Appvance IQ

Release 5.0.0

Release Summary

Document Version 1.0

April 19th, 2024

TOC

About Appvance

Appvance is the technology leader in and inventor of AI-driven autonomous testing, which has revolutionized software testing.

Visit us at Appvance.ai.

Support

Contact our support team to request technical help, report an issue, or suggest a new feature. See the Appvance Support page for more information.

Visit the Appvance Service Desk to submit a request.

Online Documentation

Visit the Appvance Documentation site for AIQ product documentation.

Training and Certification

Your journey to Autonomous Testing Engineer Certification (ATEC) starts here. Take our classes and be among the very first to use and deploy the world’s only Level-5 Autonomous software testing technology.

We offer self-paced classes, scheduled classes and tailored training. See the Training and Certification page for more information.

Content Library

Explore the content library to access a range of resources to help you succeed in all your quality initiatives. Understand best practices, see examples of success, and learn more about AIQ in action.

News

Visit the Appvance Newsroom to read about our most recent announcements and happenings.

Events

Connect with Appvance, learn more about AIQ, and engage in thought-provoking conversations on delivering world-class software quality, autonomous testing, and more.

See where you can meet up with the Appvance team.

Blog

Read the blog and learn about testing strategy & methods, AI, and get AIQ product updates.

Contact Appvance

Online: Contact form

Email: info@appvance.ai

Phone: (845) 254-1164

Mailing address: 3080 Olcott Street Suite B240 Santa Clara, CA 95054

Appvance IQ 5.0.0 Release

The Appvance IQ (AIQ) 5.0 release includes a significant leap forward in the world of AI-native automated testing. This release introduces Generative AI V3, a comprehensive update that enhances performance, reporting, visual object recognition, and bug-finding accuracy.

Read the press release for AIQ 5.0.

About this Document

This document contains information on the following functional enhancements and functional changes in the AIQ 5.0.0 release.

- "AI 3.0 in AIQ" on page 27
- "Introducing the Web Designer" on page 123
- "Mobile Testing Enhancements" on page 161
- "Add a Proxy in the API Test Designer" on page 170
- "Shadow DOM Setting" on page 175
- "Functional Changes" on page 177

This document also contains information on the required dashboard and database upgrades.

- "AIQ 5.0 Database Upgrade" on the next page
- "Updating CICD Dashboard Code Location for AIQ 5.x.x" on page 20

Depending on what AIQ release you are upgrading from to AIQ 5.0.0, there may be additional functional enhancements included in this release. For your convenience, information about the enhancements in the AIQ 4.10 release has been added to this document. See "AIQ 4.10.x Enhancements" on page 183 for more information.

Accessing AIQ 5.0.0

Appvance customers can access the 5.0.0 release page. The release page includes a link to the AIQ 5.0.0 installer, the list of fixes included in the release, information on upgrading from a previous release of AIQ and other implementation considerations.

See the "Upgrading your Controller and Test Nodes" section of the 5.0.0 release page for upgrade information.

See "About Appvance" on page 6 for contact information.

AIQ 5.0 Database Upgrade

Most of the primary keys of the AIQ schema have been migrated to a long data type. (from `INT` datatype to `BIGINT` datatype) This change will prevent you from reaching the limit of an integer value (`-2147483648` to `2147483647`) in the tables of the AIQ schema.

As part of this update additional changes were made to the AIQ architecture including stored procedures and functions.

This upgrade is required when you install AIQ 5.0.0. Contact your Appvance Customer Success representative or Appvance Support if you have any questions about the upgrade.

Upgrade Considerations

You should complete a full backup before starting the upgrade process. This highly recommended in case the upgrade process fails or if you need to revert to a pre-AIQ 5.0.0 release for any version. Once the database has been upgraded it cannot be used for a pre-5.0.0 release of AIQ.

Checking the Max Values in Columns

Running the attached script will find the max value in a column with data type INT. This tells you if the value is close to reaching the limit.

Download the checkMaxValues.sql script from the AIQ 5.0 Database Upgrade topic.

AIQ 5.0.0 Database Configuration

You have two options to implement the database changes. You can create a new database or upgrade an existing database. Below are the requirements and process for each option.

There is a project tracking spreadsheet attached for each option. Use the spreadsheet to track the steps in the process. See the “Project Tracking Information” sub section for each scenario.

Project Tracking Information

Define the following data in the project tracking spreadsheet (attached below) before starting the process in your environment:

- Start time - Start time of the first step.
- End time - End time of each step.
- Responsible - Person responsible for each step.
- Appvance Support Contact - Appvance support person name and email that will help with the process.

Scenario 1: Create a New Database

Requirements

- MySQL client (E.g. MySQLWorkbench)

Project Tracking Spreadsheet

- Download the AIQ-5.0.0 - New Database - Minute By Minute.xlsx file from the AIQ 5.0 Database Upgrade topic.

Upgrade Process

1. Stop service of the current database (If exists).
2. Stop AIQ Test Nodes.
3. Stop AIQ Controller.
4. Prepare the new database. See MySQL 8 Database Installation for instructions.
5. In your MySQL client open a new session as a root user.
6. Open and execute the MySQLInitial.sql script.

> Download the MySQLInitial.sql script from the AIQ 5.0 Database Upgrade topic.

7. Verify env var running `show variables like 'log_bin_trust_function_creators';` . It should be `ON`.

8. Verify that the user `aiq` was created with two locations: `localhost` and `%`.

9. Verify that the schema `aiq` was created with two locations: `localhost` and `%`.

10. Close session with the root user.

11. In your MySQL client open a new session as user `aiq` with password `aiq`.

12. Open and execute the MySQLAppvance.sql script.

> Download the MySQLAppvance.sql script from the AIQ 5.0 Database Upgrade topic.

13. Close session with the `aiq` user.

14. Start AIQ controller.

15. Check database connection. See Results for instructions.

Scenario 2: Existing Database

> This process is intended to be executed by Devops/DBA personnel with enough privileges to manage AWS/RDS instances or with access to server hosting the mysql database.

Before upgrade, make sure that the database is backed up.

- AWS - Snapshot the RDS instance.
- Azure - Snapshot volume of the database instance.

Requirements:

- MySQL client (E.g. MySQLWorkbench).

Preparation

You must gather the following data is required before executing the upgrade process:

- Current Database Space Provisioned/Available. Gather this information from the appropriate source(s):
  - AWS RDS Console
  - Instance provisioned Space
  - Server Hardware
- Current Database Size: Column “`Total_Db_Allocated_Space_Mb`“ from below script
- Expected Grow: Column “`Estimated_Data_Storage_Grow_Mb`“ from below script
- Minimum space required for the upgrade process: Column “`Minimum_Free_Space_Required_For_Upgrade_Mb`“ from below script

Script to gather information

```
select t.TABLE_SCHEMA DB_Schema,
       sum(t.table_rows) Total_Rows,
       max(t.table_rows) Maximum_Rows_In_Table,
       max(t.AUTO_INCREMENT) Maximum_Identity_Value,
       sum(t.data_length)/1024/1024 +
       sum(t.index_length)/1024/1024 Total_Db_Allocated_Space_Mb,
        sum(CASE t.TABLE_NAME
            WHEN 'SummaryAttribute' THEN t.table_rows * 8
            WHEN 'SummaryStatsOnTime' THEN t.table_rows * 4
            WHEN 'DefaultPercentil' THEN t.table_rows * 4
            WHEN 'TypeSuccessFailure' THEN t.table_rows * 4
            WHEN 'Definition' THEN t.table_rows * 4
            WHEN 'TypeLong' THEN t.table_rows * 4
            WHEN 'SummaryElement' THEN t.table_rows * 4
            WHEN 'FailSummaryElement' THEN t.table_rows * 12
            WHEN 'DetailedTypeRequest' THEN t.table_rows * 16
            WHEN 'DetailedElementAttribute' THEN t.table_rows * 12
            WHEN 'DetailedElement' THEN t.table_rows * 4
            WHEN 'TypeDouble' THEN t.table_rows * 4
            WHEN 'ResultOnTime' THEN t.table_rows * 4
            WHEN 'TypeCharacter' THEN t.table_rows * 4
            WHEN 'TypeDefinition' THEN t.table_rows * 8
            WHEN 'TypeNode' THEN t.table_rows * 4
            WHEN 'TypeInt' THEN t.table_rows * 4
            WHEN 'TypeBinary' THEN t.table_rows * 4
            WHEN 'TypeExternal' THEN t.table_rows * 8
            WHEN 'TypeString' THEN t.table_rows * 4
            WHEN 'SummaryChildElement' THEN t.table_rows * 12
            WHEN 'TypeChildOf' THEN t.table_rows * 8
            WHEN 'DetailedDuration' THEN t.table_rows * 4
            WHEN 'DetailedElementChild' THEN t.table_rows * 12
            WHEN 'DetailedTypeBlob' THEN t.table_rows * 4
            WHEN 'DetailedTypeCharacter' THEN t.table_rows * 4
            WHEN 'DetailedTypeChildOf' THEN t.table_rows * 8
            WHEN 'NodePerScenario' THEN t.table_rows * 4
            WHEN 'DetailedTypeDefinition' THEN t.table_rows * 8
            WHEN 'TypeStringArray' THEN t.table_rows * 4
            WHEN 'DetailedTypeDouble' THEN t.table_rows * 4
            WHEN 'DetailedTypeInt' THEN t.table_rows * 4
            WHEN 'DetailedTypeLongBlob' THEN t.table_rows * 4
            WHEN 'DetailedTypeMediumBlob' THEN t.table_rows * 4
            WHEN 'DetailedTypeNode' THEN t.table_rows * 4
            WHEN 'DetailedTypeStep' THEN t.table_rows * 16
            WHEN 'DetailedTypeString' THEN t.table_rows * 4
            WHEN 'DetailedTypeTestcase' THEN t.table_rows * 8
            WHEN 'DetailedTypeTinyBlob' THEN t.table_rows * 4
            WHEN 'CustomDetailedAttribute' THEN t.table_rows * 16
            WHEN 'CustomSummaryAttribute' THEN t.table_rows * 4
            WHEN 'UTPIterationPerExecution' THEN t.table_rows * 12
            WHEN 'UTPScenarioExecution' THEN t.table_rows * 24
        END*2)/1024/1024 AS Estimated_Data_Storage_Grow_Mb,
        max(t.data_length)/1024/1024 + max(t.index_length)/1024/1024 Minimum_Free_
Space_Required_For_Upgrade_Mb

from information_schema.tables t
where  t.table_TYPE = 'BASE TABLE'
group by DB_Schema;
```

Maintenance Window Calculation

From the Previous script use the columns `Total_Rows and Maximum_Rows_In_Table` to have a brief estimation of time required:

- A Database with `Maximum_Rows_In_Table` close to 2000 million rows:
    - Between 10 to 12 hours in a xlarge rds instance for the database modifications.
    - Between 30 minutes to 1 hour for space allocation if needed.
    - Around 24 hours to reach normal performance after the upgrade. (System is fully usable during this period be expect slowness in report generation.)
- A database with less than 20 million rows in `Maximum_Rows_In_Table`:
    - Between 30 minutes to 1 hour in a xlarge rds instance for the database modifications.
    - Between 30 minutes to 1 hour for space allocation if needed (not expected to be needed).
    - Around 24 hours to reach normal performance after the upgrade. (System is fully usable during this period be expect minor slowness in report generation.)
- The time relationship is not proportional, it obeys more an exponential curve, expected to take 25% of the time for 1 billion rows.

Space allocation usually takes the same amount of time, in case of rds instances, but that number is impossible to be calculated. Provided numbers are based on several tests made on different size databases, however they cannot be guaranteed. On basic mysql servers the expected time may be reduced to 20% in most cases as seen during tests.

Server Preparation

Pre-allocate space for rds instance to ensure the Minimum space required for the upgrade process is available before running the script.

- Space Provisioned for RDS must be greater than `Total_Db_Allocated_Space_Mb` + `Minimum_Free_Space_Required_For_Upgrade_Mb` + `Estimated_Data_Storage_Grow_Mb`

DO NOT RELY ON AUTOGROW SPACE FEATURE: During tests, on large databases, the space allocated may not be sufficient to finish some table schema updates, and due to particular aws rds architecture, the next space allocation cannot be done within 6 hours, bringing the database in some cases to a corrupted state and not allowing to continue the upgrade process.

- Upgrade RDS instance temporary to xlarge/2xlarge just for the upgrade process, instance can be downgraded after finishing the process.

Upgrade Process

1. Stop AIQ Test Nodes.

2. Stop AIQ Controller.

3. Create a DB Snapshot/Backup before Starting the process

4. In your MySQL client open a new session as a root user.

5. Open and execute the MySQLInitial.sql script.

> Download the MySQLInitial.sql script from the AIQ 5.0 Database Upgrade topic.

If any error of failure encountered, review the cause and re-run the script. The script can be run multiple times without risk of damaging data or database.

6. Open and execute the MySQLAppvance_ViewsAndProcs.sql, script selecting the schema for aiq database.

> Download the MySQLAppvance_ViewsAndProcs.sql script from the AIQ 5.0 Database Upgrade topic.

If any error of failure encountered, review the cause and re-run the script. The script can be run multiple times without risk of damaging data or database.

7. Start AIQ controller.

8. Check database connection. See Results for instructions.

Test Designer User Upgrade

If upgrading, the existing Test Designer (proxy) must be fully stopped before beginning the new installation. You must also click No when prompted to “Keep the current settings”.

Steps

1. Navigate to Test Designer install directory (C:\TestDesigner or /Users/<UserName>/TestDesigner)

2. Navigate to the NetworkProxy directory.

3. Run (double-click) the shutdown.cmd /shutdown.sh for Mac.

4. Download and install the new Web Designer (proxy) from your AIQ Web Designer interface.

Chrome Policy Settings

If the either of the following situations apply to you, you must run a command to update Chrome Policy settings:

- You are using the Chrome System Profile

- You need to use Downloads

Run the appropriate command for your operating system:

- Linux: set_chrome_policy_for_system_profile_downloads_linux.sh

- Windows: set_chrome_policy_for_system_profile_downloads.cmd

- Mac OSX: set_chrome_policy_for_system_profile_downloads.sh

Run as Administrator and/or root verification is required to run this command.

Install Location of Designer Proxies

After upgrading you may notice that you have two separate Designer Proxies installed in different locations.

- The Designer Proxy for the new Web Designer implementation is installed at `C:\Users\<user>\WebDesigner`.
- The Designer Proxy for the pre-AIQ 5.0 implementation was installed at `C:\Users\<user>\TestDesigner`. This instance of the Designer Proxy is not used by AIQ 5.0,

Switching Between Database Servers

If you have both 4.X and 5.X database servers, and you want to use the same controller VM to use one or the other, use the following procedure to switch between the 4.10.X and 5.X database servers.

1. Uncheck the **Enable** checkbox in **Global Options** > **Admin Options** > **Results** tab.
2. Install the AppvanceIQ build in controller that you want to run. You can use the Lab Management upgrade process (see “How to Upgrade Using Lab Management” in Lab Management) or change the build files manually.
3. Re-enable the database configuration after editing the database URL with the endpoint that matches the installed AppvanceIQ build.
4. Click the **Test Connection** button.
5. If the connection succeeds click **Accept**.

Updating CICD Dashboard Code Location for AIQ 5.x.x

If your CICD dashboard was created with a CICD client downloaded from an Appvance S3 service., you must follow the update process to point your CICD `build.xml` file to your AIQ controller CICD client.

The Appvance S3 service that contains this code will be removed in the near future. If you have not performed the update for your 5.0.0 release your CICD dashboard will not work.

Update build.xml Process

1. Follow the appropriate steps for your operating system:

    - Linux / Mac OS

      Open a terminal and type the following commands. They will download a file `updateGetURL.jar` to the folder where you have your `build.xml` file. Do replace values between `<>` with your correct values.

      ```
      cd <folder-with-build.xml-file>
      ```

      ```
      wget http://<aiq controller domain>:8080/UI/container/updateGetURL.jar
      ```

      ```
      java -jar updateGetURL.jar <controller-domain-or-ip> build.xml
      ```

      Value for `<controller-domain-or-ip>` can be similar to any of the following:

      - `java -jar updateGetURL.jar ec2-52-52-192-202.us-west-1.compute.amazonaws.com build.xml`

      - `java -jar updateGetURL.jar localhost build.xml`

  - `java -jar updateGetURL.jar 52.52.192.202 build.xml`

  - `java -jar updateGetURL.jar aiq-server.mydomain.com build.xml`

- Windows

  a. Open a browser and download a file by copying the following URL into the browser:

  ```
  http://<controller-domain-or-ip>:8080/UI/-container/updateGetURL.jar
  ```

  b. Copy the downloaded `updateGetURL.jar` file to the same folder where your `build.xml` file is located.

  c. Execute the following command in a cmd prompt:

  ```
  java -jar updateGetURL.jar <controller-domain-or-ip> build.xml
  ```

  Value for `<controller-domain-or-ip>` can be similar to any of the following:

  - `java -jar updateGetURL.jar ec2-52-52-192-202.us-west-1.compute.amazonaws.com build.xml`

  - `java -jar updateGetURL.jar localhost build.xml`

- `java -jar updateGetURL.jar 52.52.192.202 build.xml`
- `java -jar updateGetURL.jar aiq-server.mydomain.com build.xml`

2. After executing the above commands, open your `build.xml` file in a text editor and verify that a line similar to the following exists:

   ```
   <get dest="${lib}/RestClient.jar" src-
   c="http://<controller-domain>:8080/UI/-
   container/AppvanceCIClientWithDeps.jar"/>
   ```

   The text `<controller-domain>` should be replaced by your controller domain/ip as shown in the `controller.json` file.

3. Push the changes to the `build.xml` file in your repository.

Troubleshooting

In case of any issues running the java `updateGetURL.jar`, you can manually edit the `build.xml` file to have the `get` line to have the `src` property use your controller to download the client such as :

```
<get dest="${lib}/RestClient.jar" src-
c="http://<controller-domain>:8080/UI/-
container/AppvanceCIClientWithDeps.jar"/>
```

General Notes

Test Designer Best Practices

Running other applications and browser tabs during recording and playback can dramatically impact Test Designer’s performance.

It is the best practice to close as many other applications and browser tabs as possible.

See Best practices to improve AIQ performance for more information.

Important Configuration for Safari Users

To use Safari with Test Designer please follow the steps defined in Apple Safari - Mac Configuration.

Known Issue with Google Chrome

A known compatibility issue with Google Chrome for Windows requires users to frequently reload the Mobile Designer IDE while working on scripts. Appvance is investigating the issue and hopes to have a fix in an upcoming release.

Enterprise Managed Chrome Browser Policies

If you are using an Enterprise Managed Google Chrome browser, be sure to check that the policy settings are not blocking AIQ from functioning correctly.

UserData Profile, Proxy and PromptForDownloadLocation are examples of critical settings that need to be checked.

See Enterprise Managed Chrome Policies for more information.

Documentation Updates

The enhancements and changes in AIQ release 5.0.0 are outlined in later sections of this document. Updated documentation can be viewed at AIQ Online

Documentation. Documentation for AIQ 5.0.0 is not updated in the Customer Portal space on the Confluence wiki.

AI 3.0 in AIQ

AIQ release 5.0.0 contains a comprehensive update to the AIQ AI engine that results in enhanced performance, reporting, and bug-finding accuracy.

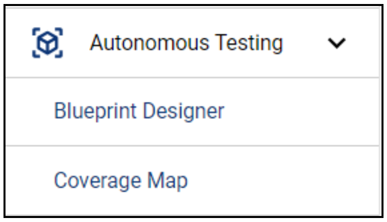
Some highlights of the improvements include:

- Optimized performance means that your blueprints will run faster.
- Calculated page states are more accurate and reliable when executing custom actions on specific pages.
- User interface has been re-factored to make it more intuitive and accessible to new users.
- You can data drive your Blueprint by adding data sets prior, during, and after Blueprint execution. You do not have to reset or restart the Blueprint if you forgot to add a data set when you added an execution.
- Existing blueprints that were created in older versions of AIQ will automatically be upgraded to the new format when you open them for the first time in AIQ 5.0. See "Upgrading Blueprints" on page 36 for more information.
- Blueprint Designer is the new AIQ 5.x Blueprinting user interface. Unlike the updated Web Designer interface, there is no corresponding "classic" version of Blueprint Designer in AIQ 5.0.

> In AIQ 5.0.0 AI Blueprinting is only supported on Windows test node.

Autonomous Testing Options

This is the Autonomous Testing menu in AIQ release 5.0.0.



The following sections detail working with those options:

- Blueprint Designer - "Working with the Blueprint Designer" on page 35
- Coverage Map - "Exploring the Blueprint Coverage Map" on page 84

Changes from Previous Releases

- Mobile Blueprinting is not enabled in the AIQ release 5.0.0. This function is being enhanced and will be reintroduced in a future release of AIQ.
- The Blueprint Template Editor has been removed as part of the blueprint enhancements.
- The Compare Blueprints option is not enabled in AIQ release 5.0.0. This function is being enhanced and will be reintroduced in a future release of AIQ.
- The Generate Scripts option is not enabled in AIQ release 5.0.0. This function is being enhanced and will be reintroduced in a future release of AIQ. However, you can now generate a DS script from the Coverage map, or download all actions in the Blueprint as a DS script.

A dedicated training course on Blueprinting is available as part of Appvance's training offerings. See the Appvance Training page for more information.

Resource Considerations

Performance of the blueprinting process is highly dependent on the resources available to your test node as well as configuration choices that you make. Here are some general guidelines.

Browser Considerations

You can run the AI browsers in headless or non-headless mode. Headless mode is recommended because of increased performance and lower resource usage. As a general rule of scale, if you are able to run five browsers in non-headless mode, you can likely run ten to fifteen headless browsers with the same performance and resource usage.

Non-Headless Browsers

For non-headless mode, the main limiting factor is memory (RAM) on your test node. If you have the following amount of memory available:

- 16 GB you will max out the memory with five non-headless browsers
- 32 GB you will max out the memory with ten non-headless browsers

Headless Browsers

For headless mode, the main limiting factor is your CPU speed and number of threads. You should be able to run ten to fifteen headless browsers with a base processor speed in the 2.4 to 2.6 GHz range. If you are able to increase that speed you will have better performance.

Number of Browsers vs Performance

If you are experiencing slow performance, try reducing the number of browsers and see if performance increases.

For example, if you run too many browsers and max out the machine's resources, the Blueprint will run slower. For example, if you run fifteen browsers, but the machine resources actually max out at ten browsers, you will get poor performance. This is because the machine is being overworked trying to run the fifteen browsers. If you actually ran ten browsers, you would see a significant performance increase because the machine isn't being overworked, even though resources may be maxed out.

Amazon Web Services Test Nodes

AWS servers use a base processor speed of 2.4 to 2.6 GHz. Since the AWS instance is not a fixed machine, you may be able to pay for credits to increase the processor speed.

Delay Between Requests

Delay Between Requests (ms) is an option that you specify for a Blueprint execution that determines how much time AIQ waits for the application to load before it reports a failure while navigating and performing actions. The longer the wait time, the longer it will give the application a chance to respond before reporting a failure.

A side effect of the higher a delay you specify is that you may be able to run more browsers. Delays make some browsers go to sleep, and because of this you can you share processor resources with other browser instances during that time.

For example, you find that ten browsers work best with a delay between requests of 0 ms. However, if the delay between requests is 20 ms then you can run fifteen browsers.

The ability to run more browsers if you specify a higher delay between requests is not a direct correlation. It is merely a side effect. Benchmarking and performance testing is encouraged. See "Basic Benchmarking" on the facing page for suggestions.

Types of Applications

The types of applications that you are testing will factor into how many browsers you can run.

- Static - If the web application is very static and very few elements, with a standard 8 core CPU with 32 GB of memory, you may be able to run up to twenty headless browsers.

- Dynamic - If the application is highly dynamic (moving elements, Shadow DOM, iframes, etc) and the scan type is tags you should do some basic benchmarking.

Basic Benchmarking

You can perform some initial benchmarking to see if you may need to change some execution parameters.

- Run a quick test blueprint. You could even test a mix of headless and non-headless browsers depending on your situation.

  For example if you want to test the difference between headless and non-headless browsers:

  - Run an execution with ten headless browsers.

  - Run a second execution with five non-headless browsers.

- Monitor how much memory and processor resources are utilized.

- Use this information to decide if you need to run a new blueprint with more or fewer browsers.

SmartTags and Validations

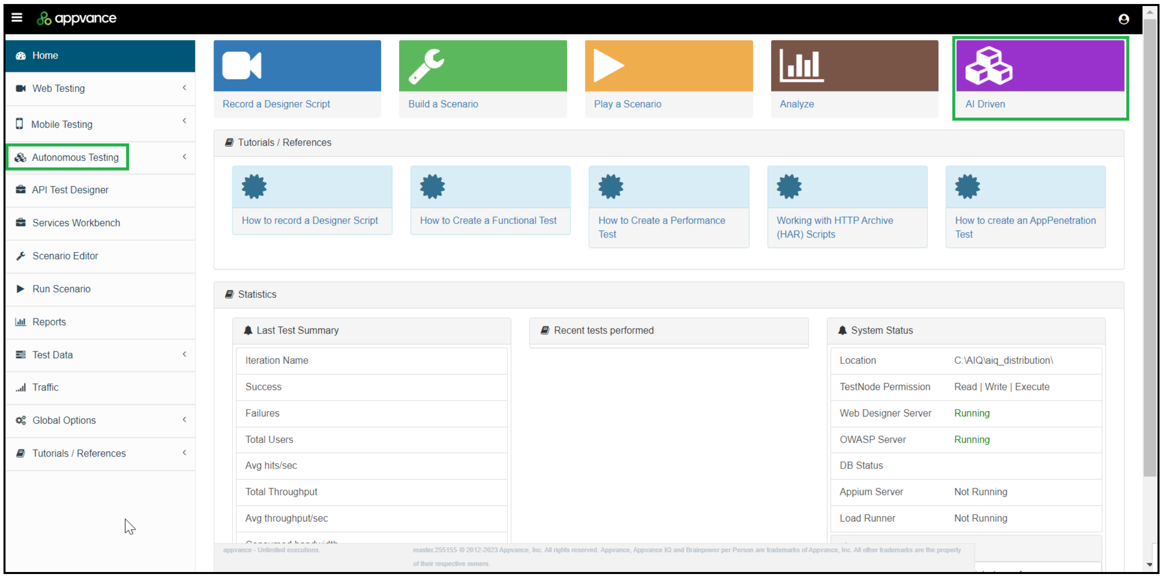
Running scans with large numbers of SmartTags and Validations will impact performance.

AIQ Limitations

Despite any external resources, how powerful your test node is, configuration settings, number of SmartTags and validations you use, AIQ has its own limits. The absolute maximum number of browsers that you can use is fifty.

However, the optimum number of browsers even if you have abundant computer resources is actually somewhere between thirty-five and fifty browsers.

Launching the Blueprint Designer



Last Test Summary

Iteration Name
Success
Failures
Total Users
Avg hits/sec
Total Throughput
Avg throughput/sec

System Status

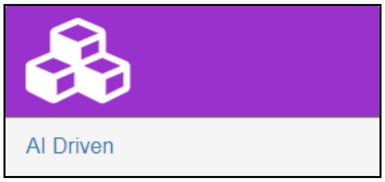
Location	C:\AIQ\aiq_distribution\
TestNode Permission	Read | Write | Execute
Web Designer Server	Running
OWASP Server	Running
DB Status	
Appium Server	Not Running
Load Runner	Not Running

appvance - Unlimited executions.

master.255155 © 2012-2023 Appvance, Inc. All rights reserved. Appvance, Appvance IQ and Brainpower per Person are trademarks of Appvance, Inc. All other trademarks are the property of their respective owners.

