is a standardized communication protocol for XML-based message exchange. SOAP can use different protocols like HTTP or JMS. In SOAP, all parameters a server needs to be able to respond should be in a W3C standardized XML format and part of the body in a request.

REST web services encode most parameters in a request URL, instead of an XML. Data is encoded in the URL or as request parameters, where encoding is not standardized by W3C.

Assertion

Assertions are used to validate functionality, usually through messages by comparing parts of the message (or the entire message) to some expected value. Any number of assertions can be added to a test step, each validating some different aspect or content of a response.

They are very important for testing at the most basic level. Using assertions, you can verify that you got the expected result. For example, you can verify a proper connection, correct response format, valid data returned, and many other aspects of testing.

For more information, see [About Assertions](#).

B

Bandwidth

Bandwidth is used in [Virt Behavior](#) to simulate the capacity of the network.

Basic Authorization

Basic access authorization is a method for an HTTP user agent to provide a username and password when making a request.

HTTP Basic authorization uses static, standard HTTP headers to provide authorization information.

C

Congestion

Congestion is used in [Virt Behavior](#) to simulate overworked nodes in the network, or other capacity problems.

Continuous Integration

Continuous Integration (CI) is a development practice that requires developers to integrate code into a shared repository several times a day. Each check-in is then verified by an automated build, allowing teams to detect problems at an early stage. The main aim of CI is to prevent integration problems.

CRUD HTTP Methods


Most database operations can be described using the CRUD acronym: you Create data, you Read data, you Update, and, finally, you Delete data.

These operations describe the lifecycle of data in a database. RESTful web services map CRUD operations using the HTTP verbs - GET, PUT, POST, PATCH, and DELETE.

A common mapping between CRUD operations and HTTP verbs is:

CRUD operation	HTTP Verb
Create	PUT with a new URL
	POST to a base URL returning a newly created URL
Read	GET
Update	PUT with an existing URL
	PATCH

CRUD operation	HTTP Verb
Delete	DELETE

 The exact behavior varies with the exact RESTful service.

Client-side Certificate Authorization

Digital certificates provided by clients in TLS or SSL can be used as authorization tokens by verifying the certificate authenticity.

Custom property

Custom Properties are used for storing values in a Project and can be used to transfer or manipulate those values across the entire testing solution.

A Custom Property is a named string value accessible to Scripts, Property Transfers, or Property Expansions.

D

Data-driven testing

Data-driven testing means taking larger amounts of test data (input, expected output, and so on) in some external storage (database, spreadsheet, xml-files, and so on) and using it iteratively in your tests.

Delay

Delay is used in Virt Behavior to simulate time delays in service response time.

Distributed Mode

The Distributed Mode setting is used when running load test scenarios. When running in distributed mode, tests are run on the agents to which the tests have been deployed.

The opposite of the distributed mode is the local mode.

Dynamic Properties

E

Encoding

A format used for request messages. Encoding can be defined in the TestStep properties, and also overridden by adding a `.vmoptions` file to the `bin` directory.

Endpoint

A message destination. It can be part of a target URL (including the port number).

For example, for the following URL:


`http://maps.googleapis.com/maps/api/geocode/xml?address=Rio&sensor=false`

the endpoint is **<http://maps.googleapis.com>**.

Environments

A ReadyAPI feature that allows you to quickly change the test setup depending on the target environment.

Using environments, you can easily switch between different sets of SOAP and REST services, project properties, and database connections.

 Environments contain only project properties.

Event

A notification of something happening in the service.

Typically, events are handled in test step predefined features, but it is also possible to create custom Event Handlers through scripting.

Event Handler

An Event Handler handles Events. When a particular event occurs, a matching event handler is automatically called to perform various tasks or modifications.

F

Form-Based Authorization

A form-based authorization uses a form to provide a username and password (or other credentials) that are handled by dedicated processes in the backend.

G

GSSAPI

Generic Security Services Application Program Interface (GSSAPI) is an IETF standard for an application programming interface for programs to access security services.

H

HATEOAS

HATEOAS is an acronym for *Hypermedia as the Engine of Application State*. It means that clients only interact with applications through dynamic hypermedia provided by servers.

Clients only need a general understanding of hypermedia to interact with services. Basically, it is the server allowing another server to discover how to interact, rather than having it defined beforehand.

Header

HTTP messages start with headers. Headers transfer information about the message and the ways it should be handled, rather than the message body.

