Automation Testing

TECHNOLOGY

JMeter Assertions

A Day in the Life of an Automation Test Engineer

Alex is working on JMeter for non-functional testing. As part of his load and performance tests, he tested several applications and their servers.

However, he now wants to ensure that the response he receives from the server is accurate and not affected by the increased load on the server while testing different types of applications. He wants to compare the expected and actual results.

To achieve the above, he must learn about assertions that will help him ensure an accurate and exact response.

Learning Objectives

By the end of this lesson, you will be able to:

- Define assertions
- Importance of assertions
- Analyze type of assertions and their properties

TECHNOLOGY

What Are Assertions?

simplilearn

Assertions

JMeter assertions are used to validate the response received from the server for a specific sampler request. Users can add different assertions as child elements in the sampler request.

The sampler request is marked as failed if an assertion fails; the same is reflected in the test results.

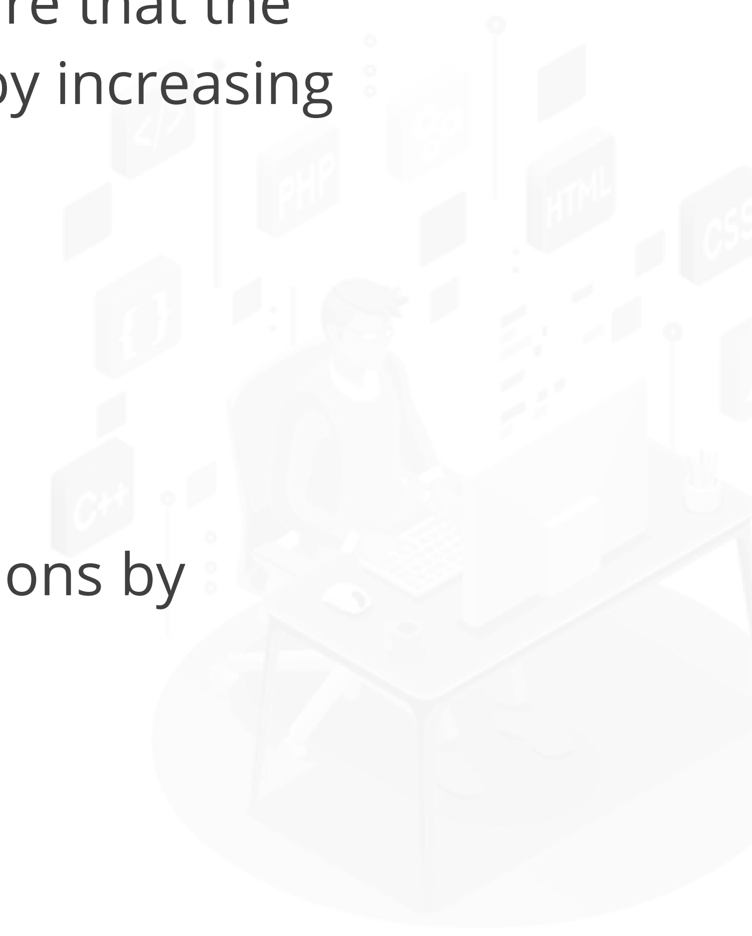
Importance of Assertions

01

Assertions are necessary for performance test scripts to ensure that the response received from the server is correct and not affected by increasing load on the server.

02

They are also useful while testing different types of applications by comparing the actual and expected results.