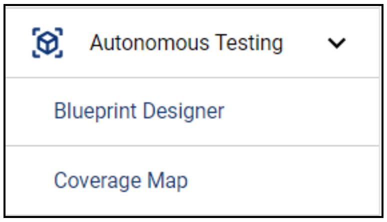
You can access the Blueprint Designer in two ways from the AIQ home screen.

- From the top line menu, click **AI Driven**.



- or -

- From the left menu, click **Autonomous Testing** to expand the menu,
- Select **Blueprint Designer**.



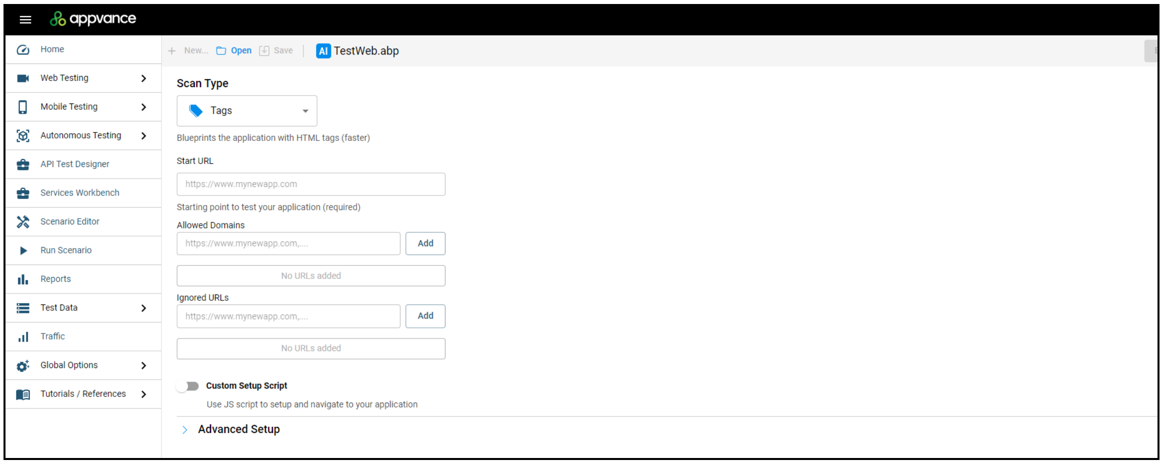
This is still the same way to access the Blueprint Designer as in recent releases.

Prerequisite. Before you can begin working with Blueprints, you must have a repository configured to store your Blueprint files. Storing files locally is not supported.

- See "Repository Considerations" on page 67 for information on what type of repository to use, especially if you have large file sizes.
- If you do not already have a linked repository, see "Linking a Repository" on page 67 for information.
- If you want to know more details on how the Blueprint artifacts are stored in the repository, see "Respository Details" on page 69.

Working with the Blueprint Designer

This is the new Blueprint Designer user interface that was introduced in the 5.0.0 release of AIQ.



TestWeb.abp

Scan Type

Tags

Blueprints the application with HTML tags (faster)

Start URL

https://www.mynewapp.com

Starting point to test your application (required)

Allowed Domains

https://www.mynewapp.com,....

No URLs added

Ignored URLs

https://www.mynewapp.com,....

No URLs added

Custom Setup Script

Use JS script to setup and navigate to your application

There are three ways in which you can begin your interaction with new Blueprint Designer:

- "Upgrading Blueprints" on the next page
- "Create a New Web Blueprint" on page 41
- "Create a New Mobile Blueprint" on page 52
- "Blueprint Templates" on page 62

> Prerequisite. Before you can begin working with Blueprints, you must have a repository configured to store your Blueprint files. Storing files locally is not supported.
> See "Linking a Repository" on page 67 for information, if you do not already have a linked repository.

Upgrading Blueprints

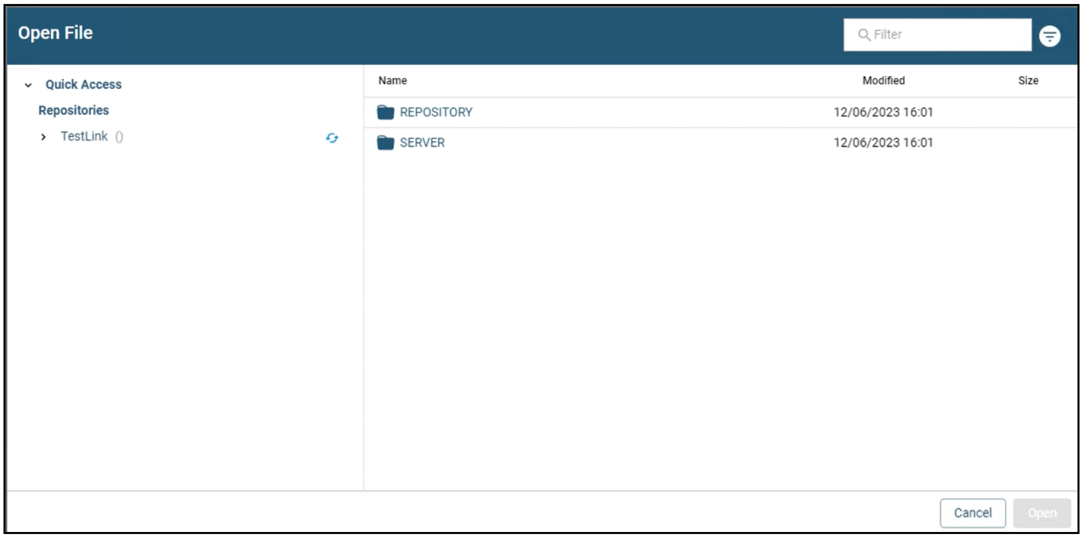
Open an existing blueprint to begin the process of upgrading it to the new AIQ release 5.0 format.

> This is a required process for each of your pre-AIQ 5.0 Blueprint that you want to use in AIQ 5.0.

1. Navigate to **Autonomous Testing** > **Blueprint Designer** or click **AI Driven** from the top line menu.

2. Click **Open**.

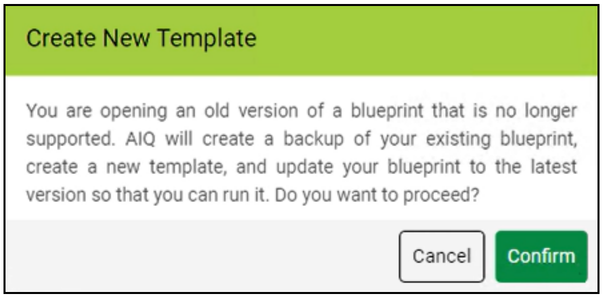
3. Navigate to the repository that contains your Blueprint files.



Name	Modified	Size
REPOSITORY	12/06/2023 16:01	
SERVER	12/06/2023 16:01	

4. Open a blueprint.

5. The **Create New Template** dialog box informs you that you must update your existing blueprint.



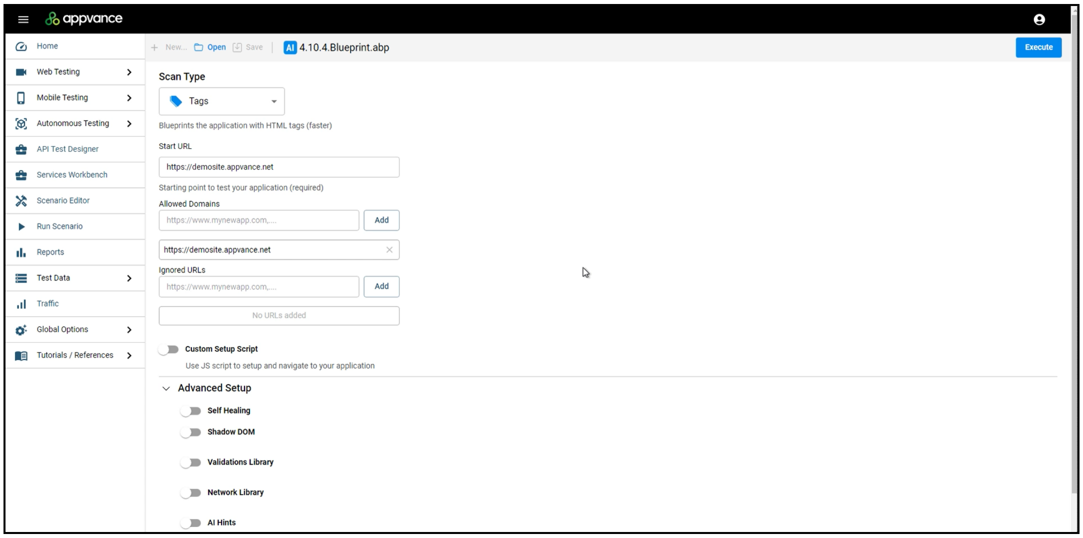
Create New Template

You are opening an old version of a blueprint that is no longer supported. AIQ will create a backup of your existing blueprint, create a new template, and update your blueprint to the latest version so that you can run it. Do you want to proceed?

> If you do not see this message it means that your blueprint file has already been upgraded to the AIA 5.x format.

6. Click **Confirm**.

7. Once your blueprint file loads, it means that your blueprint file has been upgraded. You can modify any settings as needed, or click **Execute** to run your blueprint.



AI 4.10.4.Blueprint.abp

Scan Type: Tags

Blueprints the application with HTML tags (faster)

Start URL: https://demosite.appvance.net

Starting point to test your application (required)

Allowed Domains: https://www.mynewapp.com,....

https://demosite.appvance.net

Ignored URLs: https://www.mynewapp.com,....

No URLs added

Custom Setup Script

Use JS script to setup and navigate to your application

Advanced Setup

- Self Healing
- Shadow DOM
- Validations Library
- Network Library
- AI Hints

> You should see that your existing SmartTags and validations are present in your new blueprint template. See the "Behind the Scenes" on the next page section for

information on what happens during the upgrade process.

Behind the Scenes

During the process of upgrading an existing pre-AIQ 5.x blueprint to AIQ 5.0, the following actions are performed by AIQ:

- Creates a backup of your existing pre-AIQ 5.x blueprint by renaming it to `<blueprint_name>-backup.abp`.
- Creates a blueprint template file with the name of your blueprint file `<blueprint_name>.abpt`.

  See "Blueprint Templates" on page 62 for information on Blueprint Templates.
- Creates a new folder with the name of your blueprint file `<blueprint_name>`.
- Gathers all the files linked to your original blueprint file (SmartTags, validations, etc) and makes copies of them. These files are stored in the newly created folder (`<blueprint_name>`) and linked to your newly created blueprint template.
- Opens the newly created blueprint template (`<blueprint_name>.abpt`) and creates a blueprint file with the same file name (`<blueprint_name>.abp`) as your original pre-AIQ 5.x blueprint.
- When you open a newly upgraded blueprint, your SmartTags and validations will automatically be part of the new blueprint template file.

- Templates will contain all custom actions that were present in the converted Blueprint. However, you will not see them as if they were newly created by you. This is true for all Blueprint templates, not just ones that were converted from pre-AIQ 5.0 versions.

  Example:

  - You create 20 custom actions on Blueprint and then export that Blueprint to a template.
  - When you create a new Blueprint from that template, the Blueprint will contain those custom actions and will execute them when the Blueprint is run.
  - However, if you look on the Page State menu where custom actions are displayed, you will not see them. These custom actions have become part of the template.
  - If you add new custom actions, you will only see the newly created ones shown on the Page State menu.

> See "Creating Custom Actions" on page 96 for more information on custom actions.
>
> See "Blueprint Templates" on page 62 for information on working with templates.

- If you need to make changes to any of the linked file (SmartTags, validations) after the upgrade, be sure to modify the files in the newly created folder with the name of your blueprint.

> ⚠ If you modify the original linked files in their original pre-conversion location (not the ones in the newly created

⚠ folder) for any reason, the changes will not be picked up by the new blueprint template.

.

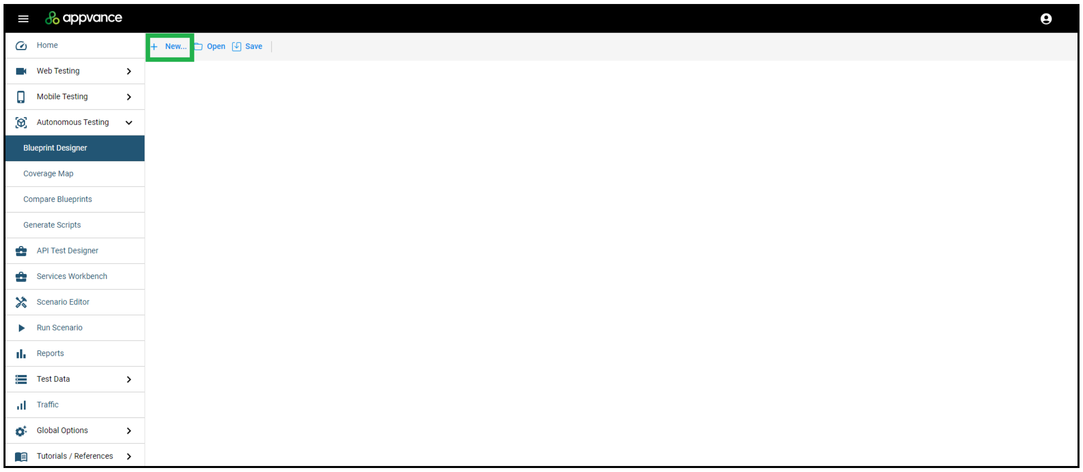
Create a New Web Blueprint

Here are the basic steps for creating and configuring a new Web Blueprint.

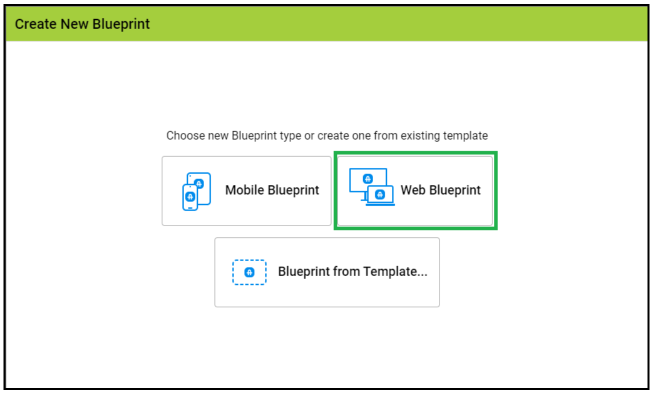
Creating a New Blueprint File

1. Open the Blueprint Designer. See "Launching the Blueprint Designer" on page 33 if you need more information on how to access the Blueprint Designer.

2. Click **New**.



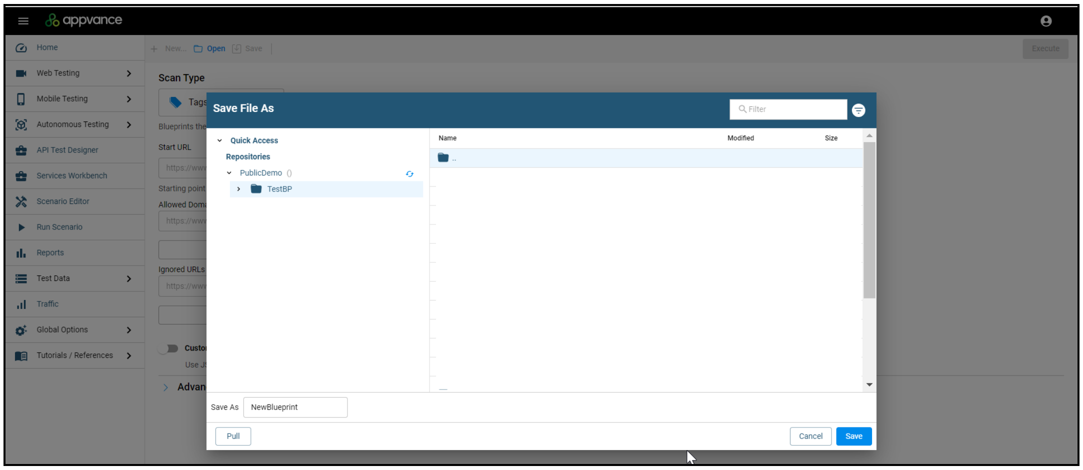
3. Click **Web Blueprint**.


Create New Blueprint

Choose new Blueprint type or create one from existing template

4. Select the appropriate repository in the Repository pane.

5. Name your file and click **Save**.



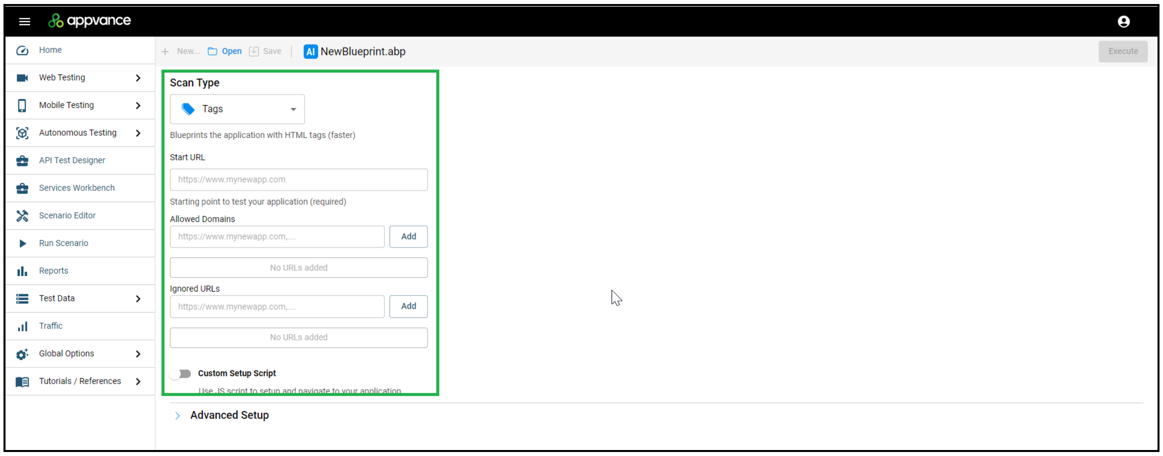
Save As: NewBlueprint

The file is not actually saved until you execute the Blueprint. This means that after you configure everything, if you reload the page or leave the page and come back before executing the Blueprint, all data and configuration settings will be lost.
Once the execution starts, the file will be auto-saved at five minute intervals.

6. You can now configure your Blueprint. See "Configuring the Web Blueprint" on the facing page and "Advanced Setup Options" on page 47.

7. Once the configuration is complete, click Execute. See "Add an Execution" on page 71 for information on that process.

Configuring the Web Blueprint



Scan Type

Tags

Blueprints the application with HTML tags (faster)

Start URL

https://www.mynewapp.com

Starting point to test your application (required)

Allowed Domains

https://www.mynewapp.com,...

Add

No URLs added

Ignored URLs

https://www.mynewapp.com,...

Add

No URLs added

Custom Setup Script

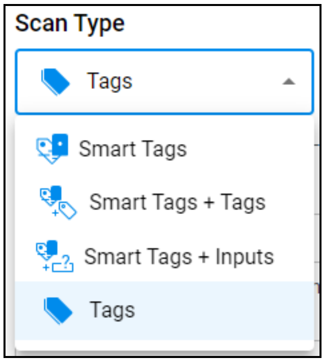
Use JS script to setup and navigate to your application

Advanced Setup

The following sections detail the basic configuration settings and options (highlighted section).

Blueprint configurations cannot be modified after the Blueprint has been executed. See "Modifying Blueprint Configuration Settings" on page 77 for more information.

Scan Type



Blueprints can be created with the following Scan Types.

Tags

Scans the application with HTML tags.

- Default option. This allows you to start basic Blueprinting of your application with only a basic URL as the only required configuration option. Once you have done this you can add more nuances to your scans by using one of the other scan types.
- Uses the `getAllTags()` function.
- Slowest but most complete scan type.

SmartTags

Blueprints the application with a SmartTags library that serves as meta-data. This helps with organization and with clicking certain elements that may have validations.

You can develop your SmartTags library as you continue to train the AI. You can name more buttons, control where the AI goes in the application, only want to click on certain pages once,

- Uses the `getSmartTags()` function which returns the HTML elements that match the rules defined in the SmartTags library as SmartTags.
- A file that corresponds to the SmartTags library is required. Click **Select File** under **SmartTag Library** to select the corresponding `.stags` file.
- This scan type is fast, but depends on the number of SmartTags and their coverage.
- Allows Blueprinting specific portions of the application.

For more information on SmartTags see:

- Understanding your Application with SmartTags
- SmartTag Workbench

SmartTags + Tags

Blueprints the application with a SmartTags library and HTML tags

- Uses the `getTagsSmartTags()` function.
- Combination of Tags and SmartTags.
- This scan type is very slow.

SmartTags + Inputs

Blueprints the application with a SmartTags library and uses INPUT, SELECT, BUTTON, SUBMIT and FORM tags.

- Uses the `getInputsSmartTags()` function.
- Reduces the number of SmartTags in the SmartTags library.
- Input elements that also correspond to a SmartTag is replaced by the SmartTag.
- This scan type is fast.

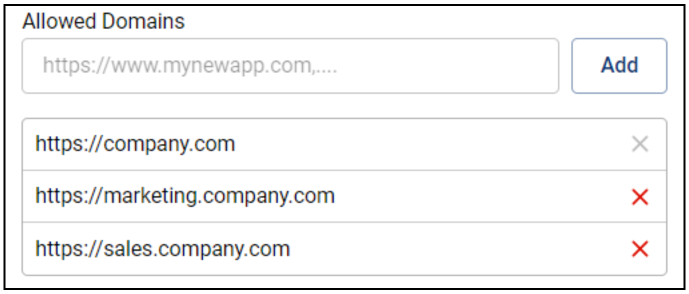
Start URL

Specify the base URL of the application as the starting point. The AI will navigate all pages with this base URL.

Allowed Domains

Determines access to URLs that might normally be ignored or blocked. This is useful in cases where your application contains URLs that are different from your base URL. One such case would be if certain pages in your application contain a prefix added to your base URL.

You can add multiple Allowed Domains to your configuration. You can either add them individually or all at once separated by comas. The Start URL is automatically added to the list of Allowed Domains.



Allowed Domains

https://www.mynewapp.com,....

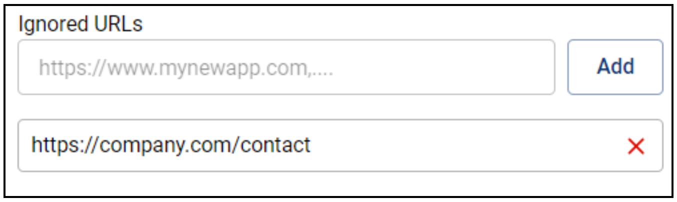
https://company.com

https://marketing.company.com

https://sales.company.com

Ignored URLs

Specify any URLs that should be ignored during the Blueprint execution. When the AI reaches this page it will ignore that URL and it won't be added to any of the coverage information or reports.



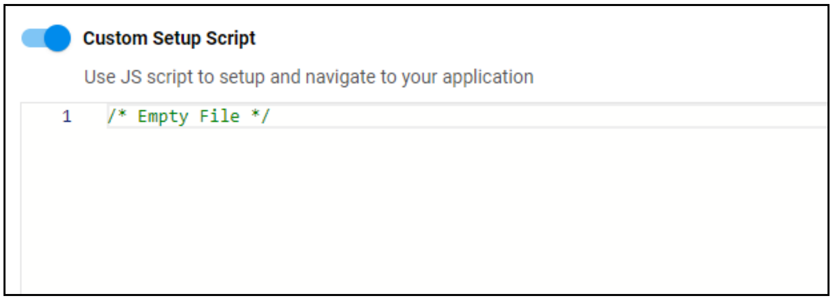
Ignored URLs

https://www.mynewapp.com,....

https://company.com/contact

Custom Setup Script

You can use a custom js script to setup and navigate to your application. If you enable this option, you can enter the js script directly.



Custom Setup Script

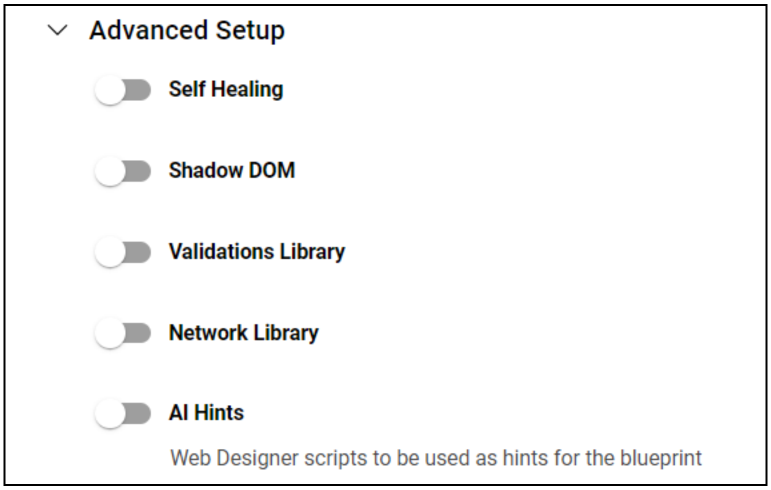
Use JS script to setup and navigate to your application

```
/* Empty File */
```

Advanced Setup Options

You can also set some more advanced options for your Blueprint. All the advanced setup options are disabled by default.

Expand the **Advance Setup** node to see the available options.



Advanced Setup

Self Healing

Shadow DOM

Validations Library

Network Library

AI Hints

Web Designer scripts to be used as hints for the blueprint

Self Healing

When working with SmartTag-based application Blueprinting, enabling self-healing and with multiple iterations of Blueprint, this has the ability to change the importance of the rules defined in the SmartTags.

This will change the importance of the rules defined (when there are multiple rules defined) depending on how the app is changing and the importance will be increased or decreased, and when the importance goes below 10%, the rule would be removed from the SmartTags and if there are any new attributes defined for the accessors, self-healing will have the ability to add new rules as well to the defined SmartTags.

Shadow DOM

If the application you are testing was built using Shadow DOM, such as Salesforce, you will need to enable this option to let the Blueprint know that this is a Shadow DOM-based application. This will allow the AI to see hidden accessors.This is disabled by default as it will not be needed by most applications.

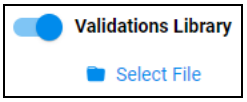
Situations where you might consider enabling this option:

- The execution stops almost immediately. Enable this option and start a new execution.
- You know that the application you are testing contains hidden accessors.

Validations Library

Adding a Validations Library allows the Blueprint to validate the application based on business logic definitions and decisions.

Once you enable the Validations Library option, the Select File link is shown. Use this to browse to the appropriate validation library file (`.valid` file extension).

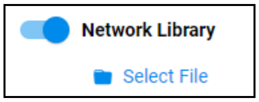


> Validation library files are created in the Validation Workbench. See Validation Workbench for more information.

Network Library

Network Library files contain validations can be extracted when they match the defined URLs. which will be in a specific format like JSON, HTML, HTTP, MD5, PDF, etc., from the response to validate it or to create variables for future usage in the scripting.

Once you enable the Network Library option, the Select File link is shown. Use this to browse to the appropriate network library file (`.netx` file extension).

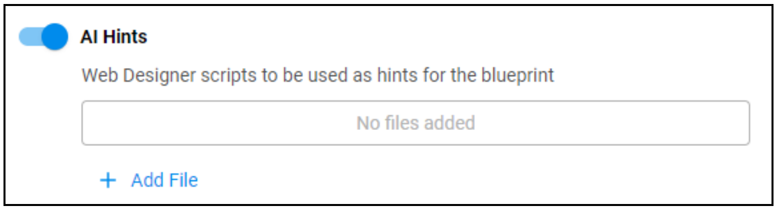


Network library files are created in the Network Workbench.
See Network Workbench for more information.

AI Hints

Enabling **AI Hints** allows you to use created Web Designer scripts as hints for the Blueprint.

Once you enable the **AI Hints** option, the Select File link is shown. You can upload multiple AI hints.



