Automation Testing

Assertion APIs to Evaluate Tests

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main()
{
    cout << "Hello, world!" << endl;
    return 0;
}
```

A Day in the Life of an Automation Test Engineer

Now John has learned the order of executions and dependencies, Test groups and TestNG.xml, and parallel testing TestNG.

He must now know assertion API, which enables verifying HTTP responses in the API tests, various assertion types, and how to add a new assertion.

To achieve the above, he will learn API assertions and types of assertions with a few additional features in this lesson.

Learning Objectives

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

- Describe assertion API
- Discuss how to add a new assertion
- List the types of assertion

Assertion API

Assertion API

Assertions can be used in API tests to validate HTTP responses.

The result of each assertion and the reason it passed or failed can be found on the test result page.

Add a New Assertion

Assertible provides a set of assertions for testing HTTP services.

There is an Assertions tab in each test configuration view. Any assertion for that test can be added, updated, or deleted from there.

Types of assertions

The following are the assertions that are available:

01 Status code

02 JSON path data

03 JSON validation

04 XPath XML data

05 Response header

Types of assertions

The following are the assertions that are available:

06 Response time

07 HTML validation

08 HTML validation

09 Link checker

Assert Status code

All new tests have this assertion enabled by default, and it can be set to any value.

Assert JSON data

Selecting parts of a JSON response and making assertions using JSON Path syntax inspired by the original JSON path implementation are possible with Asserting JSON Response Data.

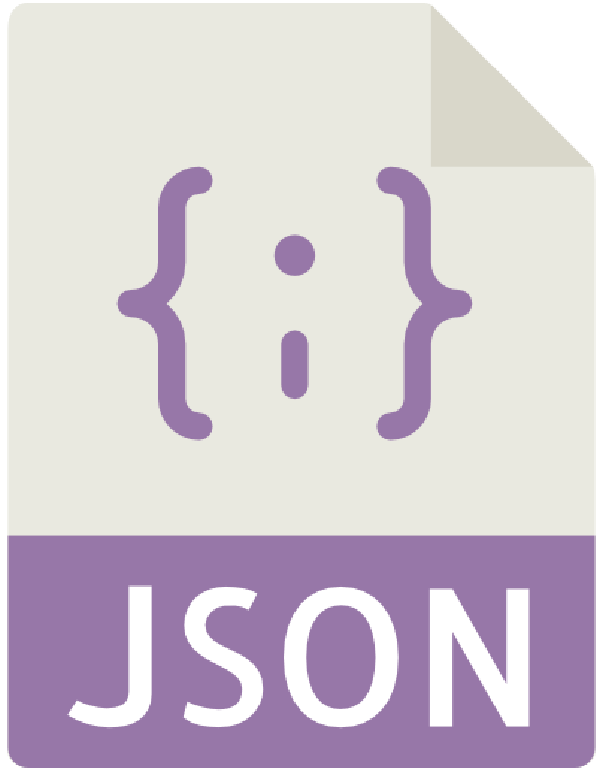
Assert JSON validation

JSON validation verifies that the JSON response bodies are valid. A JSON schema can be specified if desired.

The JSON validation assertion will check that the HTTP response value is valid JSON if no JSON schema is provided.

JSON Schema Validation API

The user can POST the JSON Schema specification and documents to validate and debug validation errors.



JSON Schema

Once the JSON schema is provided, the JSON validation assertion compares the HTTP response data to the schema and reports any errors.

Assert XML or HTML data

Asserting XML response data allows users to use XPath to select parts of an XML or HTML response and make assertions on those values.

Assert Response Header

Response Header Specify the header name to locate the header name field. The type of comparator used influences the target value field.

Assert Response Time

The response time certifies that the total response time of the request is less than a specified value, which is usually 1 second.

Assert Text Body

The assertion looks for any text in the request's response body.

The user can declare that any text is present in the response body by setting the target to that text.

Assert HTML Validation

It is useful for detecting invalid DOM and HTML errors in your webpages, which could prevent them from loading properly on some clients.

Assert Link Check

This assertion will verify that all links on an HTML page and the request to that target are valid.

Built-in Assertions

Core assertions are a set of built-in assertions in Assertible that deal with the default behavior that a test run must satisfy.

Key Takeaways

- Assertions are used to determine a test's current state.
- The null value is not returned if the JSON Path expression tries to select a node that does not exist.
- Assertible XPath support is currently limited to single node selections. Complex XPath returns types are not supported.
- OpenAPI uses an extended subset of JSON schema to define data types in an OpenAPI definition.