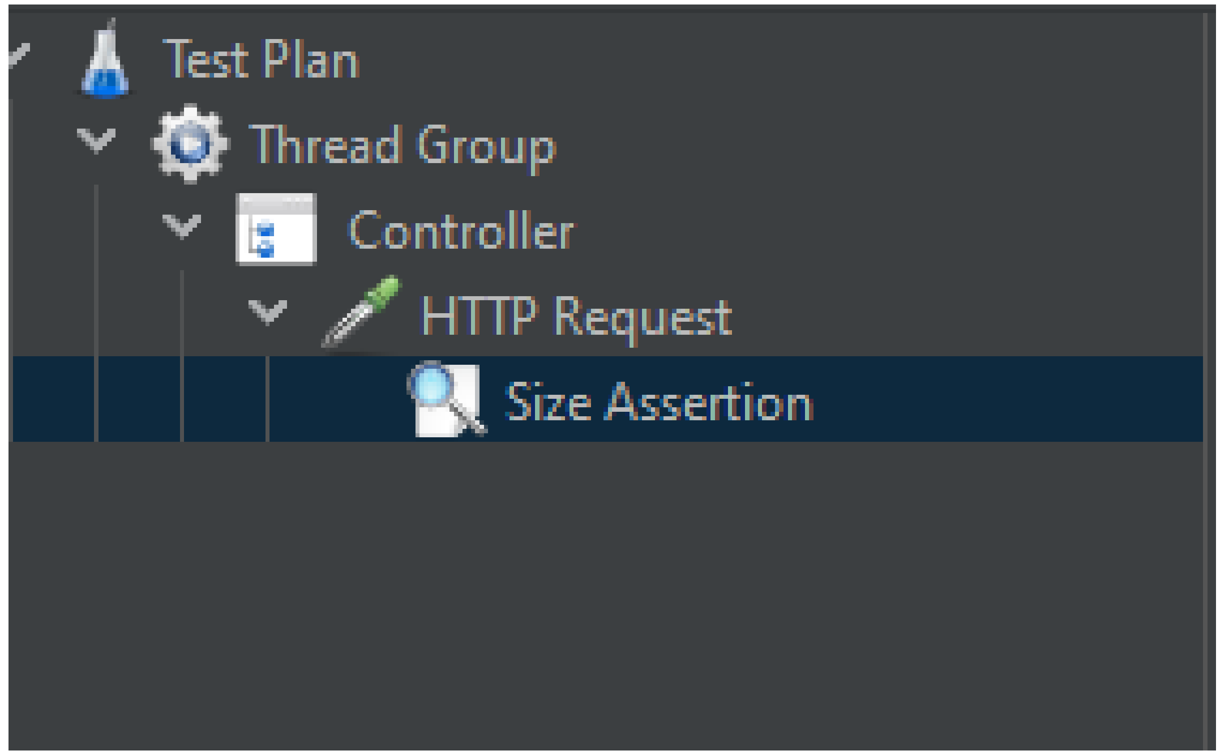
Add Assertion

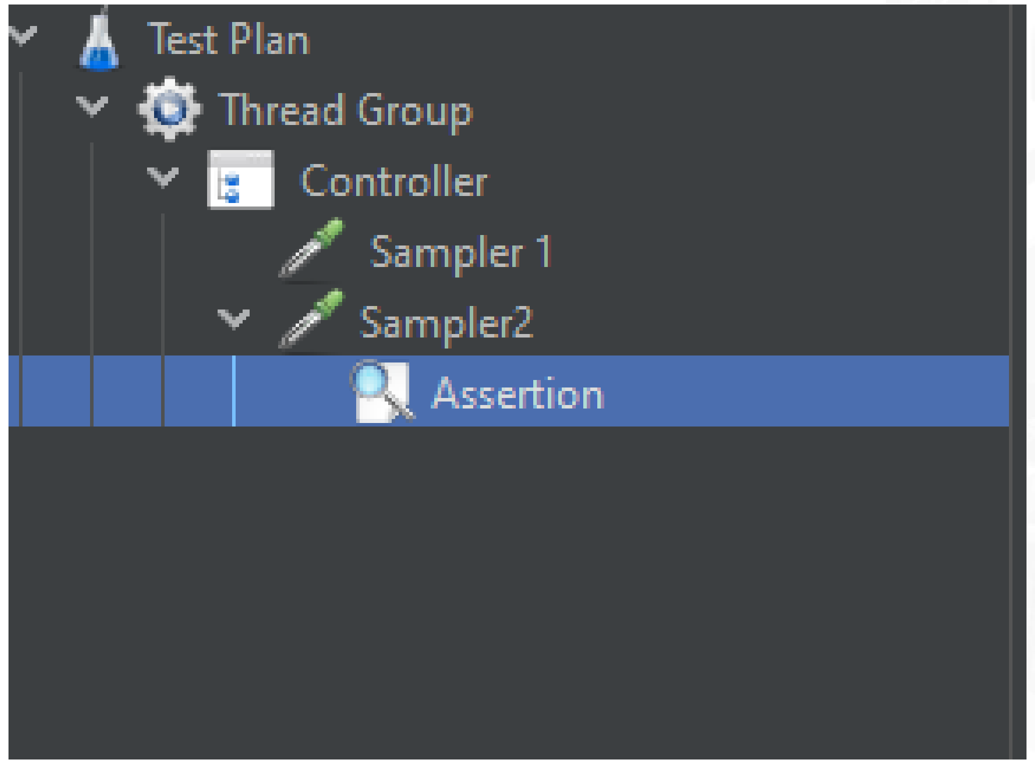
The following steps will guide the user to add assertions:

01	Right-click on **Sampler Request** and select **Add**
02	Hover over the **Add** and then **Assertion**
03	Select the required **Assertion**

Add Assertion

User can decide assertion flow per user requirement:





TECHNOLOGY

Types of Assertions

Types of Assertions

The following is a list of Assertions:

01 Response Assertions

02 Duration Assertions

03 Size Assertions

04 XML Assertions

05 Beanshell Assertions

06 MD5Hex Assertions

Types of Assertions

The following is a list of Assertions:

- 07 HTML Assertions
- 08 Xpath Assertions
- 09 XML Schema Assertions
- 10 BSF and JSR223 Assertions
- 11 Compare Assertions
- 12 SMIME Assertions

Response Assertion

Response Assertion is used in test scripts to validate a pattern within the response body, header, code, or message.

Response Assertion

Various pattern matching rules are used to validate the response, including:

- Contains
- Matches
- Equals
- Substring
- Not

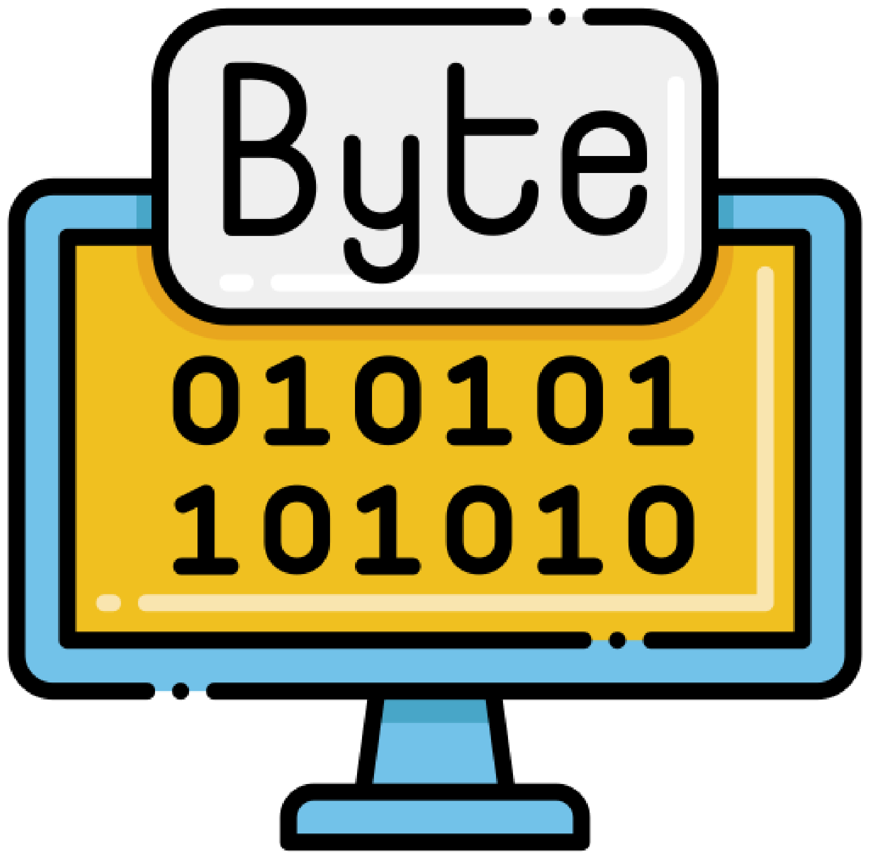
Duration Assertion

Duration Assertion is used to verify that the sampler request is processed within a specified time frame.