For example, if you already have a login script created using Web Designer, you can make use of that login script in your Blueprints as an AI Hint. This means that whenever the AI navigates to that page, it automatically uses the provided script as an input and logs into the application to perform further Blueprinting.

If you are data driving some variables in the script you are providing as a hint, you will need to add that data set before you can execute your Blueprint. See "Add an Execution" on page 71 for more information on adding a data set.

AI Hints used for blueprints must be simple scripts. For example, validations, `if`, and `while` statements are not supported at this time. Examples of supported items include clicks and setting values. You may need to simplify your scripts for them to work as AI Hints for your blueprints. This functionality will be enhanced in future releases of AIQ.

Technical Details

AI hints allow you to guarantee the coverage of your application by the Blueprint. This means that the Blueprint will scan those pages and then continue to branch out from there.

AI hints will start new threads even during script execution. At the beginning of the Blueprint, one thread will be started for each test script. For example, if five test scripts are provided, five threads will be started at the beginning.

The general process for a thread is:

- AI starts a thread.
- AI performs an action which navigates to a page.
- Page is scanned and saved.
- Actions are created from the scanned elements and added to the queue.
- Thread runs the next action from the queue.

New threads are started because the processes that happen after adding a new Action to the Blueprint, such as saving the Visual Hint and getting all the elements can take time. In the case of getting all the elements, having to scroll up and down can sometimes lead to a failure to execute the next step of the test script if AIQ cannot find the element of the next step. For this reason, another thread is started to continue the test script from where the current thread left off, while the current thread executes the processes that may take a long time to complete. By doing this the Blueprint is able to execute the test script completely while also performing all processes needed to complete the Blueprint.

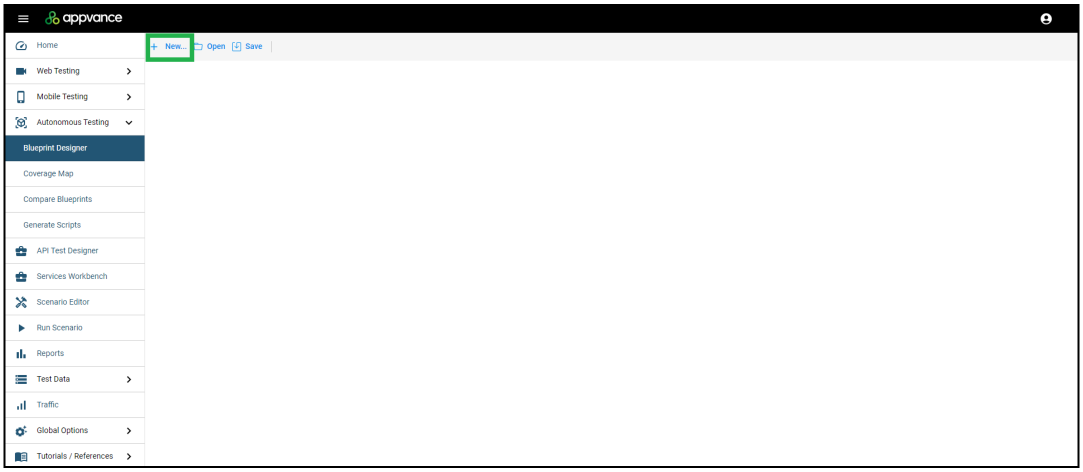
Create a New Mobile Blueprint

Here are the basic steps for creating and configuring a new Mobile Blueprint.

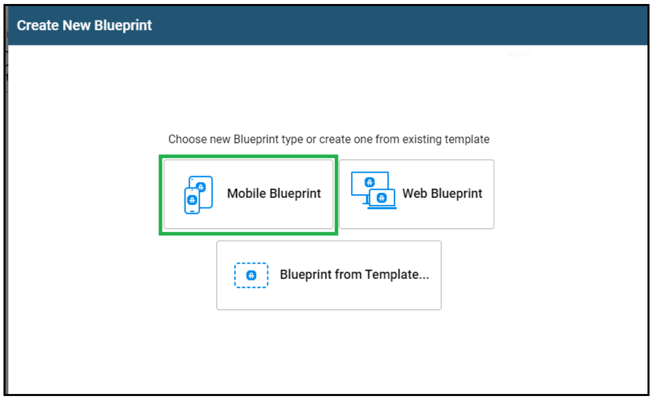
Creating a New Mobile Blueprint File

1. Open the Blueprint Designer. See "Launching the Blueprint Designer" on page 33 if you need more information on how to access the Blueprint Designer.

2. Click **New**.



3. Click **Mobile Blueprint**.

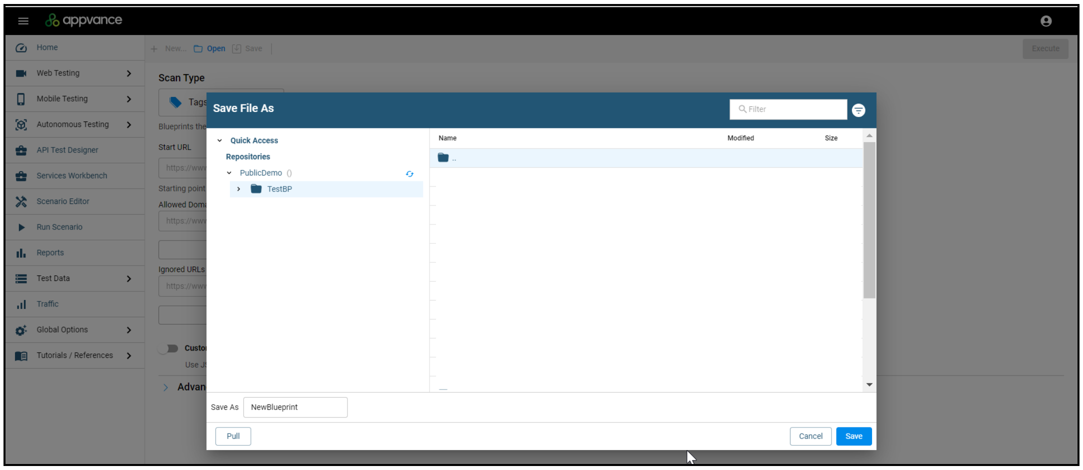

Create New Blueprint

Choose new Blueprint type or create one from existing template

4. Select the appropriate repository in the Repository pane.

5. Name your file.

6. Click **Save**. The file is saved with a `.mpb` file extension.



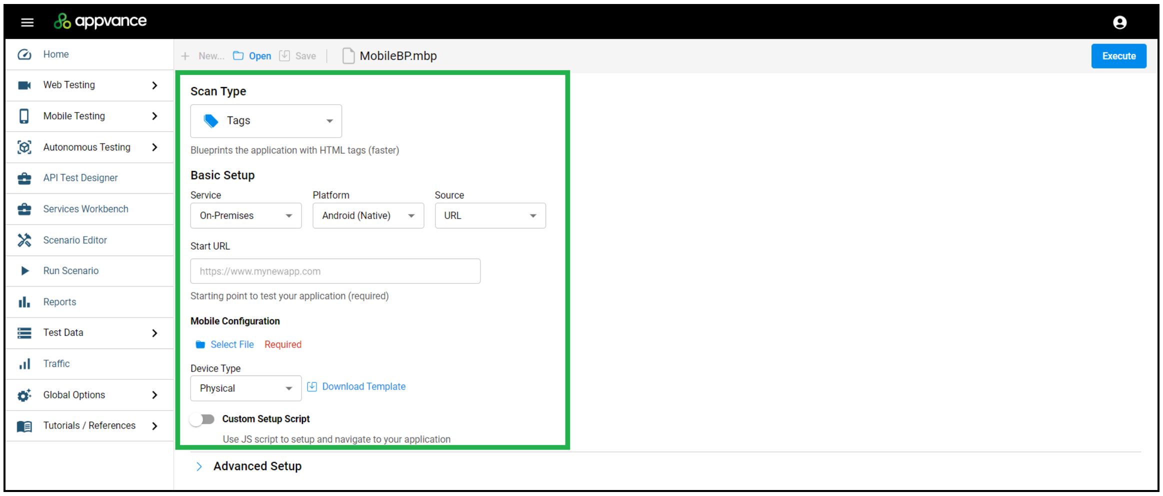
Save As NewBlueprint

The file is not actually saved until you execute the Blueprint. This means that after you configure everything, if you reload the page or leave the page and come back before executing the Blueprint, all data and configuration settings will be lost.
Once the execution starts, the file will be auto-saved at five minute intervals.

7. You can now configure your Blueprint. See "Configuring the Mobile Blueprint" on the next page and "Advanced Setup Options" on page 57.

8. Once the configuration is complete, click Execute. See "Add an Execution" on page 71 for information on that process.

Configuring the Mobile Blueprint



MobileBP.mbp

Scan Type
Tags
Blueprints the application with HTML tags (faster)

Basic Setup
Service: On-Premises
Platform: Android (Native)
Source: URL

Start URL
https://www.mynewapp.com
Starting point to test your application (required)

Mobile Configuration
Select File Required

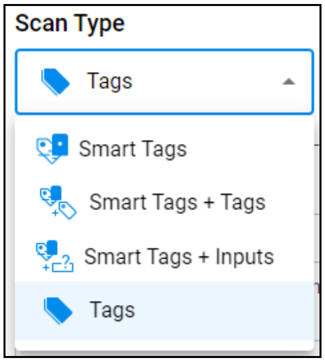
Device Type
Physical

Custom Setup Script
Use JS script to setup and navigate to your application

The following sections detail the basic configuration settings and options (highlighted section).

Blueprint configurations cannot be modified after the Blueprint has been executed. See "Modifying Blueprint Configuration Settings" on page 77 for more information.

Scan Type



Blueprints can be created with the following Scan Types.

Tags

Scans the application with HTML tags.

- Uses the `getAllTags()` function.

- Slowest but most complete scan type.

SmartTags

Blueprints the application with a SmartTags library that serves as meta-data.

- Uses the `getSmartTags()` function which returns the HTML elements that match the rules defined in the SmartTags library as SmartTags.

- A file that corresponds to the SmartTags library is required. Click **Select File** under **SmartTag Library** to select the corresponding `.stags` file.

- This scan type is fast, but depends on the number of SmartTags and their coverage.

- Allows Blueprinting specific portions of the application.

For more information on SmartTags see:

- Understanding your Application with SmartTags

- SmartTag Workbench

SmartTags + Tags

Blueprints the application with a SmartTags library and HTML tags

- Uses the `getTagsSmartTags()` function.

- Combination of Tags and SmartTags.

- This scan type is very slow.

SmartTags + Inputs

Blueprints the application with a SmartTags library and uses INPUT, SELECT, BUTTON, SUBMIT and FORM tags.

- Uses the `getInputsSmartTags()` function.
- Reduces the number of SmartTags in the SmartTags library.
- Input elements that also correspond to a SmartTag is replaced by the SmartTag.
- This scan type is fast.

Start URL

Specify the base URL of the application as the starting point.

Allowed Domains

Determines access to URLs that might normally be blocked.

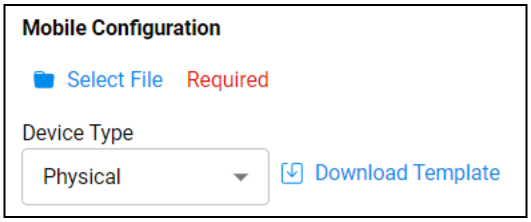
Ignored URLs

Specify any URLs that should be ignored during the Blueprint execution.

Mobile Configuration

A mobile configuration file contains all the necessary configuration information that is needed when testing a mobile application. A mobile configuration file is required for a Mobile Blueprint and should be created before you create a Mobile Blueprint.

> See Mobile Configuration for information on creating mobile configuration files.


Mobile Configuration

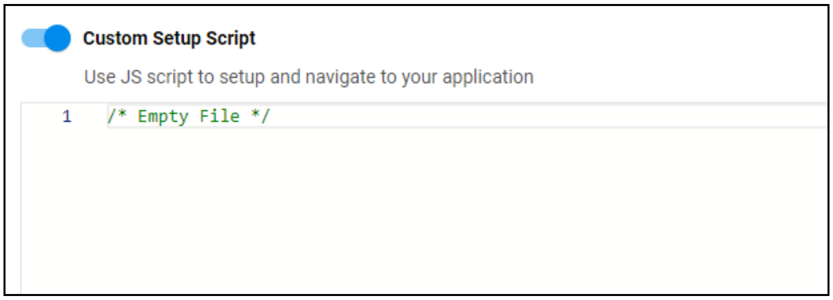
Device Type: Physical

1. Click Select File to select an existing mobile configuration file.

2. Select a **Device Type**. The options are a physical device or an emulator.

Custom Setup Script

You can use a custom js script to setup and navigate to your application. If you enable this option, you can enter the js script directly.


Custom Setup Script

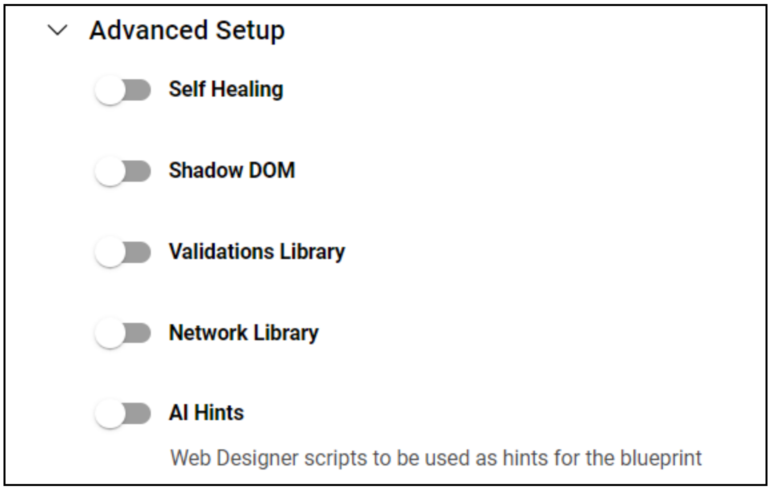
Use JS script to setup and navigate to your application

```
/* Empty File */
```

Advanced Setup Options

You can also set some more advanced options for your Blueprint. All the advanced setup options are disabled by default.

Expand the **Advance Setup** node to see the available options.



Advanced Setup

Self Healing

Shadow DOM

Validations Library

Network Library

AI Hints

Web Designer scripts to be used as hints for the blueprint

Self Healing

When working with SmartTag-based application Blueprinting, enabling self-healing and with multiple iterations of Blueprint, this has the ability to change the importance of the rules defined in the SmartTags.

This will change the importance of the rules defined (when there are multiple rules defined) depending on how the app is changing and the importance will be increased or decreased, and when the importance goes below 10%, the rule would be removed from the SmartTags and if there are any new attributes defined for the accessors, self-healing will have the ability to add new rules as well to the defined SmartTags.

Shadow DOM

If the application you are testing was built using Shadow DOM, such as Salesforce, you will need to enable this option to let the Blueprint know that this is a Shadow DOM-based application. This will allow the AI to see hidden accessors.This is disabled by default as it will not be needed by most applications.

Situations where you might consider enabling this option:

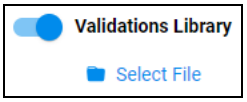
- The execution stops almost immediately. Enable this option and start a new execution.

- You know that the application you are testing contains hidden accessors.

Validations Library

Adding a Validations Library allows the Blueprint to validate the application based on business logic definitions and decisions.

Once you enable the Validations Library option, the Select File link is shown. Use this to browse to the appropriate validation library file (`.valid` file extension).



Validations Library

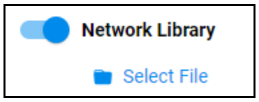
Select File

> Validation library files are created in the Validation Workbench. See Validation Workbench for more information.

Network Library

Network Library files contain validations can be extracted when they match the defined URLs. which will be in a specific format like JSON, HTML, HTTP, MD5, PDF, etc., from the response to validate it or to create variables for future usage in the scripting.

Once you enable the Network Library option, the Select File link is shown. Use this to browse to the appropriate network library file (`.netx` file extension).



Network Library

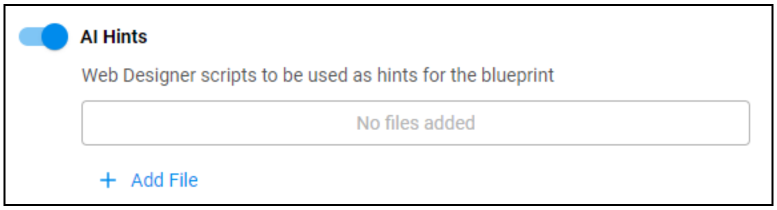
Select File

Network library files are created in the Network Workbench. See Network Workbench for more information.

AI Hints

Enabling **AI Hints** allows you to use created Mobile Designer scripts as hints for the Blueprint.

Once you enable the Network Library option, the Select File link is shown.



AI Hints

Web Designer scripts to be used as hints for the blueprint

No files added

For example, if you already have a login script created using Web Designer, you can make use of that login script in your Blueprints as an AI Hint. This means that whenever the AI navigates to that page, it automatically uses the provided script as an input and logs into the application to perform further Blueprinting.

If you are data driving some variables in the script you are providing as a hint, you will need to add that data set before you can execute your Blueprint. See "Add an Execution" on page 71 for more information on adding a data set.

AI Hints used for blueprints must be simple scripts. For example, validations, `if`, and `while` statements are not supported at this time. Examples of supported items include clicks and setting values. You may need to simplify your scripts for them to work as AI Hints for your blueprints. This functionality will be enhanced in future releases of AIQ.

Technical Details

AI hints allow you to guarantee the coverage of your application by the Blueprint. This means that the Blueprint will scan those pages and then continue to branch out from there.

AI hints will start new threads even during script execution. At the beginning of the Blueprint, one thread will be started for each test script. For example, if five test scripts are provided, five threads will be started at the beginning.

The general process for a thread is:

- AI starts a thread.
- AI performs an action which navigates to a page.
- Page is scanned and saved.
- Actions are created from the scanned elements and added to the queue.
- Thread runs the next action from the queue.

New threads are started because the processes that happen after adding a new Action to the Blueprint, such as saving the Visual Hint and getting all the elements can take time. In the case of getting all the elements, having to scroll up and down can sometimes lead to a failure to execute the next step of the test script if AIQ cannot find the element of the next step. For this reason, another thread is started to continue the test script from where the current thread left off, while the current thread executes the processes that may take a long time to complete. By doing this the Blueprint is able to execute the test script completely while also performing all processes needed to complete the Blueprint.

Blueprint Templates

Blueprints can be exported as Blueprint Templates. This allows you to create additional Blueprints more efficiently. You can create a base template and make iterative configuration changes as needed.

> You must also follow this process if you need to change the configuration of a Blueprint. Blueprint configurations cannot be modified after the Blueprint has been executed. See "Modifying Blueprint Configuration Settings" on page 77 for more information.

Exporting a Blueprint as a Blueprint Template

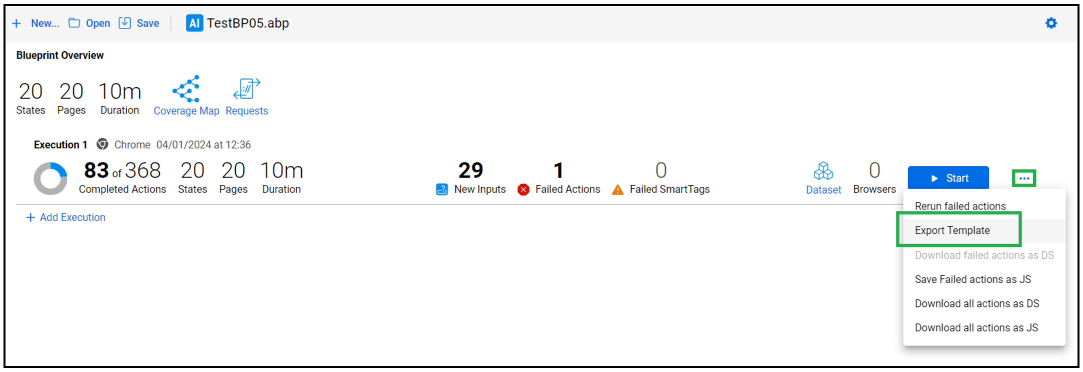
You can export an existing Blueprint to a Blueprint so you can reuse settings.

> See "Upgrading Blueprints" on page 36 for information on how custom actions are handled when creating a new blueprint from a template.

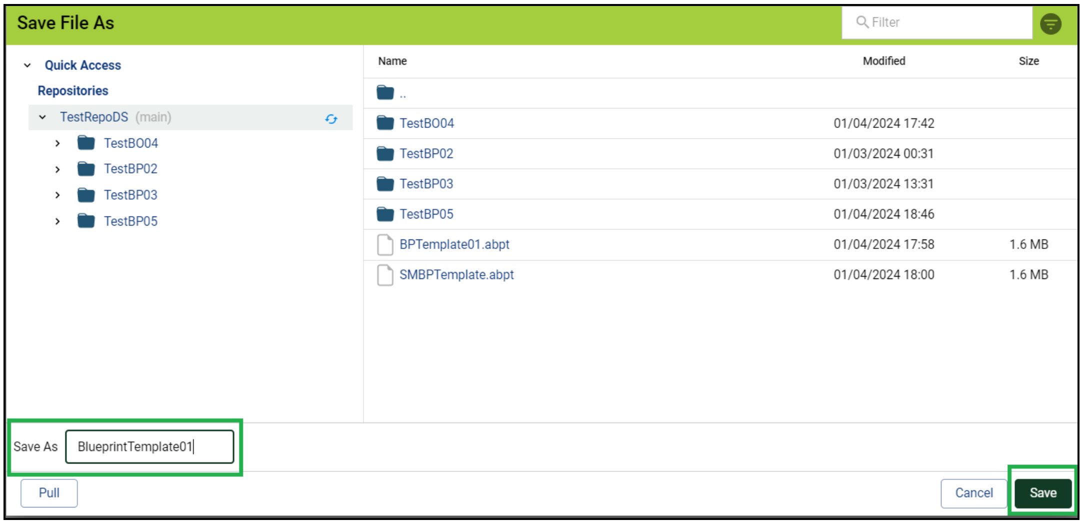
The Blueprint must not be running.

1. Click on the three dots (**...**) on the right of the Blueprint Overview page.

2. Select **Export Template**.



3. Enter a name for your template file and click **Save**.



Name	Modified	Size
..		
TestBO04	01/04/2024 17:42	
TestBP02	01/03/2024 00:31	
TestBP03	01/03/2024 13:31	
TestBP05	01/04/2024 18:46	
BPTemplate01.abpt	01/04/2024 17:58	1.6 MB
SMBPTemplate.abpt	01/04/2024 18:00	1.6 MB

Save As: BlueprintTemplate01

4. Your Blueprint Template is now available in your repository.

Create New Blueprint from Blueprint Template

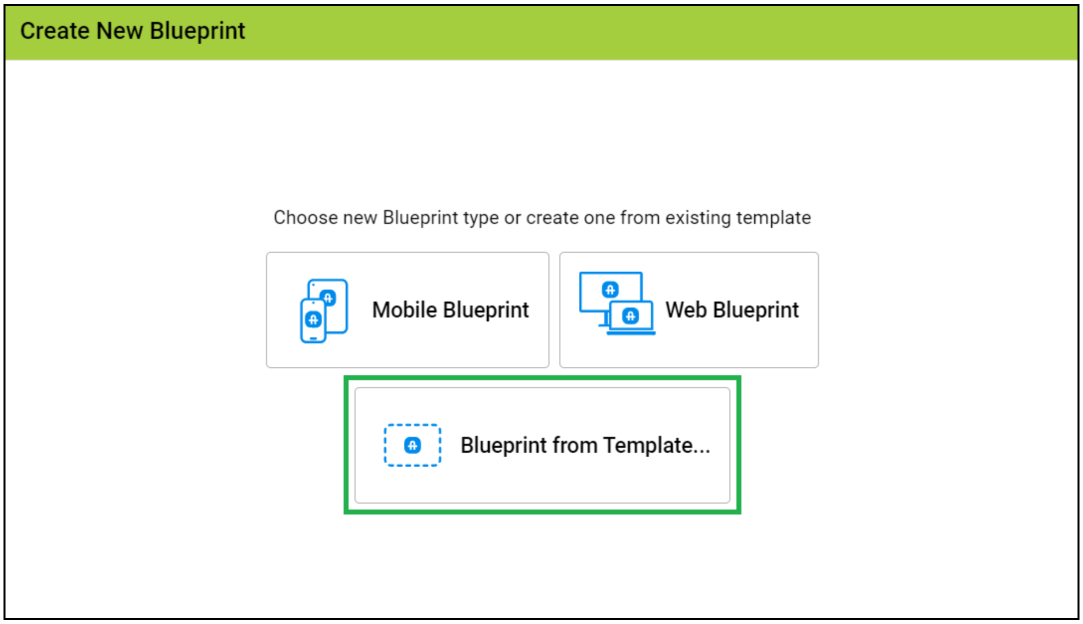
If you previously created a Blueprint Template from an existing Blueprint, you can use that template to create new Blueprints. See above for information on exporting a Blueprint to a template file.

> See "Upgrading Blueprints" on page 36 for information on how custom actions are handled when creating a new blueprint from a template.

1. Open the Blueprint Designer. See "Launching the Blueprint Designer" on page 33 if you need more information on how to access the Blueprint Designer.

2. Click **+New** from the top left of the Blueprint Overview page.

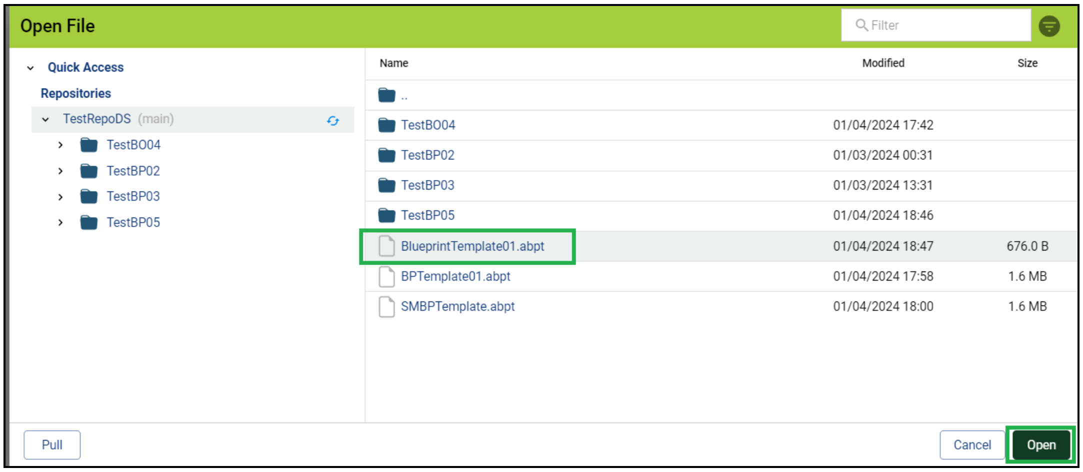
3. Select **Blueprint from Template**.