HTTP Authentication methods

ReadyAPI supports the following authentication methods:

HTTP Return Codes

With the HTTP protocol, all HTTP verbs return a response code in the form of a number value. For example, code 200 is returned if everything is OK.

There are a number of return codes, which are grouped by types:

Code Group	Description
Informational	Response codes providing server information (Example: 100 means continue , indicating that the server is reachable, but there is no result yet). These are not common in RESTful applications.
Success	The client action was successful. (Small exception for 202, which means the request was accepted and is being processed, with the actual result pending).
Redirection	Sends the calling application to some other resource.
Client Error	The request from the client was broken somehow and needs to be fixed.
Server Error	The server has problems.

Examples

Return code	Meaning	Example
200	OK	The request has succeeded. Information in the response depends on the request method.
201	Created	The request has been fulfilled and a new resource is created.
202	Accepted	The request has been accepted, processing is not completed.
401	Not authorized	The request requires user authorization.
404	Not found	The server cannot find anything matching the Request-URL. May be temporary or permanent.
500	Internal server error	The server could not fulfill the request due to an unexpected condition.
503	Service unavailable	The server is overloaded or in maintenance.

HTTP Methods

HTTP methods are actions performed on a resource. See HTTPS methods in the table below:

Method	Description
GET	Retrieves information from the service.
POST	Sends data to the service.
PUT	Replaces current representations with the content.

Method	Description
DELETE	Removes all current representations in the target URL.
HEAD	Retrieves the status line and header section only from the service.
OPTIONS	Describes the communication options for the target.
TRACE	Performs a loop-back test for the path to the target resource.
PATCH	Updates parts of the resource.

J

JMS

The Java Message Service (JMS) is an API for sending messages between two or more clients. It allows the communication between different components of a distributed application to be loosely coupled, reliable, and asynchronous.

JMS Delivery Mode

The JMS delivery mode can add insurance to the delivery of messages.

 The mode only applies to the delivery, not storage after the receipt.

JMS Header

A JMS header contains system-level information common for all messages, such as the destination and the time when it was sent, while the body contains only application-specific data.

A header can also contain some application-specific information, stored as keyword/value properties. However, not all providers allow an arbitrary amount of data to be stored in a header, it is a good idea to keep most application-specific data in the message body.

JMS Message Selector

When a receiver subscribes to a destination, it can specify a message selector, which acts as a filter for weeding out messages the receiver does not want to see. The message selector must be specified in terms of the message header. For example, a receiver can specify a message selector saying that it wants to see only messages whose JMSType is atg.das.Startup. The message selector can refer to both system-level and application-specific header properties.

JMS NON_PERSISTENT

Non-persistent messages can occasionally be lost without negative effect.

JMS PERSISTENT

Persistent messages cannot be lost in transit without causing problems.

JMSPriority

JMSPriority indicates the importance of the message. The priority is on a level from 0 (the least important) to 9 (the most important). A level up to 4 is considered "normal", and over 5 is considered "expedited".

JMSPeelyTo

JMSReplyTo indicates a JMS Topic or JMS Queue where replies should be addressed.

JMSType

JMSType is the most important header value. It is a string that is used to identify what kind of message is being sent. Handlers often examine JMSType to see how they should handle an incoming message.

JNDI

The Java Naming and Directory Interface (JNDI) is an implementation-independent Java API for directory services that allows Java software clients to discover and look up data and objects by names. Additionally, it specifies a Service Provider Interface (SPI) that allows directory service implementations to be plugged into the framework.

K

Kerberos Authorization

Kerberos uses *tickets* to provide mutual authorization between the client and the server, through the symmetric key cryptography with a trusted third party. Kerberos uses UDP port 88 by default.

KeyStore

KeyStore is used by the KeyManager class in Java. KeyManager handles authorization credentials for SSL. A keystore is needed to create self-signed certificates, since signing requires a private key.

KeyStore contains private keys and is required for the SSL connection and client authorization.

A private key is used by servers to provide a certificate from KeyStore corresponding to public keys of a client.

L

Latency

Latency is used in Virt Behavior to simulate time delays in the network for responses to requests.

Local Mode

The Local Mode setting is used when running load test scenarios. When running in local mode, tests will only be run on the local computer, and not on any agents.

The opposite of the local mode is the distributed mode.

M

Media Type

A media type is a two-part identifier for file formats.