If a request lasts longer than the specified duration in milliseconds, the sample will be marked as failed.

Size Assertion

Size Assertion is used to validate the response size based on a specified number of bytes.

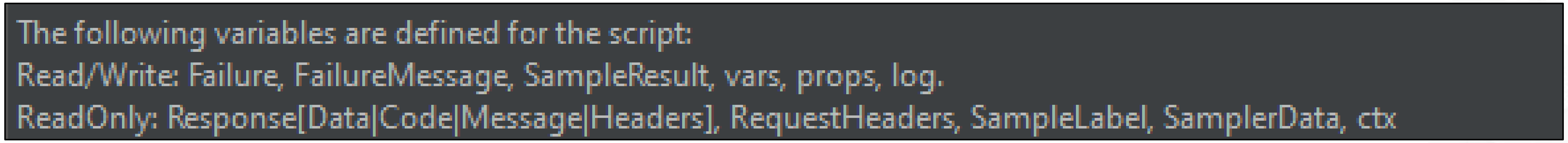


XML Assertion

An XML assertion is used to verify that the response follows a valid XML format.

BeanShell Assertion

A Beanshell Assertion allows users to perform additional checks on a sampler using Beanshell scripting.



The following variables are defined for the script:
Read/Write: Failure, FailureMessage, SampleResult, vars, props, log.
ReadOnly: Response[Data|Code|Message|Headers], RequestHeaders, SampleLabel, SamplerData, ctx

BeanShell Variables

The BeanShell Assertion offers the following variables:

- Failure
- Failure Message
- Response Data
- Response Code
- Response Headers
- Response Message

MD5Hex Assertion

The MD5Hex Assertion is used to validate the MD5 hash value of the response data. MD5Hex Assertion has the following input field:

MD5Hex to Assert

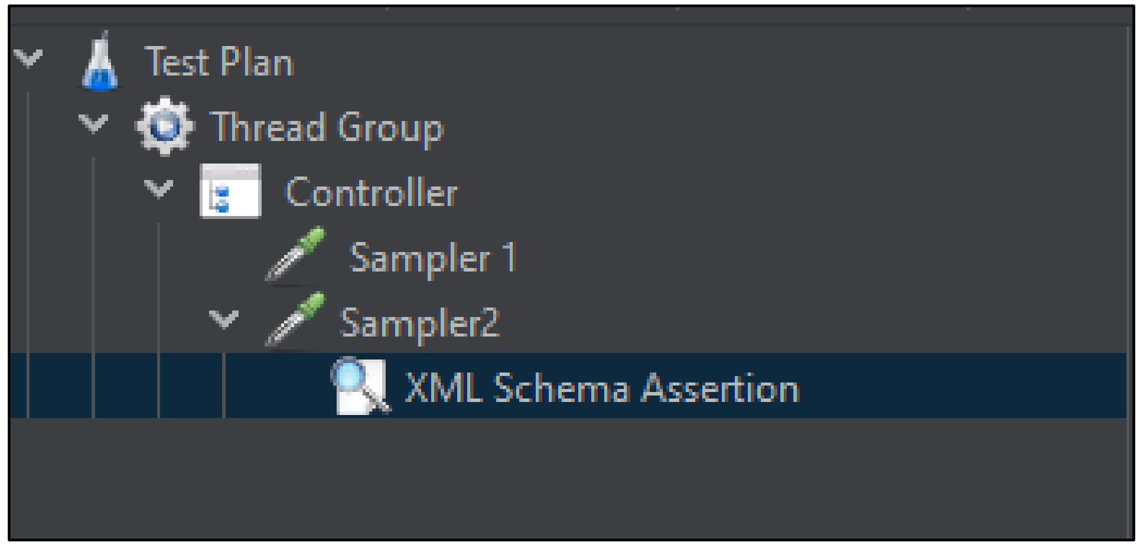
Users need to write the 32-digit MD5 hash value in the **MD5Hex** field of the assertion.