Create New Blueprint

Choose new Blueprint type or create one from existing template

4. Select the template file you created and click **Open**.



Open File

Quick Access

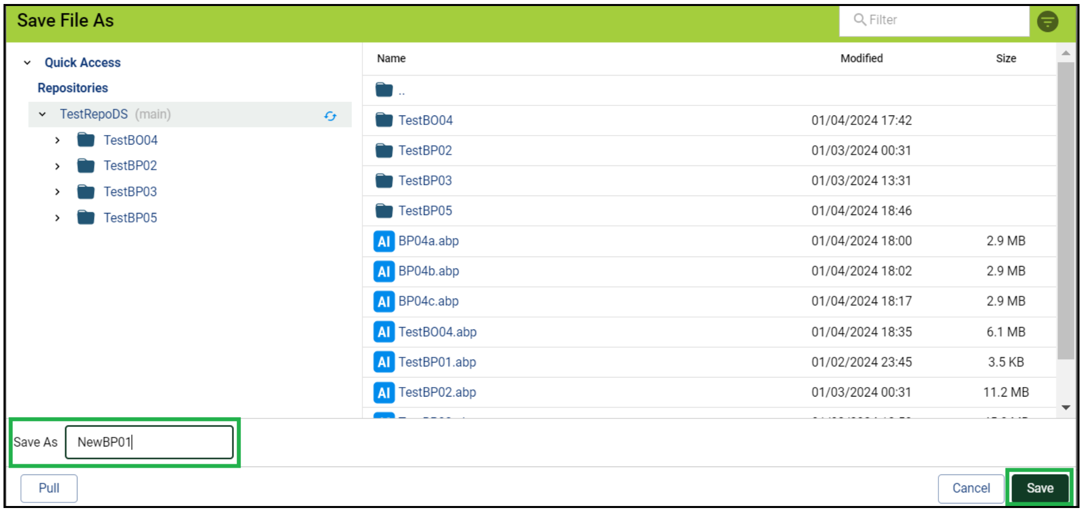
Repositories

TestRepoDS (main)

- TestBO04
- TestBP02
- TestBP03
- TestBP05

Name	Modified	Size
..		
TestBO04	01/04/2024 17:42	
TestBP02	01/03/2024 00:31	
TestBP03	01/03/2024 13:31	
TestBP05	01/04/2024 18:46	
BlueprintTemplate01.abpt	01/04/2024 18:47	676.0 B
BPTemplate01.abpt	01/04/2024 17:58	1.6 MB
SMBPTemplate.abpt	01/04/2024 18:00	1.6 MB

5. Enter a name for your new Blueprint file and click **Save**.



Save File As

Quick Access

Repositories

TestRepoDS (main)

- TestBO04
- TestBP02
- TestBP03
- TestBP05

Name	Modified	Size
..		
TestBO04	01/04/2024 17:42	
TestBP02	01/03/2024 00:31	
TestBP03	01/03/2024 13:31	
TestBP05	01/04/2024 18:46	
BP04a.abp	01/04/2024 18:00	2.9 MB
BP04b.abp	01/04/2024 18:02	2.9 MB
BP04c.abp	01/04/2024 18:17	2.9 MB
TestBO04.abp	01/04/2024 18:35	6.1 MB
TestBP01.abp	01/02/2024 23:45	3.5 KB
TestBP02.abp	01/03/2024 00:31	11.2 MB

Save As NewBP01

6. You can now configure your Blueprint. See "Configuring the Web Blueprint" on page 43 for more information.

The file is not actually saved until you execute the Blueprint. This means that after you configure everything, then reload the page or leave and come back before executing the Blueprint, all data and configuration settings will be gone.

Blueprint Sharing Considerations

If you are exporting blueprint templates and sharing them with other AIQ users, there are some issues to consider. When you import a blueprint into your repository, the file paths to the blueprint elements (SmartTags, validations, network library, and AI hints files) will be the absolute paths from the original repository location.

You will either need to browse to the file location and reselect the files, or make sure that your repository structure mirrors the original repository structure,

Linking a Repository

You must have a repository configured to store your Blueprint files. Storing files locally is not supported.

Supported repository types are:

- GIT
- SVN
- Amazon S3
- FTP
- SFTP

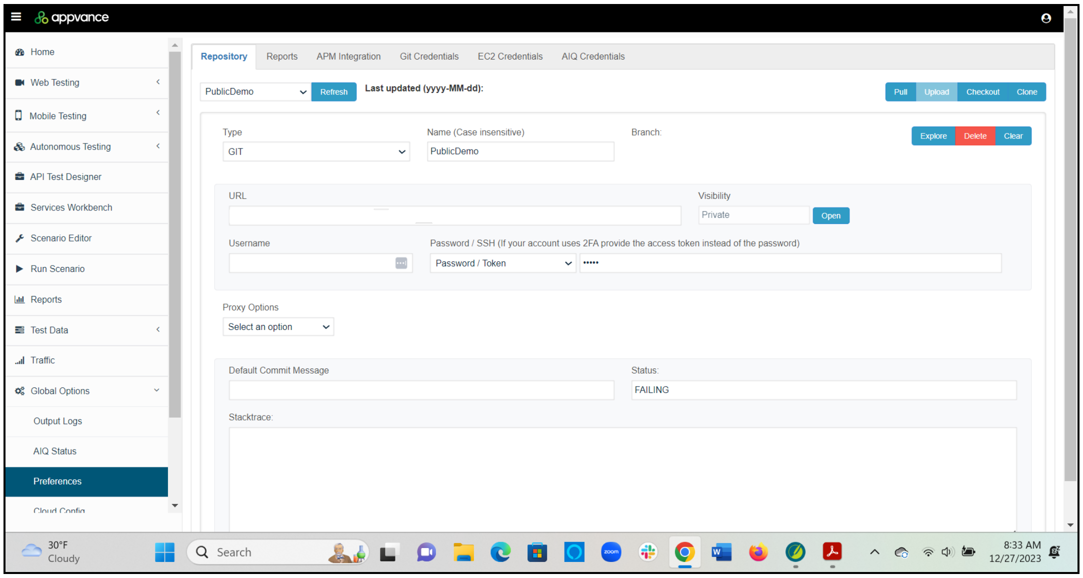
Repository Considerations

- If your Blueprint is extremely large and contains a large amount of images, you may eventually see performance issues if you are using GIT. The size limitation for GIT is around 100MB. Once you exceed 100MB, you may want to consider using an S3 repository. This is due to Git not being designed to handle large amounts of binary files efficiently. Images are resource-consuming elements, at the disk, memory, and CPU level, so using an S3 repository will yield the best performance. This is because The images will be saved with the script in your S3 repository which simplifies image management.
- If you are going to use Visual Accessors for Web Testing, you must use a SFTP, FTP, or S3 repository. Appvance does not recommend using Git to store your images if your are using the Visual Accessor functionality. Again, this is due to Git not being designed to handle large amounts of binary files efficiently.

Configuring a Repository

To configure a repository:

1. Navigate to **Global Options** > **Preferences** > **Repository** tab.



PublicDemo

Last updated (yyyy-MM-dd):

Type: GIT

Name (Case insensitive): PublicDemo

Branch:

URL:

Visibility: Private

Username:

Password / SSH (If your account uses 2FA provide the access token instead of the password): Password / Token •••••

Proxy Options: Select an option

Default Commit Message:

Status: FAILING

Stacktrace:

2. Configure your repository by defining a Type, Name, Username, URL and Password.

3. Click **Clone**.

4. A status message displays "Working" while the repository is added. Once the message is gone, you are ready to run your first blueprint.

See Repository Configuration in Appvance IQ for information on configuring various types of repositories.

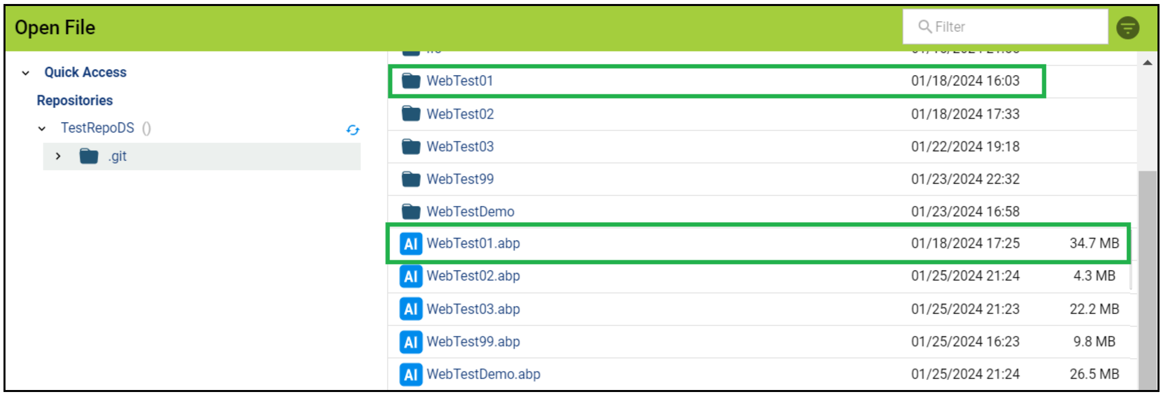
Respository Details

To improve the performance of saving to the repository the different elements of the Blueprint and the Blueprint execution are saved to the repository in two different ways.

For every Blueprint in your repository there is:

- A Blueprint file with the extension .abp.
  Example: The highlighted file WebTest01.abp in the image below.

- A folder with the same name as your Blueprint file.
  Example: The highlighted folder WebTest01 in the image below.



WebTest01	01/18/2024 16:03	
WebTest02	01/18/2024 17:33	
WebTest03	01/22/2024 19:18	
WebTest99	01/23/2024 22:32	
WebTestDemo	01/23/2024 16:58	
WebTest01.abp	01/18/2024 17:25	34.7 MB
WebTest02.abp	01/25/2024 21:24	4.3 MB
WebTest03.abp	01/25/2024 21:23	22.2 MB
WebTest99.abp	01/25/2024 16:23	9.8 MB
WebTestDemo.abp	01/25/2024 21:24	26.5 MB

The ,adp file contains:

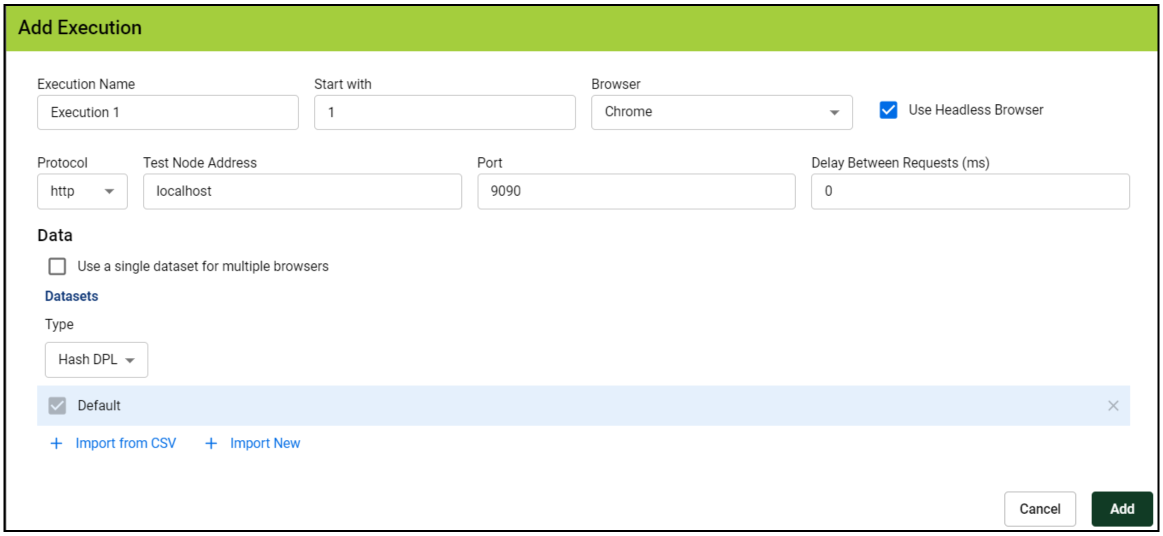
- Blueprint configuration settings
- Custom actions
- Elements
- Steps
- Element IDs

Folders contain:

- Screenshots and images
- Element bounds

Add an Execution

Before you can run the Blueprint, you must create at least on Blueprint Execution.



Add Execution

Execution Name: Execution 1

Start with: 1

Browser: Chrome

☑ Use Headless Browser

Protocol: http

Test Node Address: localhost

Port: 9090

Delay Between Requests (ms): 0

Data

□ Use a single dataset for multiple browsers

Datasets

Type: Hash DPL

☑ Default

1. Define the necessary parameters for your Blueprint Execution. See below for explanations.

2. Click **Add** once you have defined all the necessary parameters.

3. Your Blueprint will begin to run.

> In the previous implementation of Blueprinting (pre-AIQ 5.0), when the Blueprint ran you could see watch the browsers navigate your application in real time. Because of headless mode you will no longer see that happen in real time.
>
> See "Exploring Blueprint Results" on page 78 for information on what occurs when the Blueprint runs and how to explore and interpret the results.

Blueprint Execution Parameters

- **Execution Name** : Specify a name for the Blueprint execution.
- **Start with** : Specifies the number of AI bots / browsers to run. If you have a machine with sufficient resources, you can run up to fifteen AI bots / browsers concurrently.
-

> See "Resource Considerations" on page 29 for more information.

- **Browser** : Select the browser that will run the Blueprint. Chrome is the default browser.
- **Use Headless Browser** : Select whether to use a headless browser. Enabled by default.

> Headless mode is recommended because of increased performance and lower resource usage. As a general rule of scale, if you are able to run five browsers in non-headless mode, you can likely run ten to fifteen headless browsers with the same performance and resource usage.

- **Test Node** : Configure your test node.

> In AIQ 5.0.0 AI Blueprinting is only supported on Windows test node.

  - **Protocol** : Select `http` or `https`.
  - **Test Node Address** : The address of the test node.

        - Leave the default of `localhost` If you are running the Blueprint on the same controller as the test node.

        - Specify the IP address of the test node if you are running the Blueprint on the test node via the controller.

    - **Port** : Configure a proxy port. 9090 is the default.

    - **Delay Between Requests (ms) :** This determines the wait time between requests and should be adjusted based on the responsiveness of your deployment. See "Resource Considerations" on page 29 for more information.

        > The **Delay Between Requests (ms)** value was specified as Ajax Speed in the previous implementation of Blueprinting.

- **Data** and **Datasets** :You can use existing data bindings (custom actions and AI training that will be used in the next execution). You can add a Synthetic or Java Script file defining all the input values required.

    > This is used in conjunction with the **AI Hints** configuration option. See "Create a New Web Blueprint" on page 41 for more information on AI Hints.

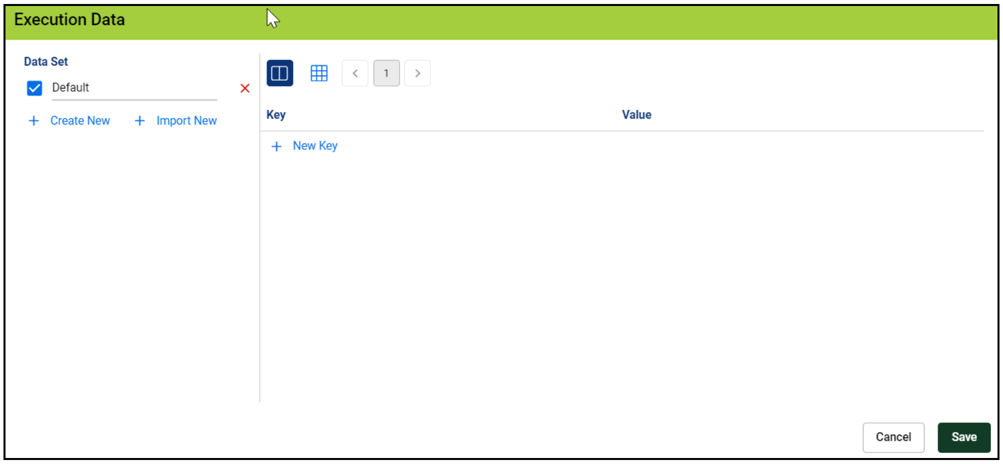
    A data set can be used to provide any inputs that are required for the application like the user names, password, search terms, etc, which could be used to when the Blueprint finds the appropriate inputs.

    Data sets can be added prior, during, and after Blueprint execution. You do not have to reset or restart the Blueprint if you forgot to add a data set when you added an execution.

The **Use a single data set for multiple browsers** option means that the number of concurrent browsers will be equal to the number of selected data sets for the execution. This is helpful when the application accepts only one use session at a time and when you will be using several browsers it is important to choose this option that way each credential is passed to a browser execution while doing the application Blueprinting.

> Data sets can also be linked to any Custom Actions that you create. See "Creating Custom Actions" on page 96 for more information.

1. Click the **Datasets** icon on the Blueprint Dashboard to open the **Execution Data** page.



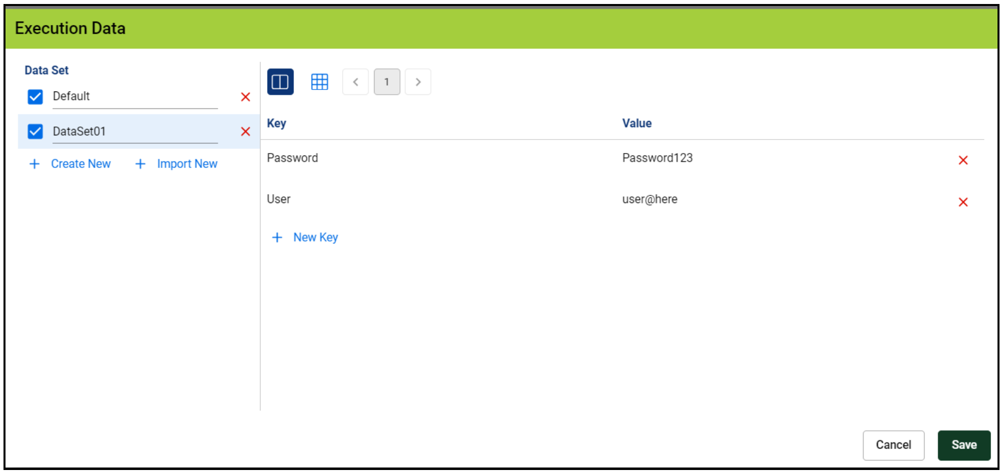
Execution Data

Data Set

☑ Default

2. Click **Import New** to select a preexisting file for your data set.
3. Click **Create New** to create a new data set.

4. Click New Key to add a new field and then define the **Key** and **Value**.



☑ Default ×

☑ DataSet01 ×

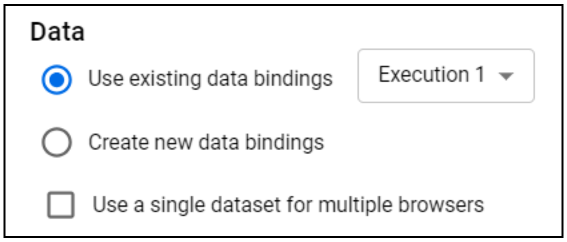
Key	Value	
Password	Password123	×
User	user@here	×

This was called DPL (Data Production Library) in the previous implementation of Blueprinting.

Adding Additional Executions

Adding additional executions of your Blueprint essentially means that you are running a new Blueprint with the existing configuration. If you need to change settings, you should create a new Blueprint.

When you add additional executions to a Blueprint, there are additional options that become available.



Data

● Use existing data bindings — Execution 1 ▾

○ Create new data bindings

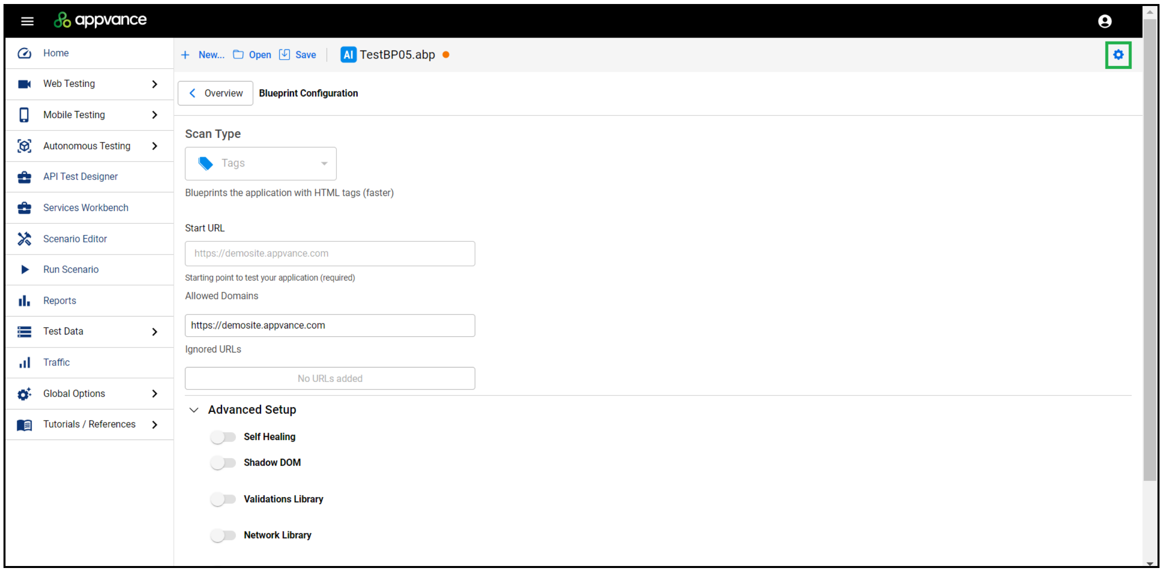
□ Use a single dataset for multiple browsers

- **Use existing data bindings** allows you to use the data sets from the selected execution, This does not apply to any custom actions. You will either have to re add the custom actions. You can also export the Blueprint as a template and execute a new blueprint. See "Upgrading Blueprints" on page 36 for information on how custom actions are handled when creating new Blueprint templates.

- **Create new data bindings** allows you to define a new data set for this execution.

Modifying Blueprint Configuration Settings

Click the Configuration icon (highlighted below) in the top menu line to view the configuration settings for a Blueprint. This is informational only. Blueprint configurations cannot be modified after the Blueprint has been executed.



Scan Type

Tags

Blueprints the application with HTML tags (faster)

Start URL

https://demosite.appvance.com

Starting point to test your application (required)

Allowed Domains

https://demosite.appvance.com

Ignored URLs

No URLs added

Advanced Setup

Self Healing

Shadow DOM

Validations Library

Network Library

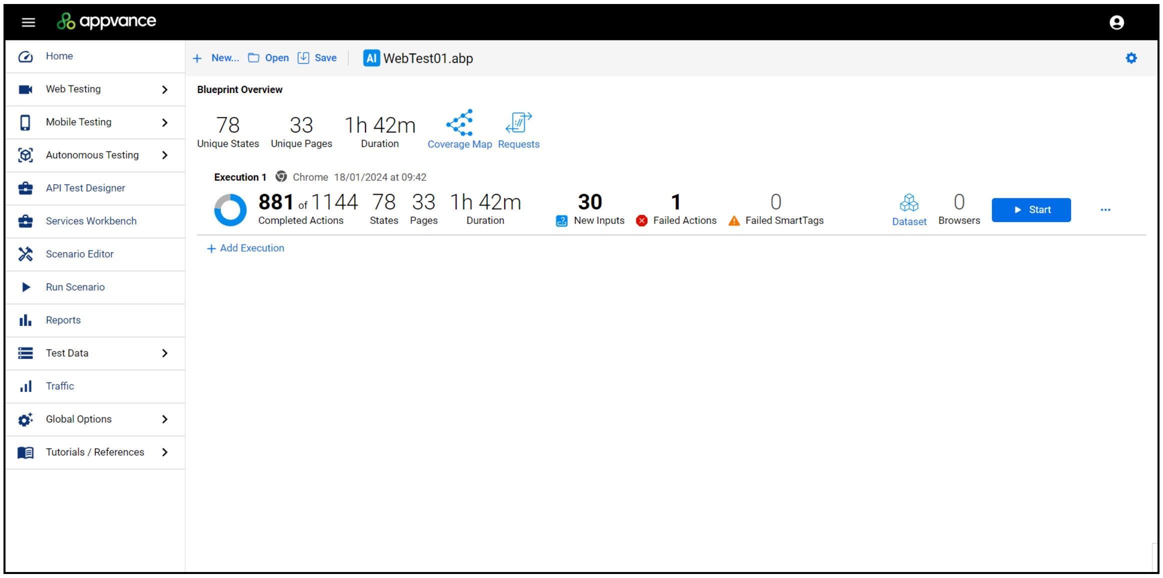
To change the configuration settings for a Blueprint, you must export the Blueprint as a template and create a new Blueprint from the template.

See "Blueprint Templates" on page 62 for information on the process of exporting a template and creating a new Blueprint from that template.

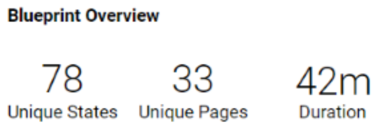
Exploring Blueprint Results

This is what the new Blueprint Overview, or Blueprint Dashboard looks like while a Blueprint is running or after a Blueprint has run. From here you can see the details of what the Blueprint found and interpret the results.

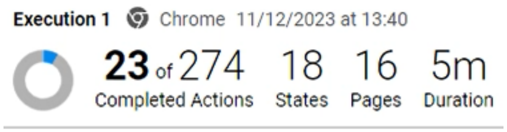
> In the previous implementation of Blueprinting (pre-AIQ 5.0), when the Blueprint ran you could see watch the browsers navigate your application in real time. Because of headless mode you will no longer see that happen in real time.

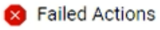
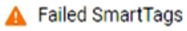


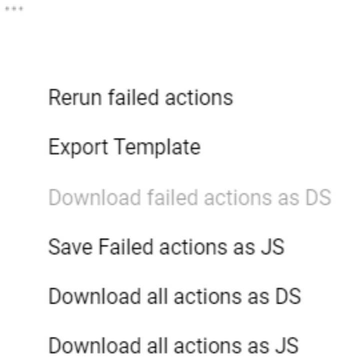
The following table details the icons and controls in the Blueprint Dashboard.

Icon / Control	Function
 Blueprint Overview 78 Unique States 33 Unique Pages 42m Duration	**Blueprint Overview** shows you a summary of all executions of the blueprint. • **Unique States** details the unique number of page states discovered for all

Icon / Control	Function
	executions. Shows the total application coverage. See "Unique States and Unique Pages" on page 88 in "Pages and States" on page 88 for more information. • **Unique Pages** details the number of unique pages discovered for all executions. Shows the total application coverage. See "Unique States and Unique Pages" on page 88 in "Pages and States" on page 88 for more information. • **Duration** is the total run time of your all blueprint executions.
Coverage Map	Click **Coverage Map** to launch the coverage map. The Coverage Map shows you the the multiple paths that the AI can take while interacting with the application being tested. See "Exploring the Blueprint Coverage

Icon / Control	Function
	Map" on page 84 for information on interacting and interpreting the Coverage Map.
Requests	Displays all the requests made by the Blueprint. See "Failed Requests and Failed Actions" on page 92.
SmartTags	This button is only visible if your Blueprint uses SmartTags. See "Exploring SmartTags" on page 100 for more information.
Validations	This button is only visible if your Blueprint uses Validations. See "Exploring Validations" on page 99 for more information.
 Execution 1 Chrome 11/12/2023 at 13:40 23 of 274 Completed Actions 18 States 16 Pages 5m Duration	**Execution** details shows you a summary of a particular blueprint execution. • **Completed Actions** details the actions that were performed by the blueprint. • **States** details the number of page states discovered. See "Pages and States" on page 88 for more information. • **Pages** details the number of pages discovered. See

Icon / Control	Function
	"Pages and States" on page 88 for more information. - **Duration** is the total run time of the particular blueprint execution. If you only have one blueprint execution, this duration will match the number in the **Blueprint Overview** section of the dashboard.
New Inputs	Click **New Inputs** to open the Pages information of the execution. See "Pages and States" on page 88 for more information. This replaces the **Inputs Found** button in the previous Blueprint user interface.
 Failed Actions	Click **Failed Actions** to see the details for the selected execution. From there you can see the pages with failed actions for the selected execution.
 Failed SmartTags	Click **Failed SmartTags** to see the failed SmartTags for the execution. See "Exploring SmartTags" on page 100 for

Icon / Control	Function
	more information.
Dataset	Allows you to add execution data from the Blueprint Dashboard. See the "Data and Datasets" information in "Add an Execution" on page 71 for more information.
10 Browsers	Displays the number of browsers that are running the Blueprint. This is set in the Add Execution page.
 ... Rerun failed actions Export Template Download failed actions as DS Save Failed actions as JS Download all actions as DS Download all actions as JS	• Rerun failed actions - This option allows you to retest the failed actions. This is useful if you forgot to add a dataset before running the Blueprint. • Export Template - Exports your Blueprint as a Blueprint Template. You can use that Template to create other Blueprints. See "Blueprint Templates" on page 62 for more information. • Download failed actions as DS • Save Failed actions as JS • Download all actions as DS • Download all actions as JS

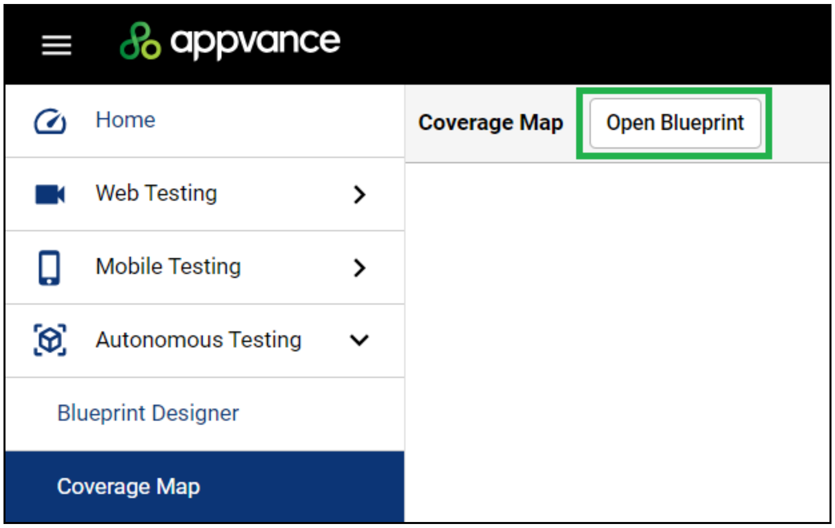
Icon / Control	Function
	See "Failed Requests and Failed Actions" on page 92 for information on how to interpret the various downloaded data.
+ Add Execution	Adds a Blueprint execution. At least one execution is required per Blueprint. See "Add an Execution" on page 71 for more information.
⚙	View your configuration settings. This is information only. Blueprint configurations cannot be modified after the Blueprint has been executed. See "Modifying Blueprint Configuration Settings" on page 77 for more information.