For details, see <http://www.iana.org/assignments/media-types/media-types.xhtml>.

Mock

Mocks (mock services) are simulated, approximated, virtualized representations of actual services.

Service Mocking is used when you cannot or do not want to run your tests against the actual service.

In ReadyAPI, mocks are called Virts.

MTOM

Message Transmission Optimization Mechanism (MTOM) is a method of efficiently sending binary data to and from Web services.

For more information, see [MTOM at the W3C site](#).

Must Understand

The Must Understand (**mustUnderstand**) attribute is used in SOAP headers to indicate if the entry is mandatory for processing.

A mandatory header must be processed by the receiver. If the header is not recognized, the request will fail with a SOAP fault.

N

Negotiate Authorization

Negotiate authorization is a Microsoft protocol that allows a client and a server to dynamically agree on one of several types of authorization.

NTLM Authorization

NTLM is a Microsoft authentication protocol which uses HTTP request/response headers to provide authorization.

O

Object Properties

OpenAPI

Open Core

A product offering where parts of the system (the core) are offered free with limited

features, and the full version is made available as a commercial product.

In ReadyAPI, the core offering is called **Base**, and the full version is called **Pro**.

For more information, see the Wikipedia article [Open Core](#).

Overhead

Any handling of service calls involves some amount of overhead – parsing of data, encoding, creation of files, and so on.

Obviously, as the load increases, it becomes more important to bring down overhead as much as possible.

One way of bringing down overhead is to use a less verbose protocol for requests and responses (for example, JSON instead of XML).

Another way would be to eliminate parsing of encoded data. In a RESTful web service, parameters are (mainly) encoded in the call itself, which eliminates the need for separate parsing of data or files (at least for that part of the service).

Since SOAP relies on XML, it tends to have more overhead than REST.

P

Passport Authorization

Microsoft Passport uses a key shared between Microsoft and a dedicated partner site to create a *cookie* that uniquely identifies the user for single-sign-in.

Property expansion

Property Expansions provide common syntax to dynamically insert ("expand") property values during the processing.

This can be used wherever properties are used: requests, values, file names, endpoints, and so on.

The [Property Expansion](#) can contain [Groovy Script](#).

Both nested and recursive property expansions are supported.

For more information, see [Property Expansion](#).

Property Transfer

A transfer of a property between places in a project: for example, taking a value from a response and passing it to a request.

Usually, the method used is the [Property Transfer](#) test step.

Q

QName

A qualified name of an element or attribute in XML content. It is used to identify and access them.

R

RAML

A REST API format based on YAML for service definition of RESTful interfaces. The file describes resources, actions/methods, query/header/path parameters, representations, and so on.

For more information, see <http://raml.org>.

Regular Expression

Coded strings that define an infinite number of possible matches.

For more information, see: [Java Regular Expressions](#).

Request Timeout

Request Timeout is an optional test step setting. If no response has been received within this time, the test step is considered as failed.

REST

Representational State Transfer (REST) is a more lightweight protocol than SOAP for the HTTP communication, where parameters are mostly encoded in the request URL, but can also be sent in the message body.

REST Method

A REST Method is the HTTP Verb used to access a [resource](#).

REST Parameter

REST parameters can be used in your requests to parameterize your resources by [Levels](#) and [Styles](#).

REST Parameter Level

Parameters can be defined at either a resource or method level.

Parameters at a resource level are inherited by all contained methods and all requests in these method nodes.

Parameters on a method level are only inherited by all requests in that method.

REST Parameter Style

[REST requests](#) have five types of parameters:

REST Representation

REST Representation Type

Representations can be of three basic types: **REQUEST** , **RESPONSE** , and **FAULT** .

Type	Description
REQUEST	An entity sent to the target web server from the client.
RESPONSE	An entity sent by the server to the client in response to the request sent by the client earlier.
FAULT	An entity sent by the server to the client that equates to the error.

REST Request

REST Requests are calls to a REST service. REST Request can include HTTP Headers and parameters.

REST Resource

A part of the target URL, referring to the object to be accessed with REST methods.

For the URL

<http://maps.googleapis.com/maps/api/geocode/xml?address=Rio&sensor=false>
the resource is [/maps/api/geocode/xml](#) .

RESTful

The term RESTful means the same as REST, but sometimes is used to mean web services that implement REST methods and the architecture. They do not require a WADL to be defined. It is possible, and common, to skip this definition. This means that the service definition needs to be implied, or provided elsewhere.