HTML Assertion

The HTML assertion is used to check the HTML syntax of the response.

XML Schema Assertion

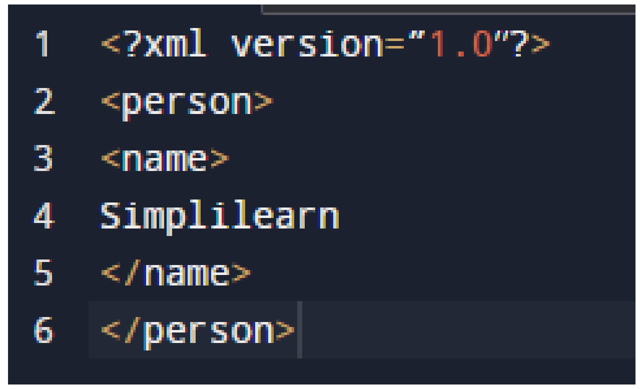
The XML Schema Assertion validates the response against the specified XML schema.



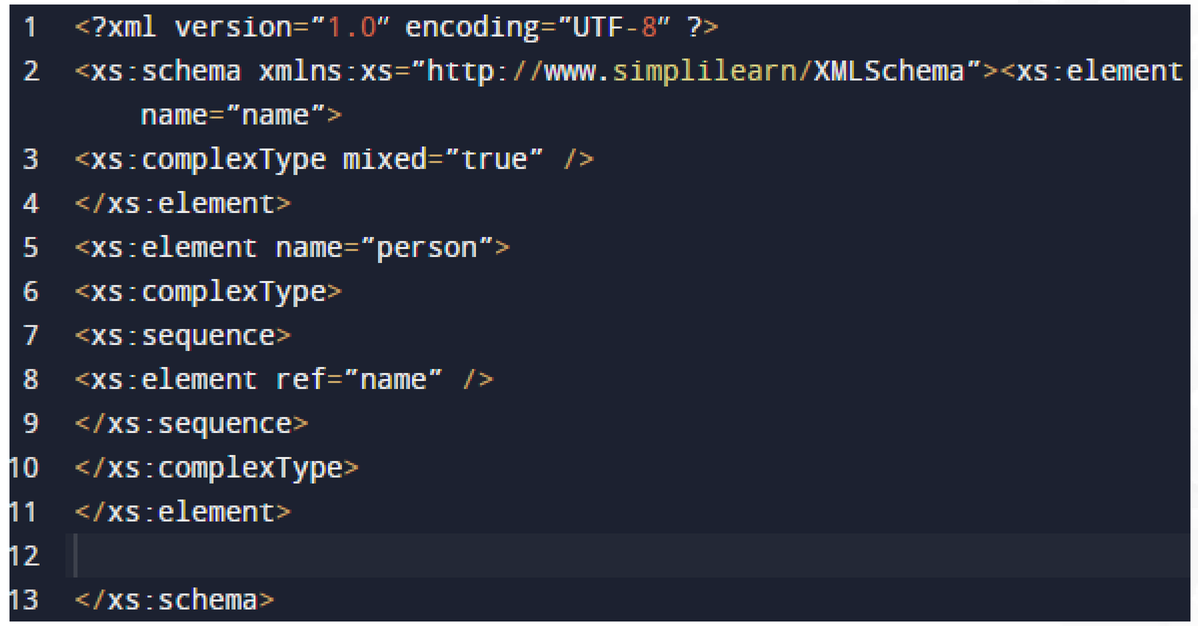
It describes the structure of an XML document. The XML Schema language is also referred to as XML Schema Definition (XSD).