Exploring the Blueprint Coverage Map

The AIQ Blueprint Coverage Map shows you the the multiple paths that the AI can take while interacting with the application being tested.

A Coverage Map allows you to visualize the health of the application being tested and understand all the potential user flows. You can easily see areas of the application that might be plagued by errors and require rework. Over time, the historical progression of the Coverage Map will also help you see where there is degrading performance, unexpected defect density, or other changes.

- Click **Coverage Map** on the Blueprint Dashboard page to launch the coverage map for that Blueprint.
- To open a Coverage Map for a Blueprint that is not currently running, navigate to **Autonomous Testing** > **Coverage Map** > **Open Blueprint**.



You should be aware of the following difference from previous AIQ releases. When logged in as the same user, you can no

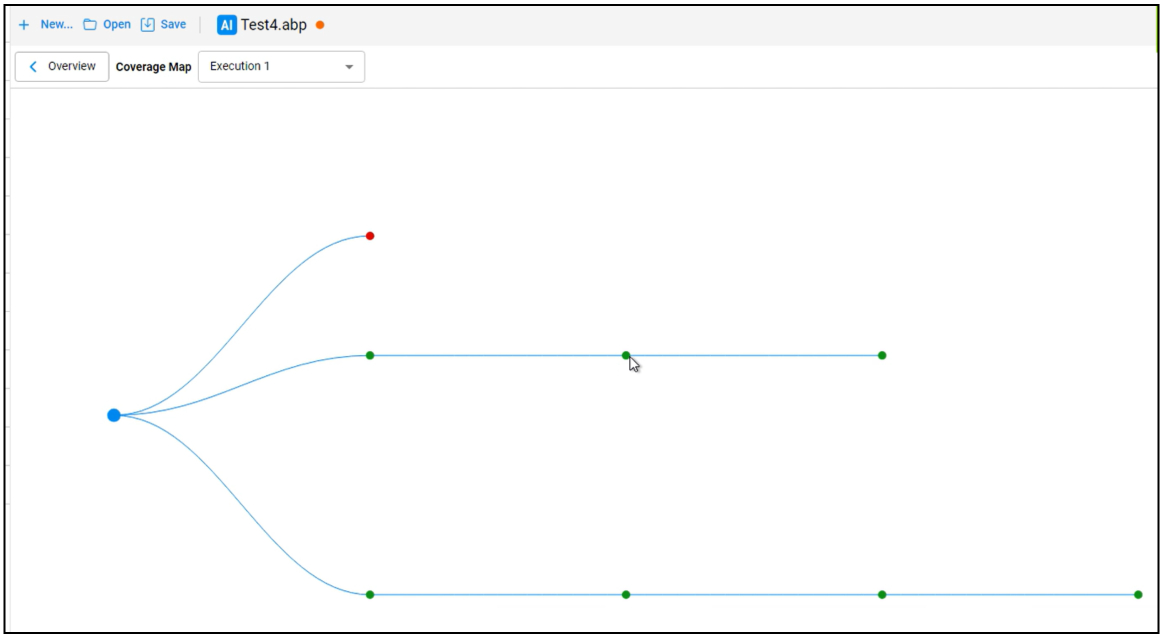
> longer select the Coverage Map of another Blueprint while a different Blueprint is running.
>
> For example, you have two Blueprints; Blueprint A and Blueprint B. Blueprint A is running and Blueprint B is not running. While Blueprint A is running, you can only access the Coverage Map for Blueprint A. You cannot access the Coverage Map for Blueprint while Blueprint A is running.

Interacting with the Coverage Map

You can to zoom in and zoom out to see more details including which actions and accessors are associated with a specific node. You can explore the Coverage Map in real time while the Blueprint is still running.

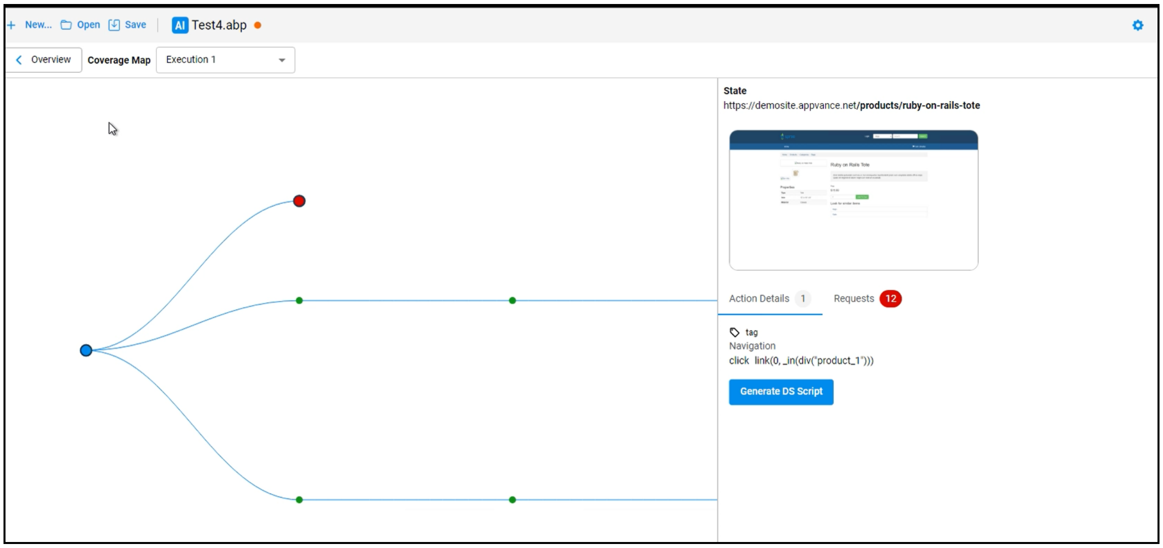
The colors of the various elements have the following meanings:

- White: Not executed
- Green: Executed with no errors
- Yellow: Executed with an non-critical error, such as a 500 error.
- Red: Executed with a critical error, such as a 404 error.
- Blue: The execution path from the start point



- You can select from the Coverage Maps for multiple executions (if you have them) in the Execution dropdown.

- Click a specific node to see the actions and accessors. From here you will also see their exact location within the application being tested.

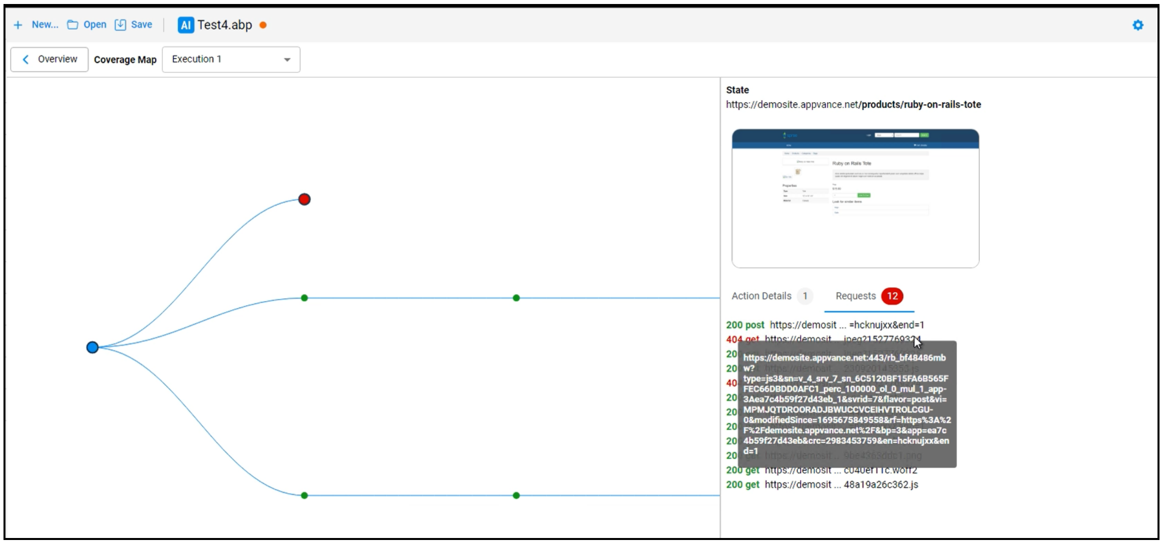


State
https://demosite.appvance.net/**products/ruby-on-rails-tote**

tag
Navigation
click link(0, _in(div("product_1")))

- Click **Generate DS Script** to generate a script that contains the steps that the AI took to reach this specific page.

- Click the **Request** tab in the detail window to see the associated requests.



State
https://demosite.appvance.net/**products/ruby-on-rails-tote**

200 post https://demosit ... =hcknujxx&end=1
404 get https://demosit ... jpeg21527769324

https://demosite.appvance.net:443/rb_bf48486mbw?type=js3&sn=v_4_srv_7_sn_6C5120BF15FA6B565FFEC66DBDD0AFC1_perc_100000_ol_0_mul_1_app-3Aea7c4b59f27d43eb_1&svrid=7&flavor=post&vi=MPMJQTDROORADJBWUCCVCEIHVTROLCGU-0&modifiedSince=1695675849558&rf=https%3A%2F%2Fdemosite.appvance.net%2F&bp=3&app=ea7c4b59f27d43eb&crc=2983453759&en=hcknujxx&end=1

200 get https://demosit ... c040ef11c.woff2
200 get https://demosit ... 48a19a26c362.js

Pages and States

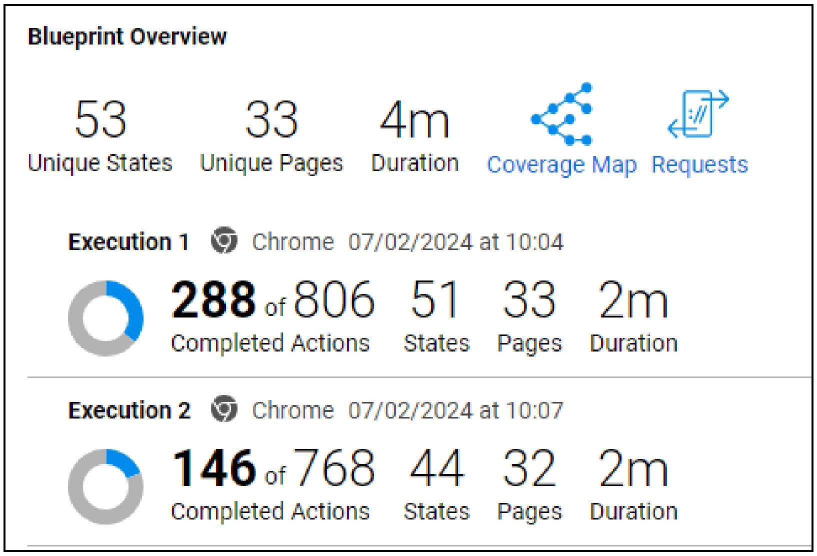
If you click the **Pages** or **States** link from the Blueprint Dashboard under an execution you will see a list of pages and page states. From here you can click on the pages of use the drop down menu to see the page states. You can use then explore by data driving fields, selecting buttons, etc.

Anything that you could do in the previous version of Blueprinting, you can do through the new Blueprint Dashboard.

Unique States and Unique Pages

The **Blueprint Overview** shows the total application coverage achieved by your Blueprint. This is reflected in the **Unique States** and **Unique Pages** totals. They are not the combination of those statistics across all of your executions, they represent the total application coverage.

For example:



Blueprint Overview

53 Unique States
33 Unique Pages
4m Duration

Execution 1 Chrome 07/02/2024 at 10:04

288 of 806 Completed Actions
51 States
33 Pages
2m Duration

Execution 2 Chrome 07/02/2024 at 10:07

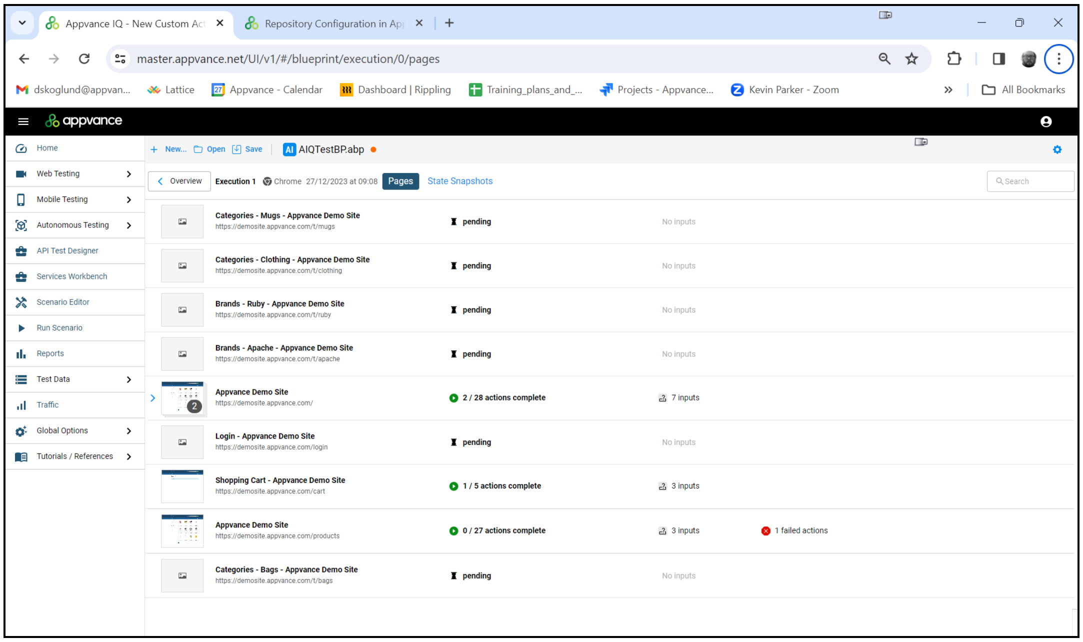
146 of 768 Completed Actions
44 States
32 Pages
2m Duration

- You run a single execution (Execution 1), and the result is 51 page states and 33 pages. The number of Unique Pages and Unique States will match the execution totals.

- You run a second execution (Execution 2) and the results are 44 page states and 32 pages. The second execution found two page states that first execution did not find, and did not find any pages that the first execution did not find. In this case, the overview totals would contain 53 Unique States and 33 Unique Pages.

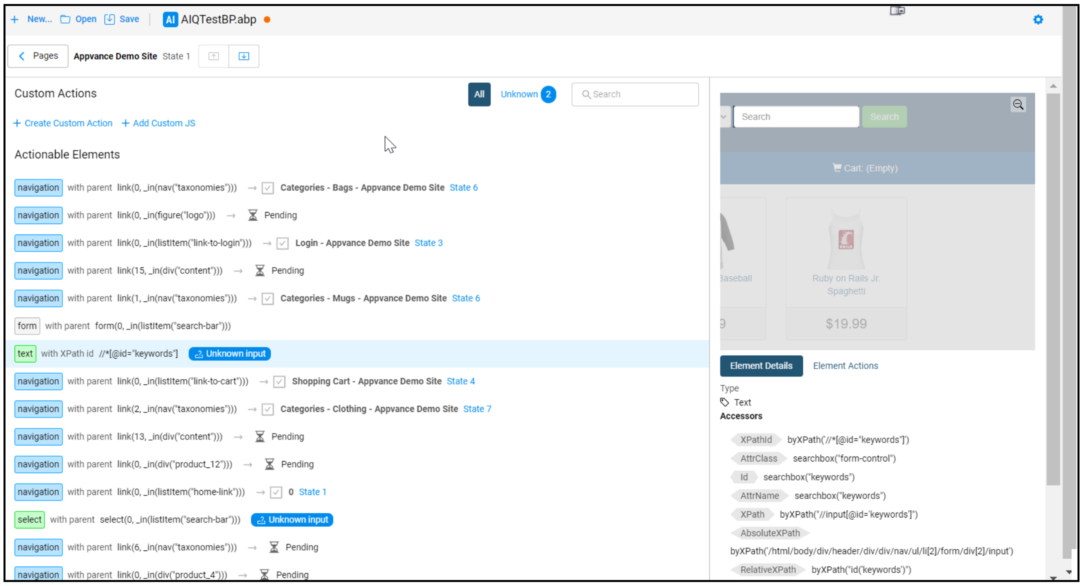
Example Page Screens

- If you click the **Pages** link from the Blueprint Dashboard under an execution you will open the main detail page for the execution. This is the default view. From there you can see a list of pages and page states. Pages are listed in the order in which the AI finds and scans pages. The first page will be the Starting URL you defined in the Blueprint configuration.



Categories - Mugs - Appvance Demo Site https://demosite.appvance.com/t/mugs	pending	No inputs	
Categories - Clothing - Appvance Demo Site https://demosite.appvance.com/t/clothing	pending	No inputs	
Brands - Ruby - Appvance Demo Site https://demosite.appvance.com/t/ruby	pending	No inputs	
Brands - Apache - Appvance Demo Site https://demosite.appvance.com/t/apache	pending	No inputs	
Appvance Demo Site https://demosite.appvance.com/	2 / 28 actions complete	7 inputs	
Login - Appvance Demo Site https://demosite.appvance.com/login	pending	No inputs	
Shopping Cart - Appvance Demo Site https://demosite.appvance.com/cart	1 / 5 actions complete	3 inputs	
Appvance Demo Site https://demosite.appvance.com/products	0 / 27 actions complete	3 inputs	1 failed actions
Categories - Bags - Appvance Demo Site https://demosite.appvance.com/t/bags	pending	No inputs	

- Click a specific page to explore more detail including seeing images of the pages. The images are saved automatically in your linked repository.



Actionable Elements

navigation with parent link(0, _in(nav("taxonomies"))) → ☑ Categories - Bags - Appvance Demo Site State 6

navigation with parent link(0, _in(figure("logo"))) → ⌛ Pending

navigation with parent link(0, _in(listItem("link-to-login"))) → ☑ Login - Appvance Demo Site State 3

navigation with parent link(15, _in(div("content"))) → ⌛ Pending

navigation with parent link(1, _in(nav("taxonomies"))) → ☑ Categories - Mugs - Appvance Demo Site State 6

form with parent form(0, _in(listItem("search-bar")))

text with XPath id //*[@id="keywords"] Unknown input

navigation with parent link(0, _in(listItem("link-to-cart"))) → ☑ Shopping Cart - Appvance Demo Site State 4

navigation with parent link(2, _in(nav("taxonomies"))) → ☑ Categories - Clothing - Appvance Demo Site State 7

navigation with parent link(13, _in(div("content"))) → ⌛ Pending

navigation with parent link(0, _in(div("product_12"))) → ⌛ Pending

navigation with parent link(0, _in(listItem("home-link"))) → ☑ 0 State 1

select with parent select(0, _in(listItem("search-bar"))) Unknown input

navigation with parent link(6, _in(nav("taxonomies"))) → ⌛ Pending

navigation with parent link(0, _in(div("product_4"))) → ⌛ Pending

Type

Text

Accessors

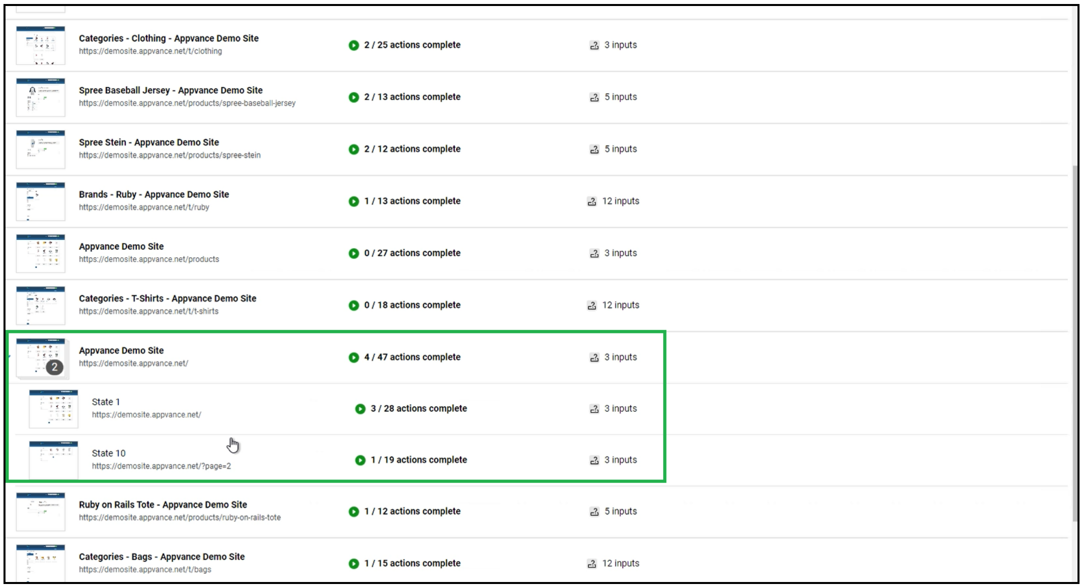
XPathId	byXPath('//*[@id="keywords"]')
AttrClass	searchbox("form-control")
Id	searchbox("keywords")
AttrName	searchbox("keywords")
XPath	byXPath("//input[@id='keywords']")
AbsoluteXPath	byXPath('/html/body/div/header/div/div/nav/ul/li[2]/form/div[2]/input')
RelativeXPath	byXPath("id('keywords')")

- To resolve the **Unknown input** errors, you can train the AI how to interact with those elements by creating a custom action or add a custom data set. See "Creating Custom Actions" on page 96 for more information.

- Using the **Search** bar you can search the Title and URL elements in the page.

Example State Screens

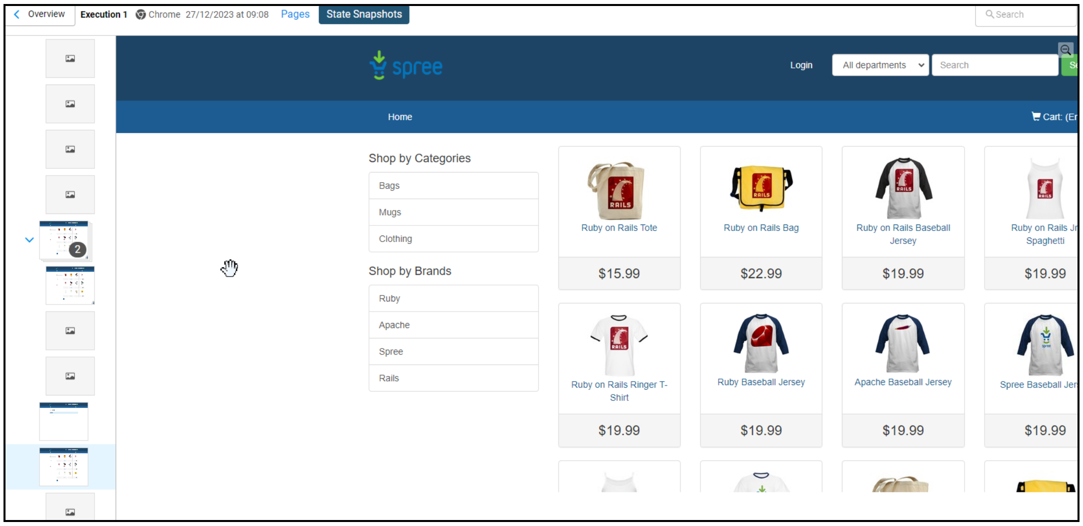
- If you click the **States** link from the Blueprint Dashboard under an execution you will see a list of pages and page states. Click the expand arrow to see the states for a specific page (highlighted in screen shot).



Categories - Clothing - Appvance Demo Site https://demosite.appvance.net/t/clothing	**2 / 25 actions complete**	3 inputs
Spree Baseball Jersey - Appvance Demo Site https://demosite.appvance.net/products/spree-baseball-jersey	**2 / 13 actions complete**	5 inputs
Spree Stein - Appvance Demo Site https://demosite.appvance.net/products/spree-stein	**2 / 12 actions complete**	5 inputs
Brands - Ruby - Appvance Demo Site https://demosite.appvance.net/t/ruby	**1 / 13 actions complete**	12 inputs
Appvance Demo Site https://demosite.appvance.net/products	**0 / 27 actions complete**	3 inputs
Categories - T-Shirts - Appvance Demo Site https://demosite.appvance.net/t/t-shirts	**0 / 18 actions complete**	12 inputs
Appvance Demo Site https://demosite.appvance.net/	**4 / 47 actions complete**	3 inputs
State 1 https://demosite.appvance.net/	**3 / 28 actions complete**	3 inputs
State 10 https://demosite.appvance.net/?page=2	**1 / 19 actions complete**	3 inputs
Ruby on Rails Tote - Appvance Demo Site https://demosite.appvance.net/products/ruby-on-rails-tote	**1 / 12 actions complete**	5 inputs
Categories - Bags - Appvance Demo Site https://demosite.appvance.net/t/bags	**1 / 15 actions complete**	12 inputs

- Using the **Search** bar you can search the Main Accessor, Value, Type, StateID, and Smart Tag elements.

- If you click the **State Snapshot** link you can explore various page states.