Richardson Maturity Model

A scale for evaluating services and their RESTfulness. It consists of four levels:

1. **HTTP transport** – Remote interaction via tunnel, no web mechanisms.
2. **Resources** – Separate resources instead of a single endpoint.
3. **HTTP Verbs** – Standardized usage of HTTP verbs to perform actions.
4. **Hypermedia controls** – HATEOAS is implemented.

For web services to be RESTful in a meaningful sense, they need to reach level 3 of the Richardson Maturity Model.

S

Schema Annotation

Comments added to the web service definition schema used to describe the nodes.

Schema Interface

Schema inference is a process of generating a Schema from a set of messages.

Service Definition

A machine readable description of an API interface, defining available resources, actions or operations, valid request-response pairs, and other communication properties.

SOAP

Simple Object Access Protocol (SOAP) is an XML-based communication protocol and encoding format for inter-application communication.

A SOAP API is defined as a standardized communication protocol for XML-based message exchange. SOAP can use different protocols like HTTP or JMS.

All parameters, the server needs to be able to respond to, should be in a W3C standardized XML format and should be part of the body in a request.

The current specification is version SOAP 1.2, though version 1.1 is more widespread. SOAP is widely viewed as a backbone to a new generation of cross-platform cross-language distributed computing applications, termed Web Services.

SOAP Actor

An attribute used to address the Header element to a specific endpoint.

An actor is an application that can both receive SOAP messages and forward them to the next actor. The ability to specify one or more actors as intermediate recipients makes it possible to route a message to multiple recipients and to supply header information that applies specifically to each of the recipients.

SOAP Operation

SOAP Operations bind abstract operations to actual SOAP operations in the service, defining the request and response format.

SOAP Request

SOAP Requests are XML-based calls to a service, which (optionally) contain SOAP parameters as part of the request body.

SSL

Secure Sockets Layer (SSL) establishes encryption between a server and client at the transport level.

Stateless

When you design your service to be stateless, you are making it treat each new request as an independent event. It does not store sessions and information about connecting clients. If your service is stateless, you do not need to have a specific client to stay connected to a specific server. Instead of a session with multiple steps between a client and a single server, you get multiple pairs of request-response traffic between a client and multiple

service instances. With the growth of traffic, you can add service instances and servers to handle the load. It does not matter where a specific client call ends up, since you do not need to keep track of it.

Stored Procedure

A subroutine available to applications that access a relational database system. A stored procedure is stored in the database data dictionary.

Swagger

The former name of the OpenAPI format of REST service specifications. This name was also used to name a framework of API developer tools for creating, editing, verifying and visualizing RESTful web services.

In January 2016, Swagger Specification was renamed to OpenAPI Specification. Since that time, the name Swagger denotes a toolset for working with OpenAPI REST APIs.

T

Test Case

Series of test steps run in sequence. A test case is successful when all steps are successful.

For more information, see the Cases section.

Test Step

A single action taken to test a web service, for example: login, logout, request.

For more information, see the Steps section.

Test Suite

Collection of test cases.

For more information, see the Suites section.

TrustStore

TrustStore is used by the Java TrustManager class. TrustManager handles authorization determines whether a connection is trusted.

TrustStore stores public keys and Certificate Authorities(CA) certificates.

The SSL client uses SSL certificates, usually .cer files, stored in TrustStore to verify the server identity.

See also: KeyStore.

TTL

Time To Live (TTL) is the duration of the message validity.

U

URL Encoding

URL encoding converts characters into a web-safe format.

V

Versioning

API Versioning means providing multiple versions of the service to different clients.

A web service may serve clients from many different sources. These clients must be able to understand the API. Yet, you want to change and improve the API constantly. This means that a service will have to deal with clients with different specifications and requirements, providing newer functionality for newer clients that can handle it.

The answer to this problem is versioning. This means keeping multiple versions of your service active and letting clients connect to the version that is appropriate for them.

It is recommended to support 3-5 past versions, or any version released within a year, whichever is greater.

Virt

Simulated, approximated, virtualized representations of an actual service.

Virting is used when you cannot or do not want to run your tests against the actual service.

In previous versions of SoapUI, Virts were called Mocks.

Virt Action

A Virt Action is used in REST Virts to define responses. An operation can contain a number of responses, which are selected according to the Dispatch Method.

The corresponding term for SOAP Virts is Operation.