XML Schema Assertion

Users can record the **test.xml** file and then validate it against **test.xsd**.



```
<?xml version="1.0"?>
<person>
<name>
Simplilearn
</name>
</person>
```



```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema xmlns:xs="http://www.simplilearn/XMLSchema"><xs:element
    name="name">
<xs:complexType mixed="true" />
</xs:element>
<xs:element name="person">
<xs:complexType>
<xs:sequence>
<xs:element ref="name" />
</xs:sequence>
</xs:complexType>
</xs:element>

</xs:schema>
```

A user may record their own XML file and validate it according to the schema. XSD generators can be used to generate the schema for an XML file.

Compare Assertion

A Compare Assertion checks the content of the response or confirms that the response time of all samplers within the scope of the assertion is equal.

Assertion Parameters:

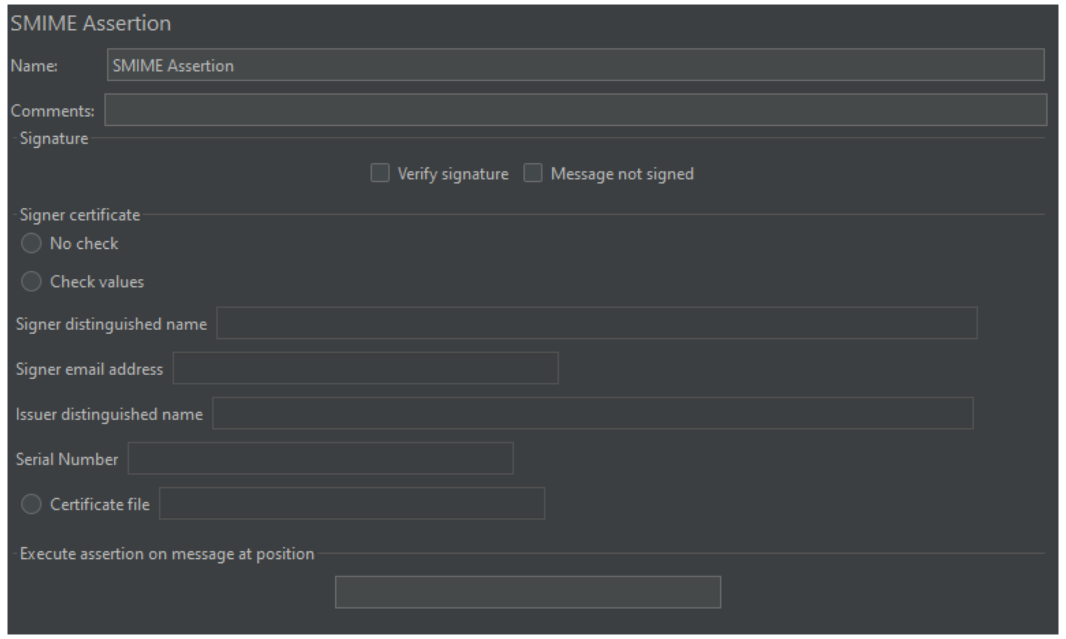
Compare Content

Compare Time

A test plan should include both requests and assertions at the same level.

SMIME Assertion

SMIME Assertions are used to validate MIME messages. A MIME (Multipurpose Internet Mail Extension) is an extension of the original Simple Mail Transport Protocol (SMTP) email protocol.



SMIME Assertion

Name: SMIME Assertion

Comments:

Signature

Verify signature Message not signed

Signer certificate

No check

Check values

Signer distinguished name

Signer email address

Issuer distinguished name

Serial Number

Certificate file

Execute assertion on message at position

It enables users to transfer various types of data files, such as audio, video, images, and application programs, via email.

Key Takeaways

- Assertions are used to validate responses received from the server.
- It can be added to sampler requests as child elements.
- It validates various patterns within the body, like header, code, message, and test scripts.