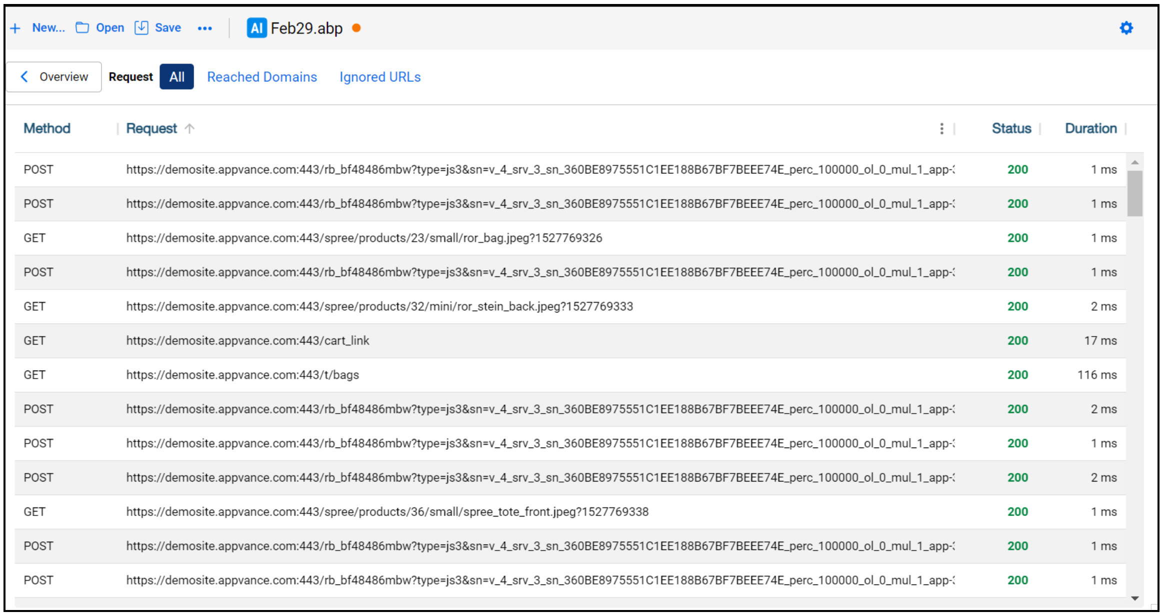


Failed Requests and Failed Actions

You can see details about failed actions and failed requests. There are different counters for each.

Failed Requests

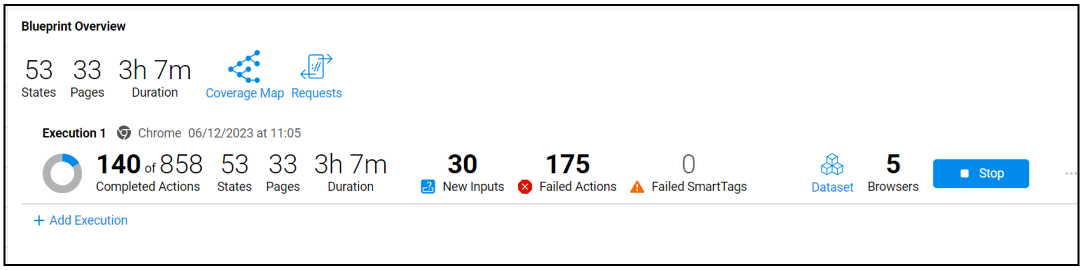
Click the **Requests** icon from the Blueprint Dashboard to see information on requests including failed requests. You can select one of the filters to show **All** requests, **Reached Domains**, or **Ignored URLs**.



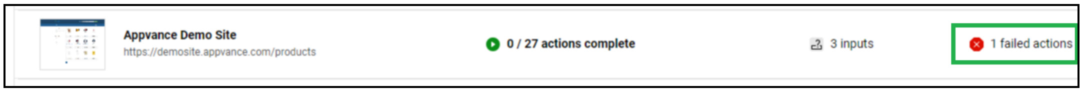
Method	Request ↑	Status	Duration
POST	https://demosite.appvance.com:443/rb_bf48486mbw?type=js3&sn=v_4_srv_3_sn_360BE8975551C1EE188B67BF7BEEE74E_perc_100000_ol_0_mul_1_app-	200	1 ms
POST	https://demosite.appvance.com:443/rb_bf48486mbw?type=js3&sn=v_4_srv_3_sn_360BE8975551C1EE188B67BF7BEEE74E_perc_100000_ol_0_mul_1_app-	200	1 ms
GET	https://demosite.appvance.com:443/spree/products/23/small/ror_bag.jpeg?1527769326	200	1 ms
POST	https://demosite.appvance.com:443/rb_bf48486mbw?type=js3&sn=v_4_srv_3_sn_360BE8975551C1EE188B67BF7BEEE74E_perc_100000_ol_0_mul_1_app-	200	1 ms
GET	https://demosite.appvance.com:443/spree/products/32/mini/ror_stein_back.jpeg?1527769333	200	2 ms
GET	https://demosite.appvance.com:443/cart_link	200	17 ms
GET	https://demosite.appvance.com:443/t/bags	200	116 ms
POST	https://demosite.appvance.com:443/rb_bf48486mbw?type=js3&sn=v_4_srv_3_sn_360BE8975551C1EE188B67BF7BEEE74E_perc_100000_ol_0_mul_1_app-	200	2 ms
POST	https://demosite.appvance.com:443/rb_bf48486mbw?type=js3&sn=v_4_srv_3_sn_360BE8975551C1EE188B67BF7BEEE74E_perc_100000_ol_0_mul_1_app-	200	1 ms
POST	https://demosite.appvance.com:443/rb_bf48486mbw?type=js3&sn=v_4_srv_3_sn_360BE8975551C1EE188B67BF7BEEE74E_perc_100000_ol_0_mul_1_app-	200	2 ms
GET	https://demosite.appvance.com:443/spree/products/36/small/spree_tote_front.jpeg?1527769338	200	1 ms
POST	https://demosite.appvance.com:443/rb_bf48486mbw?type=js3&sn=v_4_srv_3_sn_360BE8975551C1EE188B67BF7BEEE74E_perc_100000_ol_0_mul_1_app-	200	1 ms
POST	https://demosite.appvance.com:443/rb_bf48486mbw?type=js3&sn=v_4_srv_3_sn_360BE8975551C1EE188B67BF7BEEE74E_perc_100000_ol_0_mul_1_app-	200	1 ms

Failed Actions

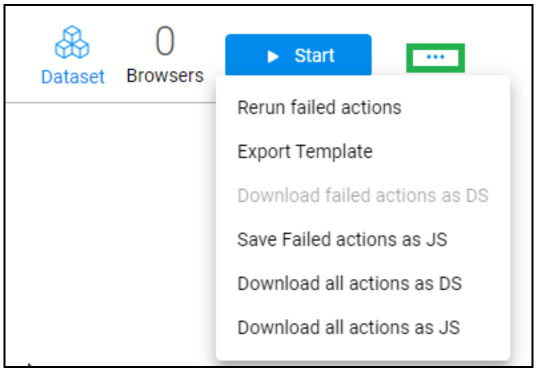
- The Blueprint Dashboard has a Failed Actions counter. Click on this counter to open the main page for the execution, From there you will need to scroll through the pages to located the failed actions.



- You can also click a link from one of the Pages or States detail screen to see information failed actions.



- You can also download a file containing failed actions from the Blueprint Dashboard.


Dataset
0 Browsers
Start
•••
Rerun failed actions
Export Template
Download failed actions as DS
Save Failed actions as JS
Download all actions as DS
Download all actions as JS

Failed Actions in Counter vs Failed Actions in Downloaded File

You may see a different number of failed actions in the Failed Actions counter than the number of failed actions in the downloaded file.

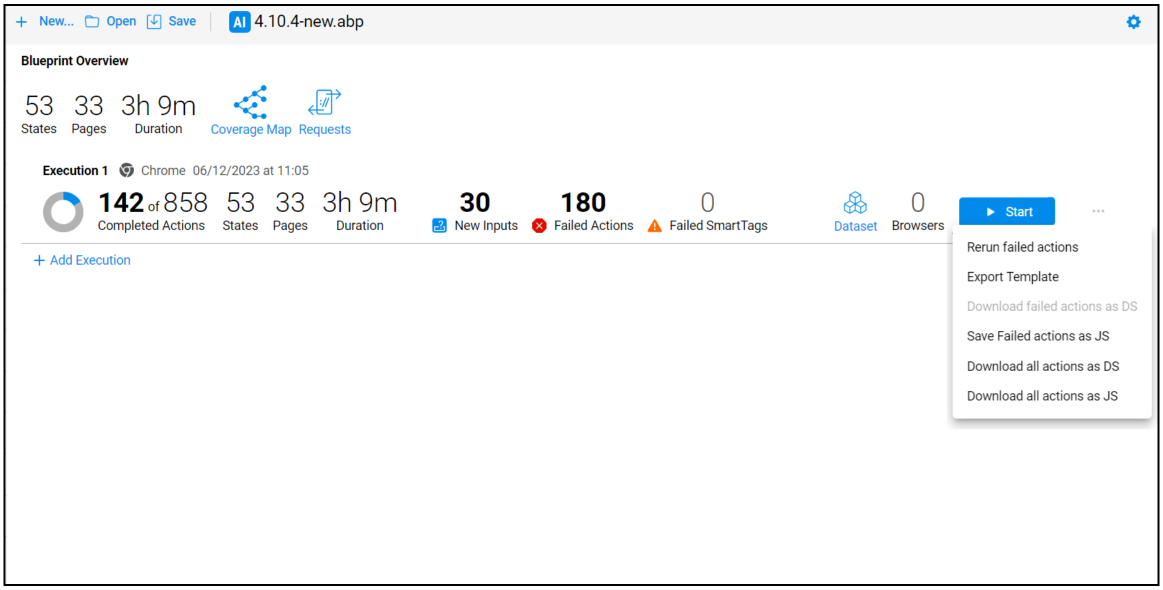
The Failed Actions counter shows the true amount of failed actions. The count increases when the Blueprint tried to interact with some thing, (Button A for example) and nothing happened or a failure resulted.

What is the difference between that number and the number in the downloaded file?

Example : Logic

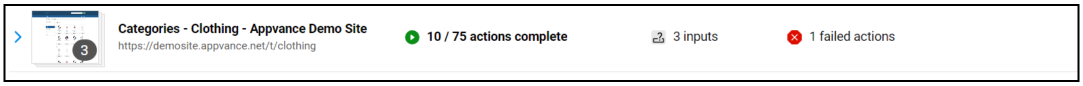
- We have four actions in sequence: A > B > C > D.
- Action A succeeded and brings us to Action B, and then to Action C.
- Action C failed.
- Because of the order in which the buttons are being interacted with, it will not show on the Failed Actions Counter. However, it will display if you download all failed actions in a file.

Example : Practical



- The Failed Action Counter displays 180 failed actions.
- However, if you download the failed actions in a DS/JS file, there are 54 failed actions.
- We have four actions in sequence: A > B > C > D. This is the sequence of steps that the AI will attempt to perform,
- The Blueprint execution fails at Step C.
- The failed actions download gives you all the steps up to the failure.
- The Failed Actions Counter lists that the execution failed at Step A because it did not get past the first action. Since, Step A failed, there are no steps to download. You have to drill down into the execution to see where the step failed.

  For example you have to click the above, then see what action failed.



Categories - Clothing - Appvance Demo Site
https://demosite.appvance.net/t/clothing
10 / 75 actions complete
3 inputs
1 failed actions

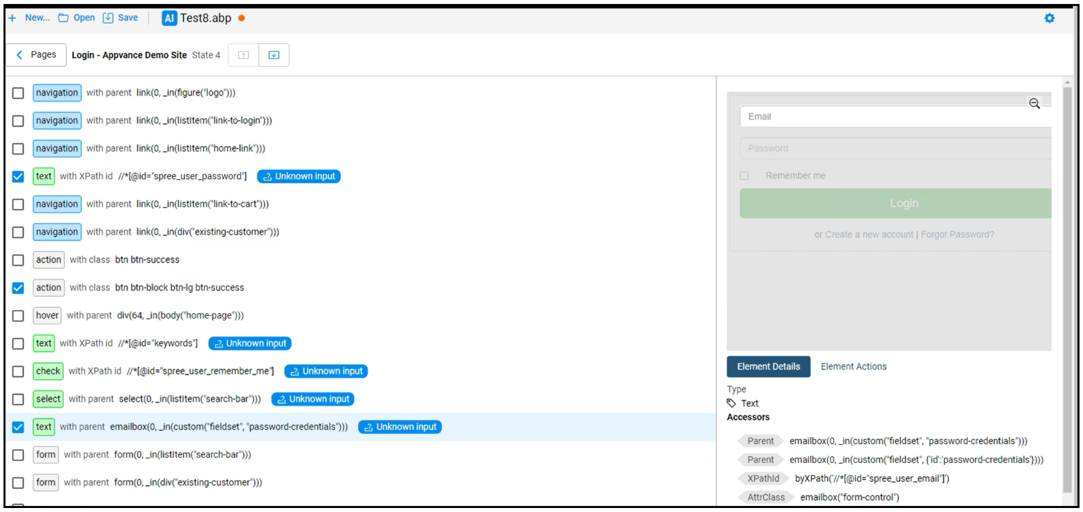
Creating Custom Actions

You can create a custom action to resolve failed actions.

1. Determine which actions you want to interact with. You can create custom actions for a single action or multiple actions.

2. Click **Create Custom Action** at the top of the page to create multiple custom actions or click the blue **Unknown Input** button by failed action to create a custom action for that specific action.

3. Select the actions you want to interact with. This shows Email, Password, and the submit button.

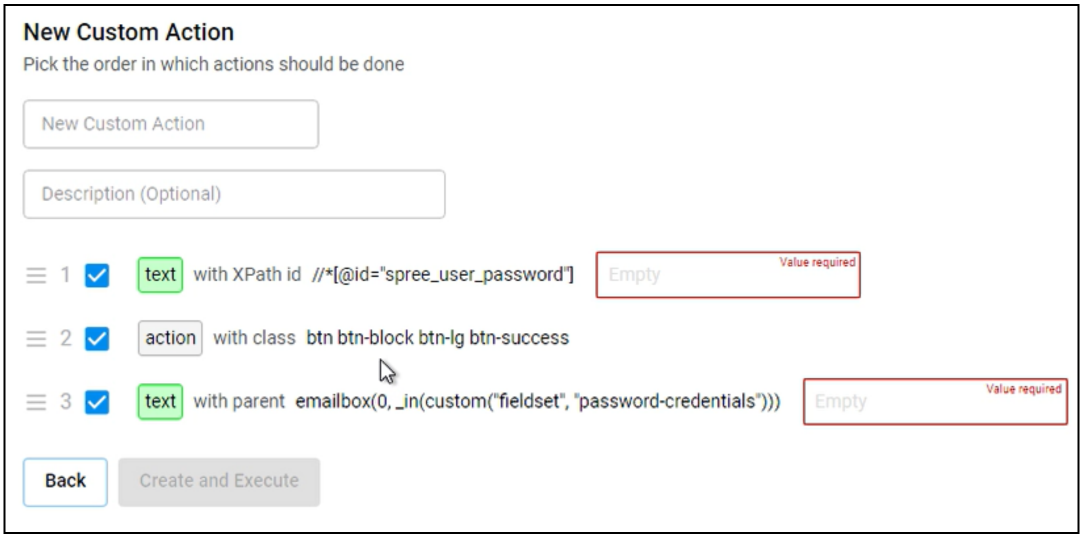


- □ navigation with parent link(0, _in(figure("logo")))
- □ navigation with parent link(0, _in(listItem("link-to-login")))
- □ navigation with parent link(0, _in(listItem("home-link")))
- ☑ text with XPath id //*[@id="spree_user_password"] Unknown input
- □ navigation with parent link(0, _in(listItem("link-to-cart")))
- □ navigation with parent link(0, _in(div("existing-customer")))
- □ action with class btn btn-success
- ☑ action with class btn btn-block btn-lg btn-success
- □ hover with parent div(64, _in(body("home-page")))
- □ text with XPath id //*[@id="keywords"] Unknown input
- □ check with XPath id //*[@id="spree_user_remember_me"] Unknown input
- □ select with parent select(0, _in(listItem("search-bar"))) Unknown input
- ☑ text with parent emailbox(0, _in(custom("fieldset", "password-credentials"))) Unknown input
- □ form with parent form(0, _in(listItem("search-bar")))
- □ form with parent form(0, _in(div("existing-customer")))

Type
Text
Accessors

Parent	emailbox(0, _in(custom("fieldset", "password-credentials")))
Parent	emailbox(0, _in(custom("fieldset", {'id':'password-credentials'})))
XPathId	byXPath('//*[@id="spree_user_email"]')
AttrClass	emailbox("form-control")

4. Click **Next** to open the **Create New Action** page.



New Custom Action

Pick the order in which actions should be done

New Custom Action

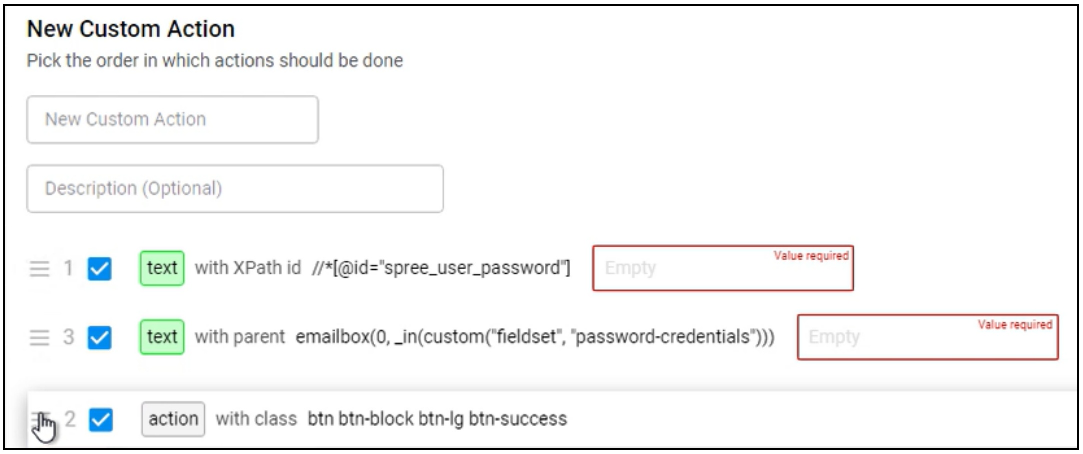
Description (Optional)

1 ☑ text with XPath id //*[@id="spree_user_password"] Empty Value required

2 ☑ action with class btn btn-block btn-lg btn-success

3 ☑ text with parent emailbox(0, _in(custom("fieldset", "password-credentials"))) Empty Value required

5. If needed you can reorder the actions by dragging and dropping them.



New Custom Action

Pick the order in which actions should be done

New Custom Action

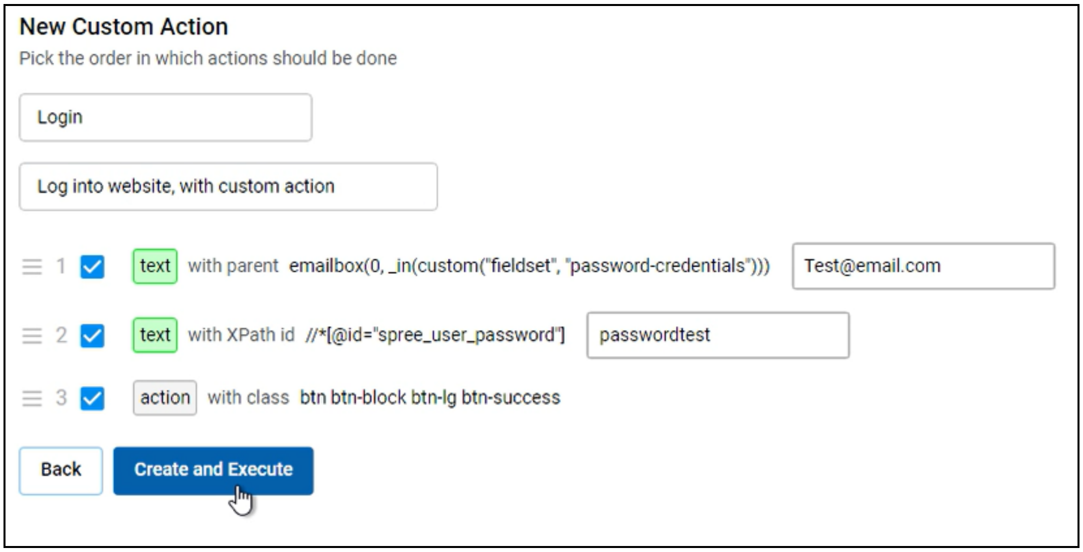
Description (Optional)

1 ☑ text with XPath id //*[@id="spree_user_password"] Empty Value required

3 ☑ text with parent emailbox(0, _in(custom("fieldset", "password-credentials"))) Empty Value required

2 ☑ action with class btn btn-block btn-lg btn-success

6. Define all the information for your custom actions. Any information you define here will be saved in a default.csv file. If you have a data set containing your variables you can select from that file here. See the information on adding Data Sets to an execution.



New Custom Action

Pick the order in which actions should be done

Login

Log into website, with custom action

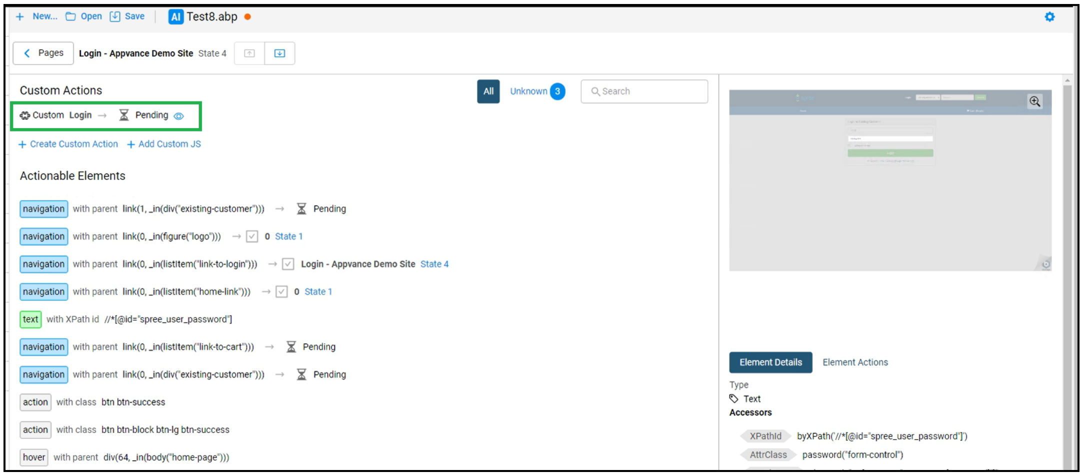
1 ☑ text with parent emailbox(0, _in(custom("fieldset", "password-credentials"))) Test@email.com

2 ☑ text with XPath id //*[@id="spree_user_password"] passwordtest

3 ☑ action with class btn btn-block btn-lg btn-success

7. Click **Create and Execute**.

8. Your new custom action displays in the **Custom Actions** section.



Login - Appvance Demo Site State 4

Custom Actions

Custom Login → Pending

Actionable Elements

navigation with parent link(1, _in(div("existing-customer"))) → Pending

navigation with parent link(0, _in(figure("logo"))) → ☑ 0 State 1

navigation with parent link(0, _in(listItem("link-to-login"))) → ☑ Login - Appvance Demo Site State 4

navigation with parent link(0, _in(listItem("home-link"))) → ☑ 0 State 1

text with XPath id //*[@id="spree_user_password"]

navigation with parent link(0, _in(listItem("link-to-cart"))) → Pending

navigation with parent link(0, _in(div("existing-customer"))) → Pending

action with class btn btn-success

action with class btn btn-block btn-lg btn-success

hover with parent div(64, _in(body("home-page")))

Type

Text

Accessors

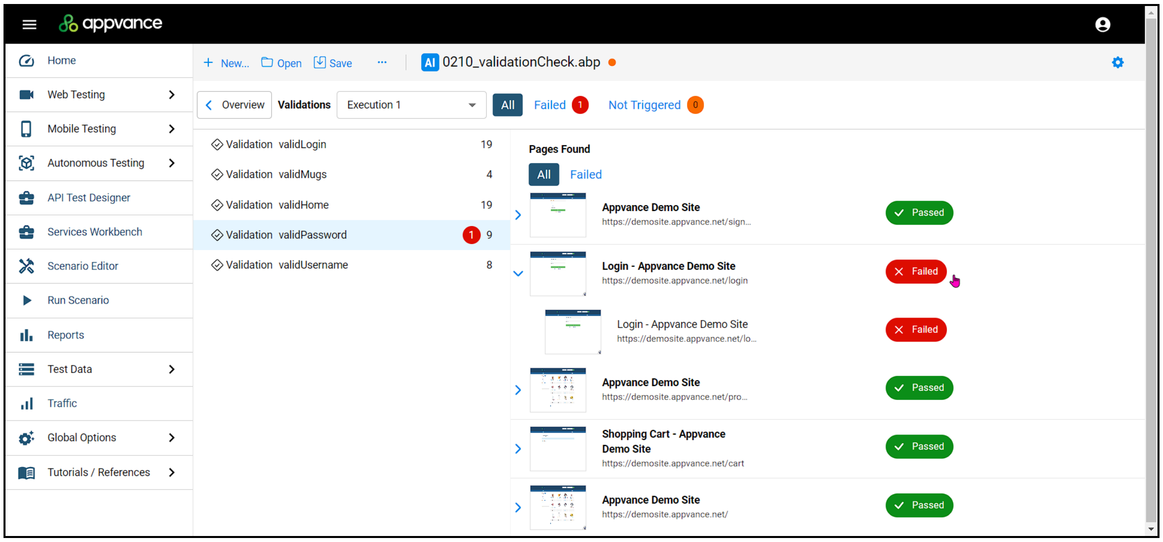
XPathId byXPath('//*[@id="spree_user_password"]')

AttrClass password("form-control")

9. The Blueprint will try and execute your new custom action as soon as possible.

Exploring Validations

If you click the **Validations** button from the Blueprint Overview, you can see details of the validations within your Blueprint.



