

API Test Designer Building Blocks

Appvance IQ Test Designer provides a collection of building blocks that can be combined to create automated tests and performance scenarios. Each building block provides a specific capability, such as sending HTTP requests, validating responses, controlling test flow, working with test data, executing JavaScript, or interacting with external services.

Building blocks are added to a test by selecting them from the **Building Blocks** panel and configuring their properties according to the requirements of the test.

This section provides an overview of the available building blocks. Each building block has its own page with detailed information about its purpose, configuration options, and usage examples.

Request and Service Blocks

These building blocks allow tests to interact with web services and APIs.

- **REST Request** — Sends HTTP requests to REST-based services.
- **SOAP Request** — Sends requests to SOAP-based services.
- **Verify** — Validates values returned by a request or another test step.

REST requests support the standard HTTP methods:

- GET
- POST
- PUT
- PATCH
- DELETE

Test Designer also provides import options for creating requests from existing definitions:

- **Import cURL**
- **Import REST**
- **Import SOAP**

Flow Control Blocks

Flow control blocks determine how and when test steps are executed.

- **Condition** — Executes steps based on a condition.
- **For Each** — Repeats a group of steps for each item in a collection.
- **For Loop** — Repeats steps a specified number of times.
- **While** — Repeats steps while a condition remains true.
- **Group** — Organizes related test steps into a logical group.
- **Delay** — Introduces a wait period during test execution.

Data and Configuration Blocks

These blocks provide mechanisms for supplying, managing, and reusing test data.

- **Data File** — Provides test data from an external data file.
- **Global Params** — Defines parameters that can be shared across a test.
- **Locker** — Stores and retrieves values for reuse during test execution.
- **Include** — Includes reusable content or functionality in a test.

Database and Integration Blocks

These blocks allow tests to interact with external systems and data sources.

- **SQL** — Executes SQL operations against a configured database.
- **Proxy** — Configures proxy-related behavior for test execution.

Scripting and Specialized Blocks

These blocks provide additional capabilities for custom logic and specialized test interactions.

- **JavaScript** — Executes JavaScript code as part of a test.
- **OCR** — Uses optical character recognition to identify and process text from supported content.

Building Block Workflow

Building blocks can be combined to create complete test flows. For example, a test might:

1. Read test data from a **Data File**.
2. Send a request using a **REST Request**.
3. Use **Verify** to validate the response.
4. Use a **Condition** to determine the next step.
5. Execute additional requests or logic inside a **For Loop**.
6. Store values in the **Locker** for use by subsequent steps.

The appropriate combination of building blocks depends on the type of application being tested and the behavior that the test needs to reproduce or validate.

The following pages describe each building block in detail, including its configuration, supported options, and examples of how it can be used in a test.

Last updated: August 12, 2026

If you have feedback about this topic, please contact documentation@appvance.ai.

Revision #3

Created 2026-07-24 14:33:07 UTC by Admin

Updated 2026-10-15 15:53:33 UTC by mmora