Virt Operation

A Virt Operation is used in SOAP Virts to define responses. An operation can contain a number of responses, which are selected according to the Dispatch Method.

The corresponding term for REST Virts is Action.

Virt Response

A Virt Response is used in SOAP and REST Virts to contain the message sent back in reply to incoming Actions or Operations.

W

WADL

Web Application Description Language (WADL) is the REST equivalent to WSDL. An important difference is that the WADL is not required.

A REST based web service (REST Service) may be defined by a WADL descriptor file. It may look like this:

```
<application xmlns="http://wadl.dev.java.net/2009/02">
<resources base="http://example.com/api">
<resource path="books">
<method name="GET"/>
<resource path="{bookId}">
<param required="true" style="template" name="bookId"/>
<method name="GET"/>
</resource>
</resource>
</resources>
</application>
```

A WADL definition is similar to WSDL in the way that it defines what you can do. It defines the supported HTTP verbs. In this example, we can see that we are able to get information about books using the HTTP method - **GET** . We can also notice that there is a required resource that must be part of the request.

For more information, see a [W3C standard suggestion](#).

Web Service

A service available on a network that allows other systems to communicate with it using a defined protocol. The web part indicates that the service is using transport protocols designed for the World Wide Web, and it uses HTTP for communication.

Web service discovery

Web pages provide links you can follow to get more information and navigate through a web site. In the same way, RESTful web services can, by implementing [HATEOAS](#), provide a way for clients to navigate through the API, perform web service discovery to create an API representation describing the interface.

Browsing web services is not supported in SOAP.

WSA Action

A required element of the **xs:anyURL** type, provides the **action** property. The children of this element provide the value of this property. It indicates the intent of the SOAP HTTP request and may provide information about the operation to invoke.

WSA IsReferenceParameter

The isReferenceParameter attribute is a required boolean which indicates if the message addressing header is a reference parameter.

WSA Parameter

WS-Addressing reference parameters are used to interact with endpoints as needed.

They contain element information items that are required to properly interact with an endpoint, or are issued by the endpoint.

Reference parameters for different endpoints can be different within the same WSDL.

WSA Relationship

A pair of values that indicates how a message relates to another message.

The type of the relationship is identified by an absolute IRI.

The related message is identified by an absolute IRI that corresponds to the related message's `message id` property.

The message identifier IRI may refer to a specific message, or be the following pre-defined URL that means unspecified message.

WSDL

Web Service Definition Language (WSDL) is a W3C web service description language standard for describing SOAP web services, using XML.

It describes network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information. The operations and messages are described abstractly, and then bound to a specific network protocol and message format to define an endpoint.

Related concrete endpoints are combined into abstract endpoints (services). WSDL is extensible to allow description of endpoints and their messages regardless of what message formats or network protocols are used for communication, however, the only bindings described in this document describe how to use WSDL in conjunction with SOAP 1.1, HTTP GET/POST, and MIME.

See [Getting started with Schema Inference](#).

WSI Correlation

A header used to provide correlation between the encrypted and unencrypted forms of a message.

WSI Profile

A set of named web service specifications at specific revision levels, together with a set of implementation and interoperability guidelines recommending how the specifications may be used to develop interoperable web services.

WS-RM

WS-ReliableMessaging is a standard for ensuring delivery of requests to the destination service. ReadyAPI supports it at both the request and test case levels.

X

XML-RPC

XML-RPC is a remote procedure call (RPC) protocol which uses XML to encode its calls via HTTP.

XML-RPC uses method calls to the service and passes parameters to the remote method to achieve results.

Parameter styles can be nested or use maps and lists, allowing for objects and structures both as an input and output.

Most HTTP security methods can be used for authorization.

XML-RPC Method

XML-RPC methods are calls made to a service which pass parameters to the remote method.

XML-RPC Parameter

XML-RPC parameters are passed by a #REST_method to a remote method in order to retrieve responses or achieve results.

Parameter styles can be nested or use maps and lists, allowing for objects and structures both as an input and output.

XML-RPC Request

XML-RPC Requests are XML-encoded calls to remote methods (optionally) containing parameters.

XSD

Originally defined as XML Schema, it is a description of a type of an XML document, typically expressed in terms of constraints on the structure and content of documents of that type, above and beyond the basic syntactical constraints imposed by XML itself. An XML schema provides a view of the document type at a relatively high level of abstraction.