Validation validLogin	19
Validation validMugs	4
Validation validHome	19
Validation validPassword	1 9
Validation validUsername	8

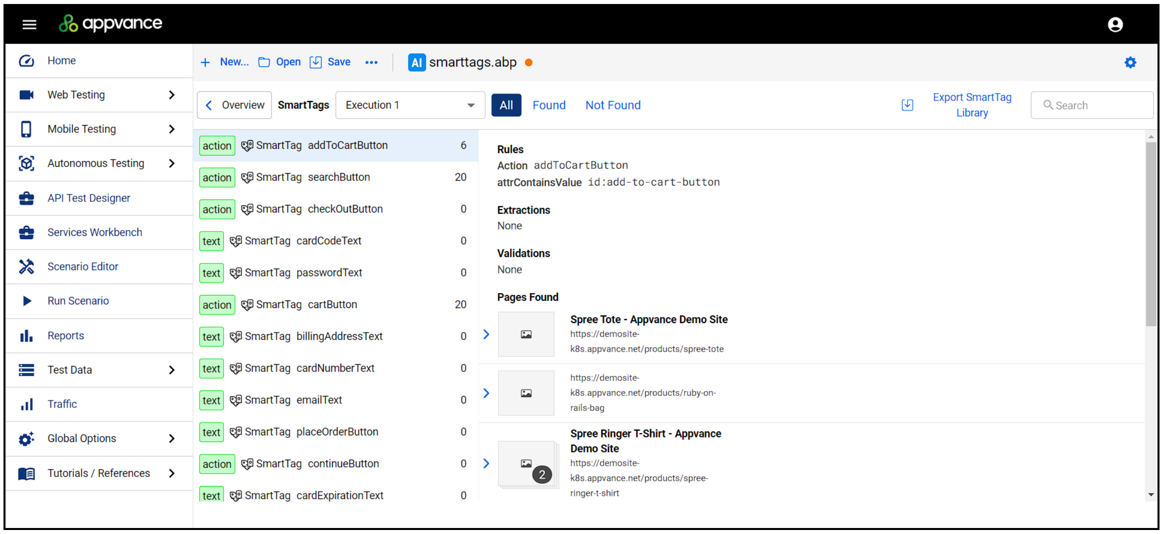
Appvance Demo Site https://demosite.appvance.net/sign...	Passed
Login - Appvance Demo Site https://demosite.appvance.net/login	Failed
Login - Appvance Demo Site https://demosite.appvance.net/lo...	Failed
Appvance Demo Site https://demosite.appvance.net/pro...	Passed
Shopping Cart - Appvance Demo Site https://demosite.appvance.net/cart	Passed
Appvance Demo Site https://demosite.appvance.net/	Passed

You can view:

- All validations
- Failed validations
- Validations that were not triggered
- Specific pages where the validation is present and view only the pages where the validation failed

Exploring SmartTags

If you click the **SmartTags** button from the Blueprint Overview, you can see details of the SmartTags within your Blueprint.



action	SmartTag addToCartButton	6
action	SmartTag searchButton	20
action	SmartTag checkOutButton	0
text	SmartTag cardCodeText	0
text	SmartTag passwordText	0
action	SmartTag cartButton	20
text	SmartTag billingAddressText	0
text	SmartTag cardNumberText	0
text	SmartTag emailText	0
text	SmartTag placeOrderButton	0
action	SmartTag continueButton	0
text	SmartTag cardExpirationText	0

Rules
Action `addToCartButton`
attrContainsValue `id:add-to-cart-button`

Extractions
None

Validations
None

Pages Found

Spree Tote - Appvance Demo Site
https://demosite-k8s.appvance.net/products/spree-tote

https://demosite-k8s.appvance.net/products/ruby-on-rails-bag

Spree Ringer T-Shirt - Appvance Demo Site
https://demosite-k8s.appvance.net/products/spree-ringer-t-shirt

You can view:

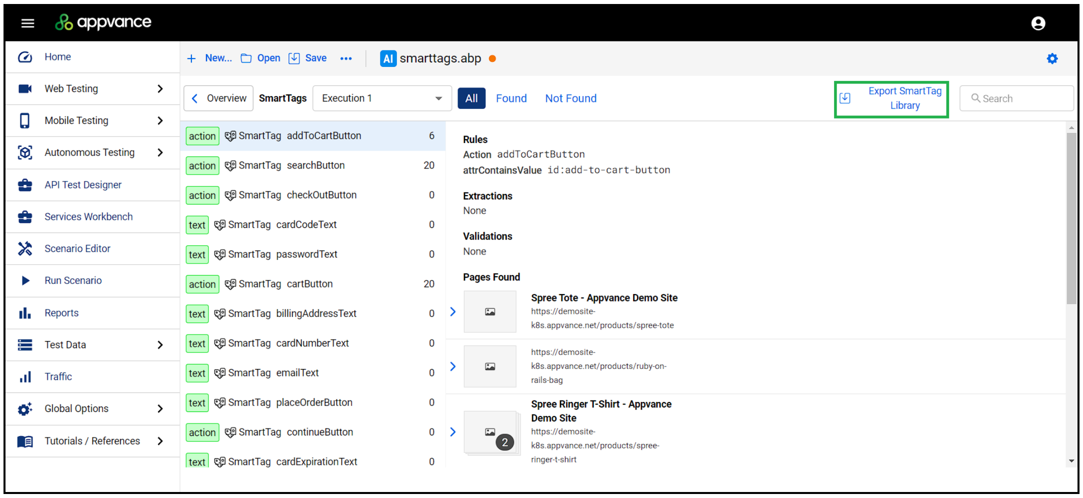
- All SmartTags

- SmartTags that have been found or not found and the associated pages

Information on SmartTags can be found in the documentation on the SmartTag Workbench.

Exporting SmartTags

1. Navigate to the SmartTags summary page and select a Blueprint execution from the dropdown.



action	SmartTag addToCartButton	6
action	SmartTag searchButton	20
action	SmartTag checkOutButton	0
text	SmartTag cardCodeText	0
text	SmartTag passwordText	0
action	SmartTag cartButton	20
text	SmartTag billingAddressText	0
text	SmartTag cardNumberText	0
text	SmartTag emailText	0
text	SmartTag placeOrderButton	0
action	SmartTag continueButton	0
text	SmartTag cardExpirationText	0

Rules
Action `addToCartButton`
attrContainsValue `id:add-to-cart-button`

Extractions
None

Validations
None

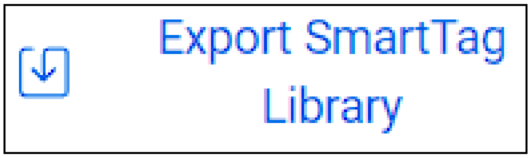
Pages Found

Spree Tote - Appvance Demo Site
https://demosite-k8s.appvance.net/products/spree-tote

https://demosite-k8s.appvance.net/products/ruby-on-rails-bag

Spree Ringer T-Shirt - Appvance Demo Site
https://demosite-k8s.appvance.net/products/spree-ringer-t-shirt

2. Click **Export SmartTag Library**.



3. Browse to the location in your repository.

4. In the **Save As** dialog, name your exported file.

5. Click **Save**.

6. To open your exported SmartTag library navigate to the SmartTag Workbench:

   - For web testing: **AIQ** > **Web Testing** > **SmartTag Workbench**.

   - For mobile testing: **AIQ** > **Mobile Testing** > **SmartTag Workbench**.

Information on SmartTags can be found in the documentation on the SmartTag Workbench.

AI 3.0 Example - Blueprints for Salesforce Applications

This is an example of creating an AI blueprint for use in testing Salesforce applications. This example covers the required pre-requisites, some recommended settings, and instructions on running the AI Blueprint.

- "Setup and Configuration of AI Blueprints for Salesforce Applications" below
- "Executing the Blueprint for the First Time" on page 109
- "Training the AI" on page 111

Setup and Configuration of AI Blueprints for Salesforce Applications

The following information details the required setup and configuration for this example.

Accessor Preferences

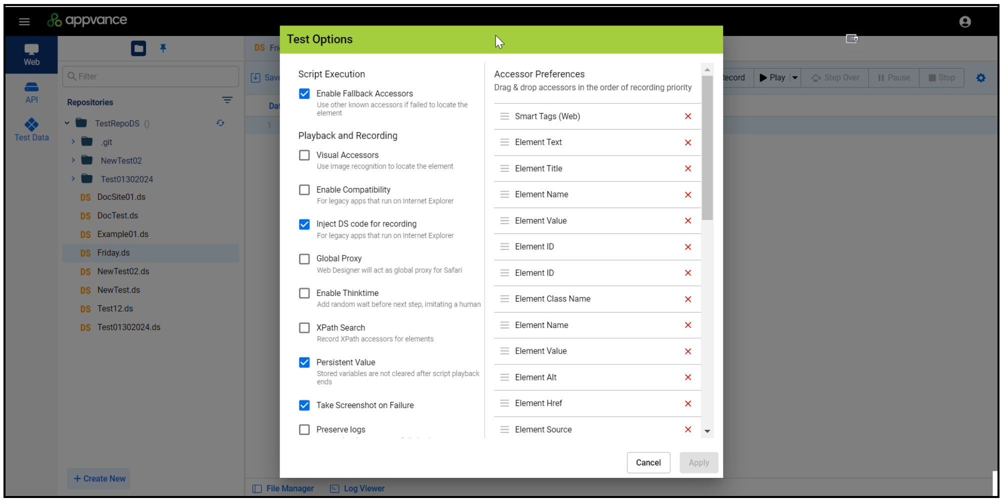
Accessor preferences must be set on the controller or test node where the blueprint will be run. This is because each Test Designer instance is local to the machine where it is installed.

For example, if you want to run a Blueprint on Machine XX via Controller YY, access machine XX via remote desktop protocol and login to an AIQ instance of either YY or XX.

> Whenever a Controller or the Test Node is upgraded, the accessor preferences are reset. You must make sure that the preferences are set again post-upgrade.

1. Navigate to the Web Designer to set the accessor preferences. You can use either the new Web Designer interface or the Web Designer (Classic) interface.

   - **Web Testing** > **Web Designer** > **Options** (gear icon) > **Accessor Preferences**.



Test Options

Script Execution

☑ Enable Fallback Accessors
Use other known accessors if failed to locate the element

Playback and Recording

□ Visual Accessors
Use image recognition to locate the element

□ Enable Compatibility
For legacy apps that run on Internet Explorer

☑ Inject DS code for recording
For legacy apps that run on Internet Explorer

□ Global Proxy
Web Designer will act as global proxy for Safari

□ Enable Thinktime
Add random wait before next step, imitating a human

□ XPath Search
Record XPath accessors for elements

☑ Persistent Value
Stored variables are not cleared after script playback ends

☑ Take Screenshot on Failure

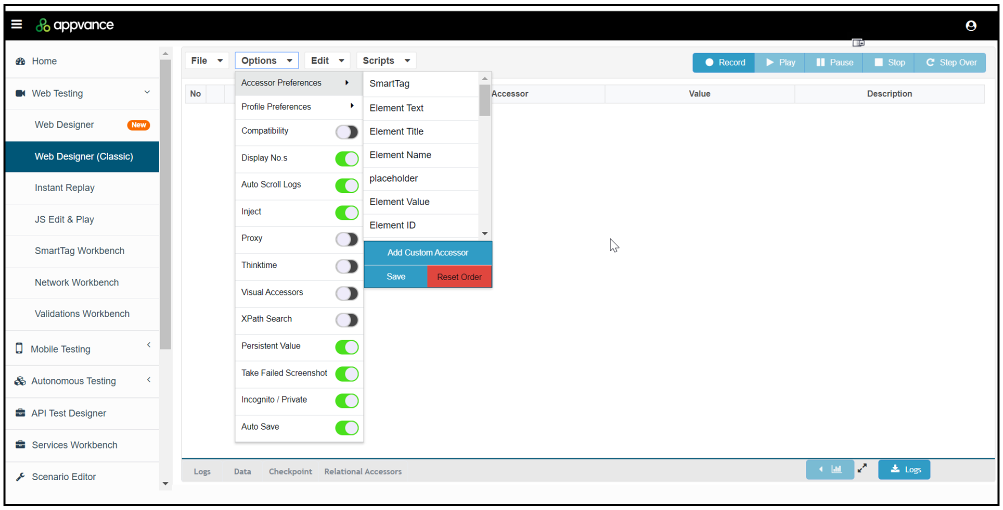
□ Preserve logs

Accessor Preferences
Drag & drop accessors in the order of recording priority

- Smart Tags (Web)
- Element Text
- Element Title
- Element Name
- Element Value
- Element ID
- Element ID
- Element Class Name
- Element Name
- Element Value
- Element Alt
- Element Href
- Element Source

   - **Web Testing** > **Web Designer (Classic)** > **Options** > **Accessor**

Preferences.



Compatibility ○
Display No.s ●
Auto Scroll Logs ●
Inject ●
Proxy ○
Thinktime ○
Visual Accessors ○
XPath Search ○
Persistent Value ●
Take Failed Screenshot ●
Incognito / Private ●
Auto Save ●

2. Make sure that the **XPath Search** is disabled.

3. Click **Reset** to reorient to the default list.

4. Update the Accessor Preferences as follows:

- SmartTag
- Element Text
- Element Title
- Element Name
- Element Value
- Element ID
- Element Class Name
- Element Parent
- Element Alt
- Element Href

5. Leave the remaining accessors in default order.

6. Click **Save** in Web Designer (Classic) or click **Apply** in Web Designer..

Blueprint Configuration

Various settings need to be enabled and configured in the Blueprint.

- Scan Type
- Blueprint Settings
- Getting Valid URLs
- Browser Settings
- Proxy Settings
- Data Production Library

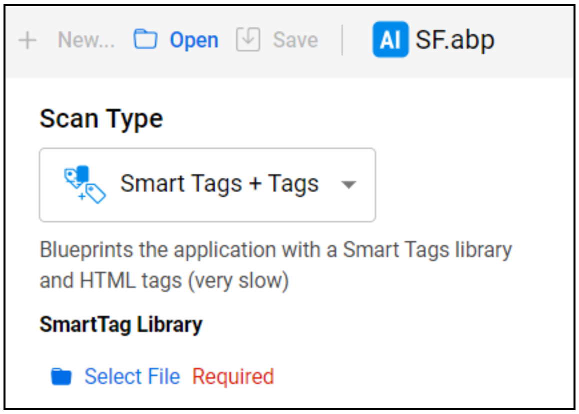
Scan Type

The scan type determines how the application will be scanned.

In the Blueprint Designer:

1. In the **Scan Type** option, select **SmartTags + Tags**. This means the scan will use a combination of Tags and SmartTags.

2. In **SmartTag Library** click **Select File** to add your SmartTag file.



Scan Type

Smart Tags + Tags

Blueprints the application with a Smart Tags library and HTML tags (very slow)

SmartTag Library

For more information about SmartTags, refer to these topics:

- Web SmartTags
- Understanding your Application (with SmartTags)

- Useful smartags workbench custom actions and other js examples

Blueprint Settings

Enable the following options:

- **Visual Hints** - This means that screenshots will be saved for each page state.

> This option only applies to AI 2.0. Screenshots are automatically saved for each page state in AI 3.0.

- **Shadow Dom** - Must be enabled for applications that are developed using Shadow Dom. Enabling this helps the AI to scan all of the custom accessors in the application.

Getting Valid URLs

You need to identify all of the valid URLs for the application you are testing, using the base URL of the application as the starting point.

Salesforce example base URL: `https://d41000000649seaq-dev-ed.lightning.force.com/lightning/page/home`

Using the Web Designer (Classic) interface

To identify all the valid URLs :

1. Navigate to **Web Testing** > **Instant Replay**.
2. Click **Record**. The status should display “Record”.
3. Navigate to **Web Designer (Classic)**.
4. Click **Record.**

5. Select “Chrome” as the browser type.

6. Enter the base URL in the new browser launched by Test Designer.

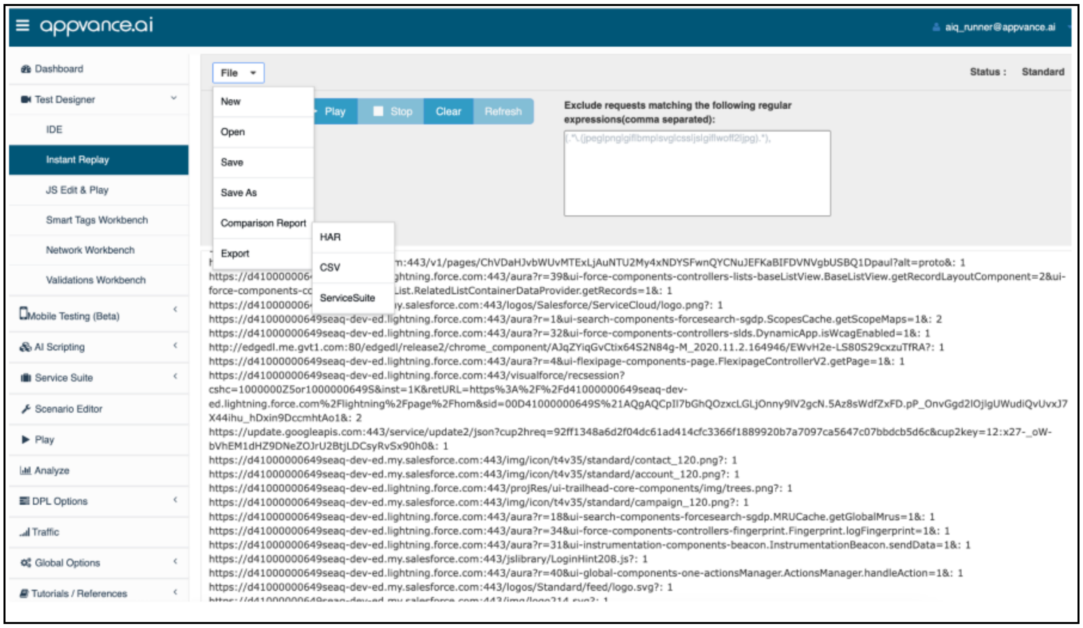
7. Login to the Salesforce application.

8. Perform a few actions.

9. Navigate back to **Instant Replay**. A list of URLs is displayed.

10. Click **Stop**.

11. Click **File** > **Export** > **CSV**.

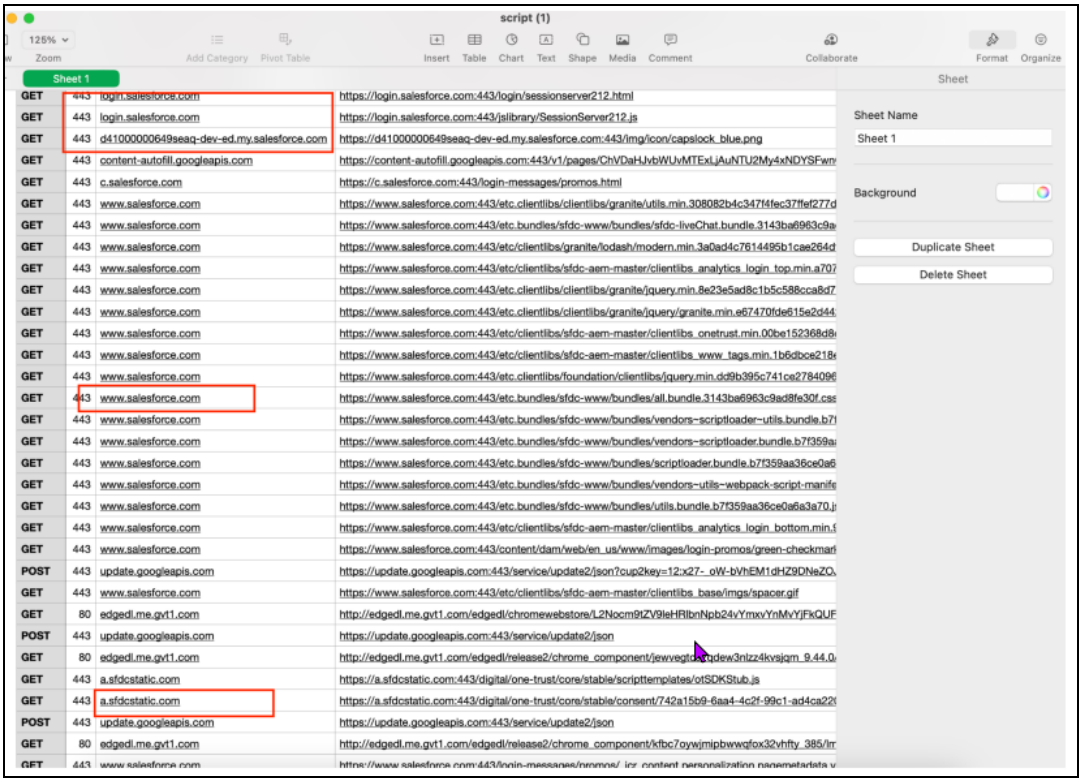


Status : Standard

```
h                                    m:443/v1/pages/ChVDaHJvbWUvMTExLjAuNTU2My4xNDYSFwnQYCNuJEFKaBIFDVNVgbUSBQ1Dpaul?alt=proto&: 1
https://d410000006                   ghtning.force.com:443/aura?r=39&ui-force-components-controllers-lists-baseListView.BaseListView.getRecordLayoutComponent=2&ui-
force-components-cc                  List.RelatedListContainerDataProvider.getRecords=1&: 1
https://d410000006                   y.salesforce.com:443/logos/Salesforce/ServiceCloud/logo.png?: 1
https://d41000000649seaq-dev-ed.lightning.force.com:443/aura?r=1&ui-search-components-forcesearch-sgdp.ScopesCache.getScopeMaps=1&: 2
https://d41000000649seaq-dev-ed.lightning.force.com:443/aura?r=32&ui-force-components-controllers-slds.DynamicApp.isWcagEnabled=1&: 1
http://edgedl.me.gvt1.com:80/edgedl/release2/chrome_component/AJqZYiqGvCtix64S2N84g-M_2020.11.2.164946/EWvH2e-LS80S29cxzuTfRA?: 1
https://d41000000649seaq-dev-ed.lightning.force.com:443/aura?r=4&ui-flexipage-components-page.FlexipageControllerV2.getPage=1&: 1
https://d41000000649seaq-dev-ed.lightning.force.com:443/visualforce/recsession?
cshc=1000000Z5or1000000649S&inst=1K&retURL=https%3A%2F%2Fd41000000649seaq-dev-
ed.lightning.force.com%2Flightning%2Fpage%2Fhom&sid=00D41000000649S%21AQgAQCpIl7bGhQOzxcLGLjOnny9lV2gcN.5Az8sWdfZxFD.pP_OnvGgd2lOjlgUWudiQvUvxJ7
X44ihu_hDxin9DccmhtAo1&: 2
https://update.googleapis.com:443/service/update2/json?cup2hreq=92ff1348a6d2f04dc61ad414cfc3366f1889920b7a7097ca5647c07bbdcb5d6c&cup2key=12:x27-_oW-
bVhEM1dHZ9DNeZOJrU2BtjLDCsyRvSx90h0&: 1
https://d41000000649seaq-dev-ed.my.salesforce.com:443/img/icon/t4v35/standard/contact_120.png?: 1
https://d41000000649seaq-dev-ed.my.salesforce.com:443/img/icon/t4v35/standard/account_120.png?: 1
https://d41000000649seaq-dev-ed.lightning.force.com:443/projRes/ui-trailhead-core-components/img/trees.png?: 1
https://d41000000649seaq-dev-ed.my.salesforce.com:443/img/icon/t4v35/standard/campaign_120.png?: 1
https://d41000000649seaq-dev-ed.lightning.force.com:443/aura?r=18&ui-search-components-forcesearch-sgdp.MRUCache.getGlobalMrus=1&: 1
https://d41000000649seaq-dev-ed.lightning.force.com:443/aura?r=34&ui-force-components-controllers-fingerprint.Fingerprint.logFingerprint=1&: 1
https://d41000000649seaq-dev-ed.lightning.force.com:443/aura?r=31&ui-instrumentation-components-beacon.InstrumentationBeacon.sendData=1&: 1
https://d41000000649seaq-dev-ed.my.salesforce.com:443/jslibrary/LoginHint208.js?: 1
https://d41000000649seaq-dev-ed.lightning.force.com:443/aura?r=40&ui-global-components-one-actionsManager.ActionsManager.handleAction=1&: 1
https://d41000000649seaq-dev-ed.my.salesforce.com:443/logos/Standard/feed/logo.svg?: 1
https://d41000000649seaq-dev-ed.my.salesforce.com:443/img/logo214.svg?: 1
```

12. From the generated CSV file, identify all valid URLs pertaining to the application, keeping in mind the application may use domains outside of the primary domain of valid URLs pertaining to the application.

For example, the highlighted URLs in the following screenshot pertain to Salesforce. Notice also the ‘noise’ from the browser, for example most of those containing google as well as Google Chrome.



GET	443	login.salesforce.com	https://login.salesforce.com:443/login/sessionserver212.html
GET	443	login.salesforce.com	https://login.salesforce.com:443/jslibrary/SessionServer212.js
GET	443	d41000000649seaq-dev-ed.my.salesforce.com	https://d41000000649seaq-dev-ed.my.salesforce.com:443/img/icon/capslock_blue.png
GET	443	content-autofill.googleapis.com	https://content-autofill.googleapis.com:443/v1/pages/ChVDaHJvbWUvMTExLjAuNTU2My4xNDYSFwn
GET	443	c.salesforce.com	https://c.salesforce.com:443/login-messages/promos.html
GET	443	www.salesforce.com	https://www.salesforce.com:443/etc.clientlibs/clientlibs/granite/utils.min.308082b4c347f4fec37ffef277d
GET	443	www.salesforce.com	https://www.salesforce.com:443/etc.bundles/sfdc-www/bundles/sfdc-liveChat.bundle.3143ba6963c9a
GET	443	www.salesforce.com	https://www.salesforce.com:443/etc/clientlibs/granite/lodash/modern.min.3a0ad4c7614495b1cae264d
GET	443	www.salesforce.com	https://www.salesforce.com:443/etc/clientlibs/sfdc-aem-master/clientlibs_analytics_login_top.min.a707
GET	443	www.salesforce.com	https://www.salesforce.com:443/etc.clientlibs/clientlibs/granite/jquery.min.8e23e5ad8c1b5c588cca8d7
GET	443	www.salesforce.com	https://www.salesforce.com:443/etc.clientlibs/clientlibs/granite/jquery/granite.min.e67470fde615e2d44
GET	443	www.salesforce.com	https://www.salesforce.com:443/etc/clientlibs/sfdc-aem-master/clientlibs_onetrust.min.00be152368d8
GET	443	www.salesforce.com	https://www.salesforce.com:443/etc/clientlibs/sfdc-aem-master/clientlibs_www_tags.min.1b6dbce218
GET	443	www.salesforce.com	https://www.salesforce.com:443/etc.clientlibs/foundation/clientlibs/jquery.min.dd9b395c741ce2784096
GET	443	www.salesforce.com	https://www.salesforce.com:443/etc.bundles/sfdc-www/bundles/all.bundle.3143ba6963c9ad8fe30f.css
GET	443	www.salesforce.com	https://www.salesforce.com:443/etc.bundles/sfdc-www/bundles/vendors~scriptloader~utils.bundle.b7
GET	443	www.salesforce.com	https://www.salesforce.com:443/etc.bundles/sfdc-www/bundles/vendors~scriptloader.bundle.b7f359a
GET	443	www.salesforce.com	https://www.salesforce.com:443/etc.bundles/sfdc-www/bundles/scriptloader.bundle.b7f359aa36ce0a6
GET	443	www.salesforce.com	https://www.salesforce.com:443/etc.bundles/sfdc-www/bundles/vendors~utils~webpack-script-manife
GET	443	www.salesforce.com	https://www.salesforce.com:443/etc.bundles/sfdc-www/bundles/utils.bundle.b7f359aa36ce0a6a3a70.j
GET	443	www.salesforce.com	https://www.salesforce.com:443/etc/clientlibs/sfdc-aem-master/clientlibs_analytics_login_bottom.min.
GET	443	www.salesforce.com	https://www.salesforce.com:443/content/dam/web/en_us/www/images/login-promos/green-checkmar
POST	443	update.googleapis.com	https://update.googleapis.com:443/service/update2/json?cup2key=12:x27-_oW-bVhEM1dHZ9DNeZO
GET	443	www.salesforce.com	https://www.salesforce.com:443/etc/clientlibs/sfdc-aem-master/clientlibs_base/imgs/spacer.gif
GET	80	edgedl.me.gvt1.com	http://edgedl.me.gvt1.com/edgedl/chromewebstore/L2Nocm9tZV9leHRlbnNpb24vYmxvYnMvYjFkQUF
POST	443	update.googleapis.com	https://update.googleapis.com:443/service/update2/json
GET	80	edgedl.me.gvt1.com	http://edgedl.me.gvt1.com/edgedl/release2/chrome_component/jewvegtd[illegible]qdew3nlzz4kvsjqm_9.44.0
GET	443	a.sfdcstatic.com	https://a.sfdcstatic.com:443/digital/one-trust/core/stable/scripttemplates/otSDKStub.js
GET	443	a.sfdcstatic.com	https://a.sfdcstatic.com:443/digital/one-trust/core/stable/consent/742a15b9-6aa4-4c2f-99c1-ad4ca22
POST	443	update.googleapis.com	https://update.googleapis.com:443/service/update2/json
GET	80	edgedl.me.gvt1.com	http://edgedl.me.gvt1.com/edgedl/release2/chrome_component/kfbc7oywjmipbwwqfox32vhfty_385/Im
GET	443	www.salesforce.com	https://www.salesforce.com:443/login-messages/promos/_jcr_content.personalization.pagemetadata.v

Browser Settings

Configure your browser settings.

On the **Add Execution** page:

- **Start with** : Specifies the number of AI bots / browsers to run. If you have a machine with sufficient resources, you can run up to twenty AI bots / browsers concurrently.

- **Browser** : Select the browser that will run the Blueprint. Chrome is the default browser.

- **Use Headless Browser** : Select whether to use a headless browser. Enabled by default.

- **Delay Between Requests (ms)** : This determines the wait time between requests and should be adjusted based on the responsiveness of your deployment. See "Resource Considerations" on page 29for more information.

Proxy Settings

Configure your proxy settings.

1. Configure a **Proxy Address**:

    - Localhost - If running the Blueprint on the same controller.

    - IP address of the test node - If running the Blueprint on the test node via the controller.

2. Configure a **Proxy Port**. Port 9090 is recommended.

Data Production Library

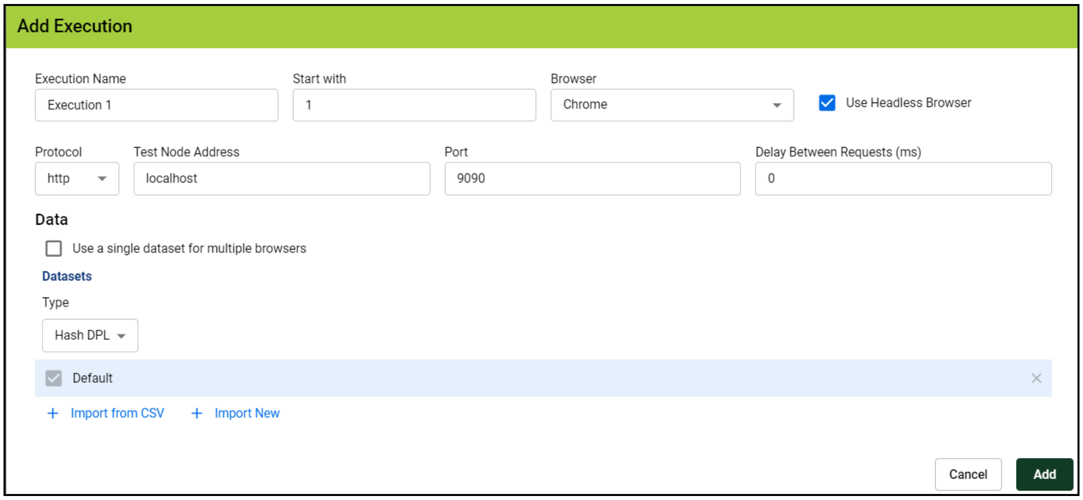
Configure your DPL settings. You can add a Synthetic or Java Script file defining all the input values required.

For more information on DPLs, see Operational Test Data with DPL's.

Executing the Blueprint for the First Time

1. Click **Execute** to start a new execution of the Blueprint.

2. The **Add Execution** screen opens.



Add Execution

Execution Name: Execution 1

Start with: 1

Browser: Chrome

☑ Use Headless Browser

Protocol: http

Test Node Address: localhost

Port: 9090

Delay Between Requests (ms): 0

Data

□ Use a single dataset for multiple browsers

Datasets

Type: Hash DPL

☑ Default

3. Define your Blueprint execution parameters. See "Add an Execution" on page 71 for more information.

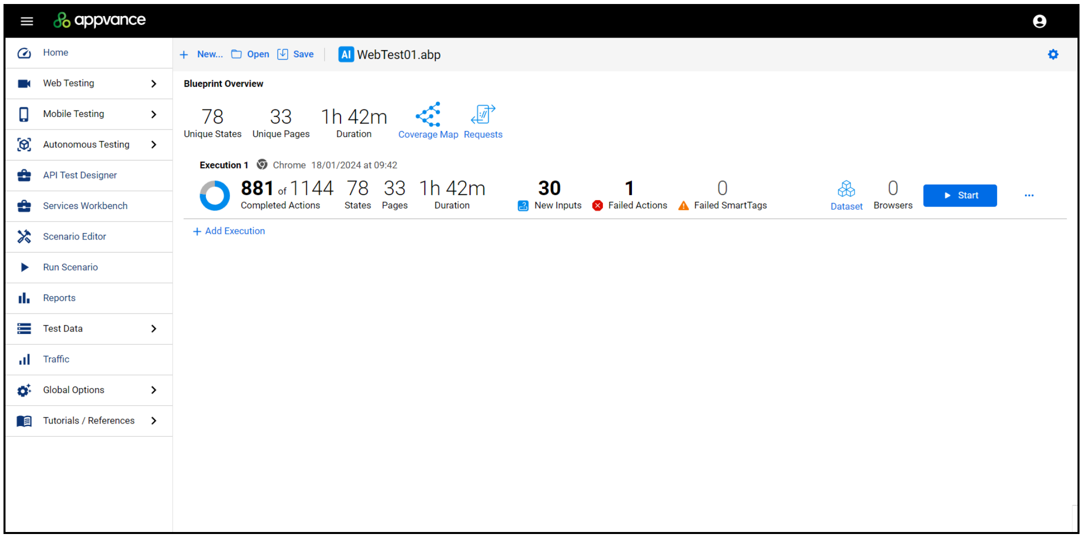
4. Click **Add** to start the Blueprint execution.

When the Blueprint for the first time, the following things happen:

- AI navigates to the application URL.

- AI tries to click all of the elements found in each page it discovers.

- The Blueprint stops when it can go no further and displays the resulting in page states.

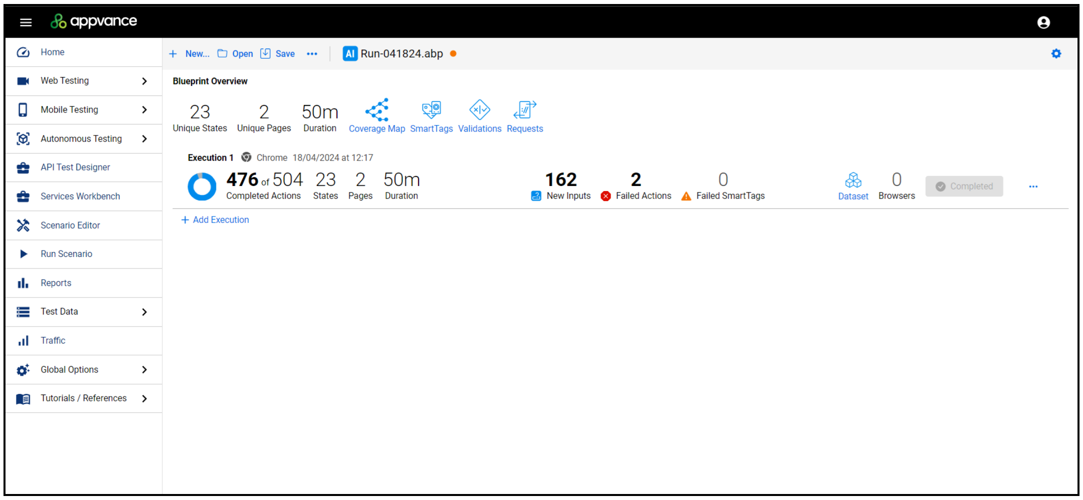


Early on, the primary reason the Blueprint stops is that it has not yet been trained with input values for required values such as the “Username” and “Password” fields for the login page. You can train the AI with the needed information so it can proceed with the Blueprint execution.

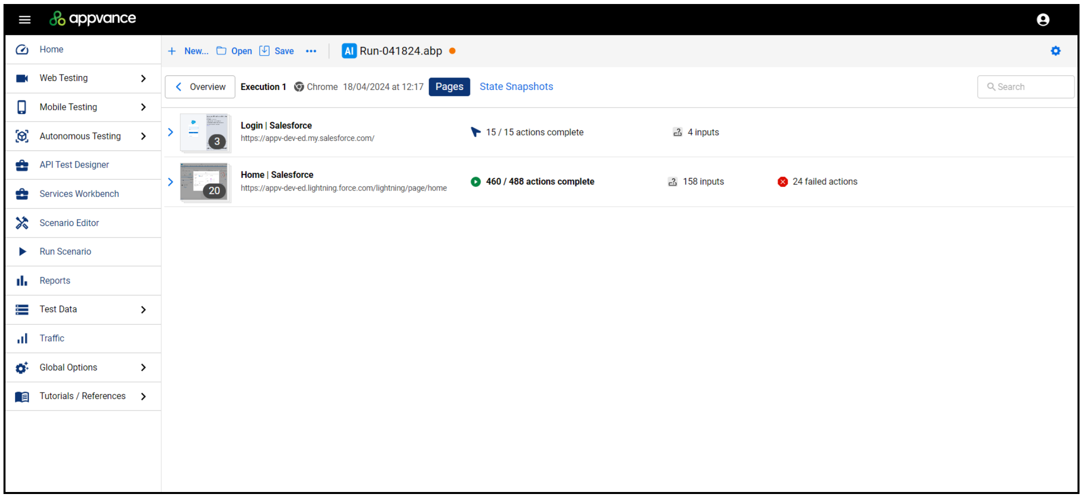
Training the AI

The following details how to train the AI for required inputs such as the “Username” and “Password” fields for the login page.

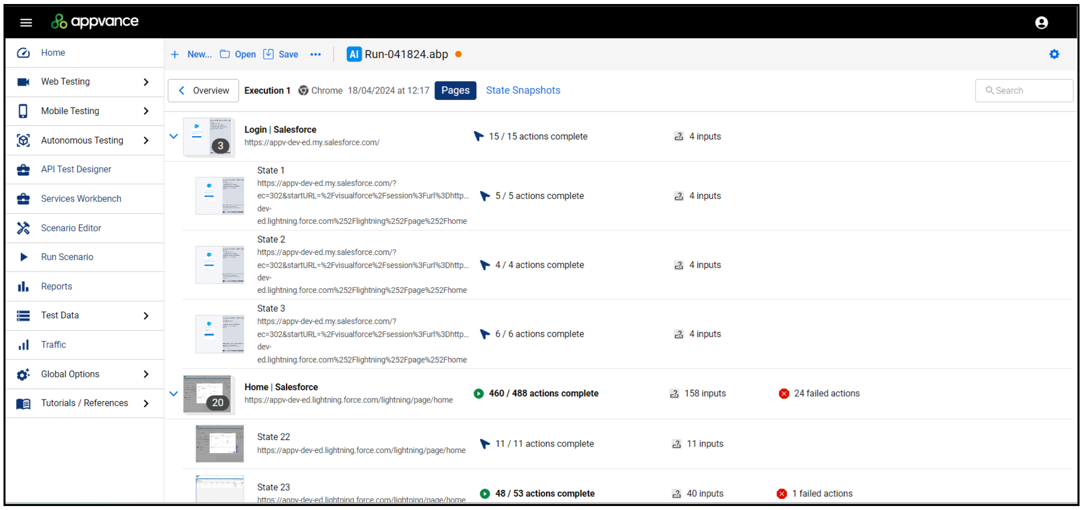
1. **Blueprint Overview** shows that the Blueprint had stopped running after entering the URL and clicking all the elements that are found on the page. The two resulting page states are displayed.



2. Click **Pages** or **State Snapshots** to see the page and page state results.



3. From there you can expand the page results to see more details about the pages and page states.



Execution 1 Chrome 18/04/2024 at 12:17

Login | Salesforce
https://appv-dev-ed.my.salesforce.com/ 15 / 15 actions complete 4 inputs

State 1
https://appv-dev-ed.my.salesforce.com/?ec=302&startURL=%2Fvisualforce%2Fsession%3Furl%3Dhttp... dev-ed.lightning.force.com%252Flightning%252Fpage%252Fhome 5 / 5 actions complete 4 inputs

State 2
https://appv-dev-ed.my.salesforce.com/?ec=302&startURL=%2Fvisualforce%2Fsession%3Furl%3Dhttp... dev-ed.lightning.force.com%252Flightning%252Fpage%252Fhome 4 / 4 actions complete 4 inputs

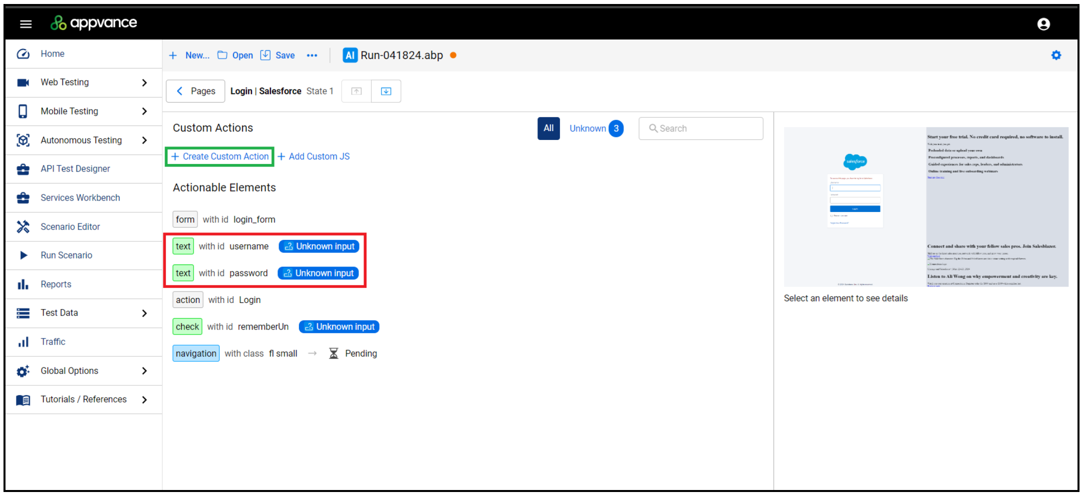
State 3
https://appv-dev-ed.my.salesforce.com/?ec=302&startURL=%2Fvisualforce%2Fsession%3Furl%3Dhttp... dev-ed.lightning.force.com%252Flightning%252Fpage%252Fhome 6 / 6 actions complete 4 inputs

Home | Salesforce
https://appv-dev-ed.lightning.force.com/lightning/page/home 460 / 488 actions complete 158 inputs 24 failed actions

State 22
https://appv-dev-ed.lightning.force.com/lightning/page/home 11 / 11 actions complete 11 inputs

State 23
https://appv-dev-ed.lightning.force.com/lightning/page/home 48 / 53 actions complete 40 inputs 1 failed actions

4. Here you can see that the Blueprint has identified unknown input on the login page for `username` and `password`. Click **Create Custom Action** to begin training the AI how to process those inputs.



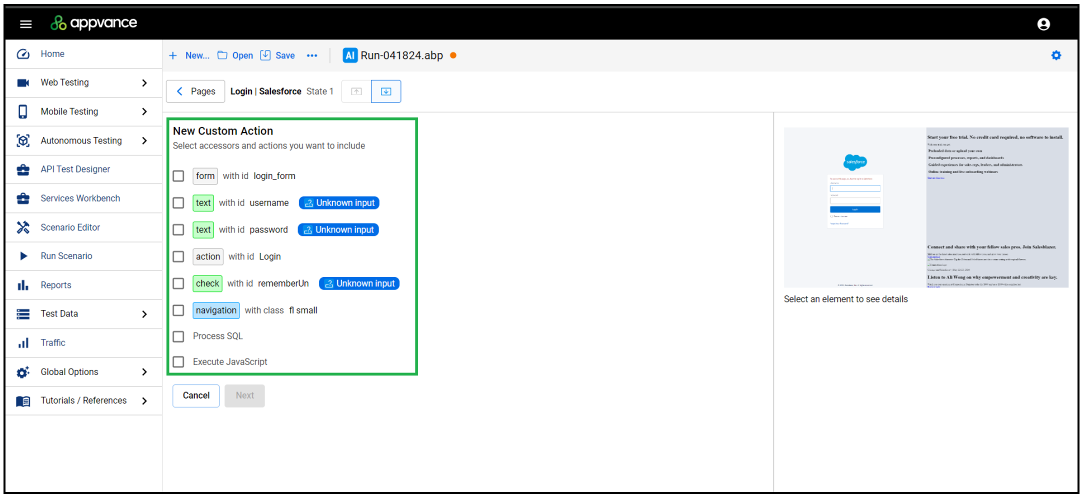
Login | Salesforce State 1

Custom Actions

Actionable Elements

form with id login_form
text with id username Unknown input
text with id password Unknown input
action with id Login
check with id rememberUn Unknown input
navigation with class fl small → Pending

5. The **New Custom Action** section shows which actions for which you can create custom actions.

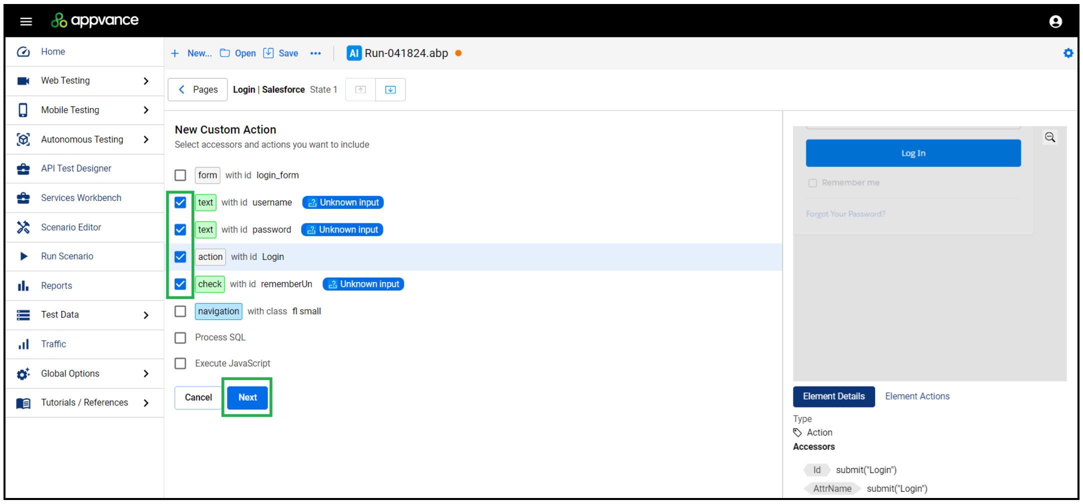


New Custom Action
Select accessors and actions you want to include

- □ form with id login_form
- □ text with id username Unknown input
- □ text with id password Unknown input
- □ action with id Login
- □ check with id rememberUn Unknown input
- □ navigation with class fl small
- □ Process SQL
- □ Execute JavaScript

6. Select the checkboxs for the actions for which you want to create new custom actions. In this example it is for the actions associated with entering the `username` and `password`.

   Click **Next**after making the selections.

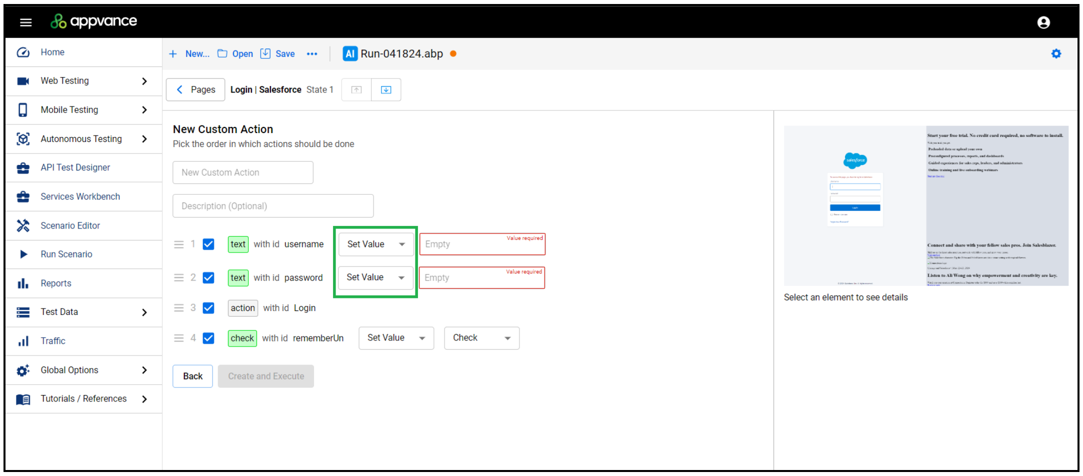


New Custom Action
Select accessors and actions you want to include

- □ form with id login_form
- ☑ text with id username Unknown input
- ☑ text with id password Unknown input
- ☑ action with id Login
- ☑ check with id rememberUn Unknown input
- □ navigation with class fl small
- □ Process SQL
- □ Execute JavaScript

Type
Action
Accessors
Id submit("Login")
AttrName submit("Login")

7. In the action dropdown select **Set Value**.



New Custom Action
Pick the order in which actions should be done

New Custom Action

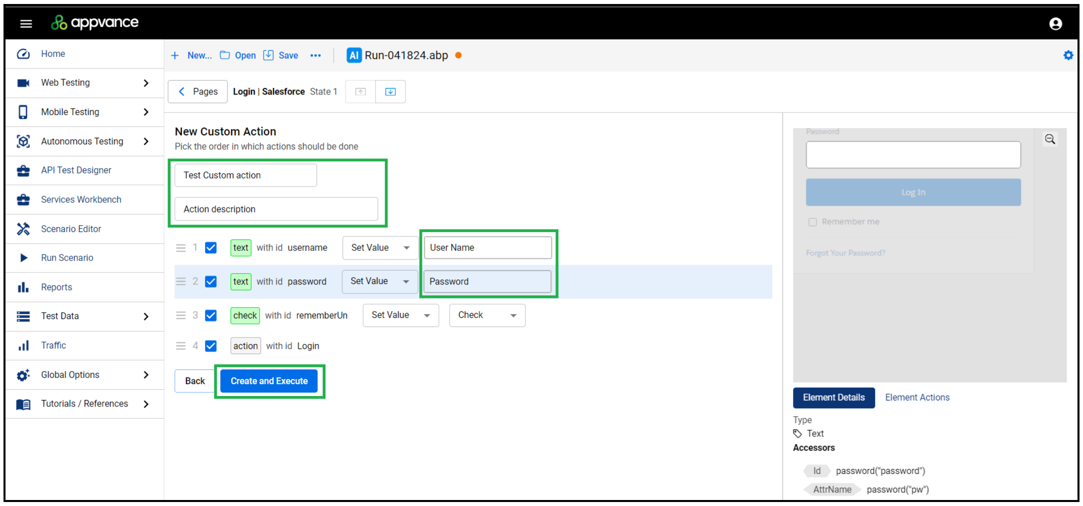
Description (Optional)

1 ☑ text with id username — Set Value — Empty (Value required)
2 ☑ text with id password — Set Value — Empty (Value required)
3 ☑ action with id Login
4 ☑ check with id rememberUn — Set Value — Check

8. Name your new custom action. You can also enter an optional description.

In the input fields, enter the necessary values. If needed you can change the sequence by dragging and dropping the individual actions.

Click **Create and Execute** when finished.



New Custom Action
Pick the order in which actions should be done

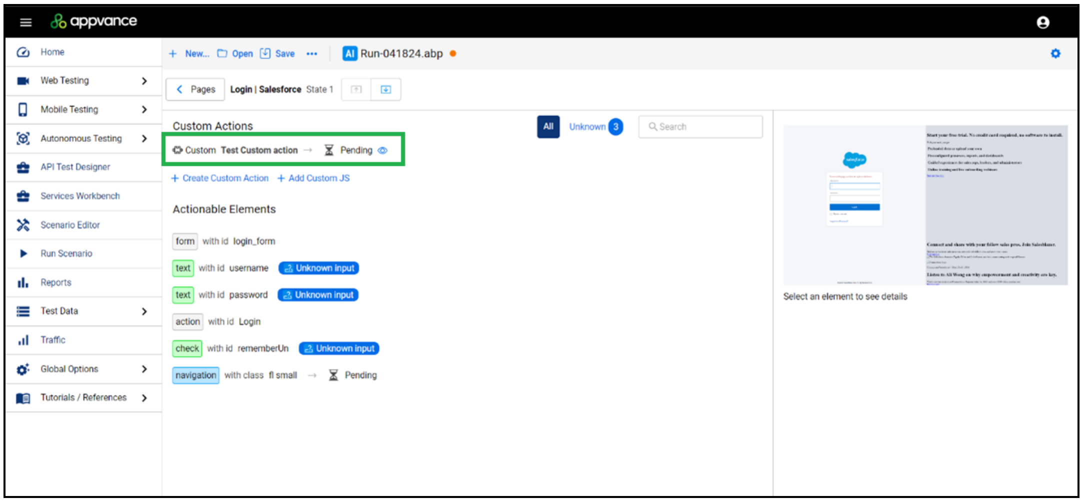
Test Custom action

Action description

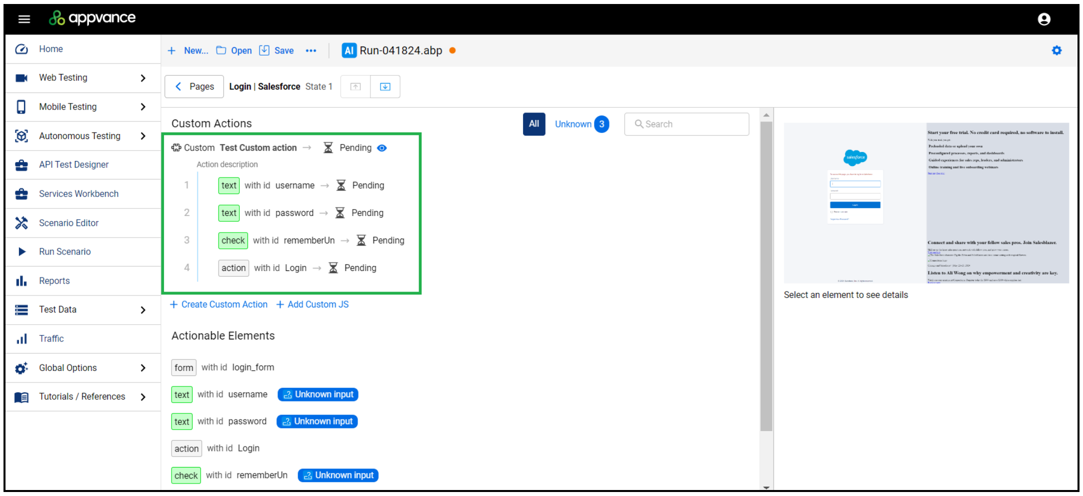
1 ☑ text with id username — Set Value — User Name
2 ☑ text with id password — Set Value — Password
3 ☑ check with id rememberUn — Set Value — Check
4 ☑ action with id Login

Type
Text
Accessors
Id password("password")
AttrName password("pw")

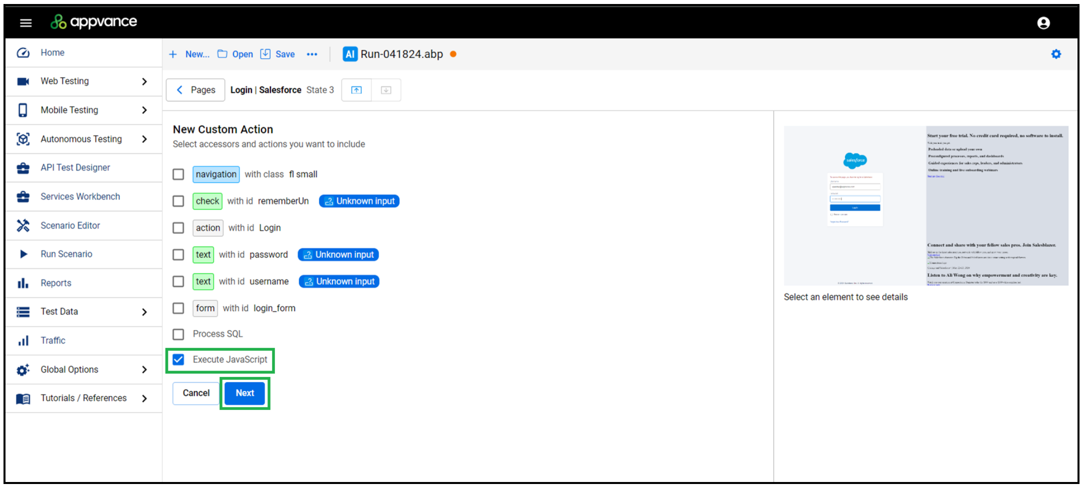
9. Your new custom action shows as **Pending**.



You can see all of the related actions.



10. Click **Add Custom Action** and select **Execute JavaScript**. Click **Next**.

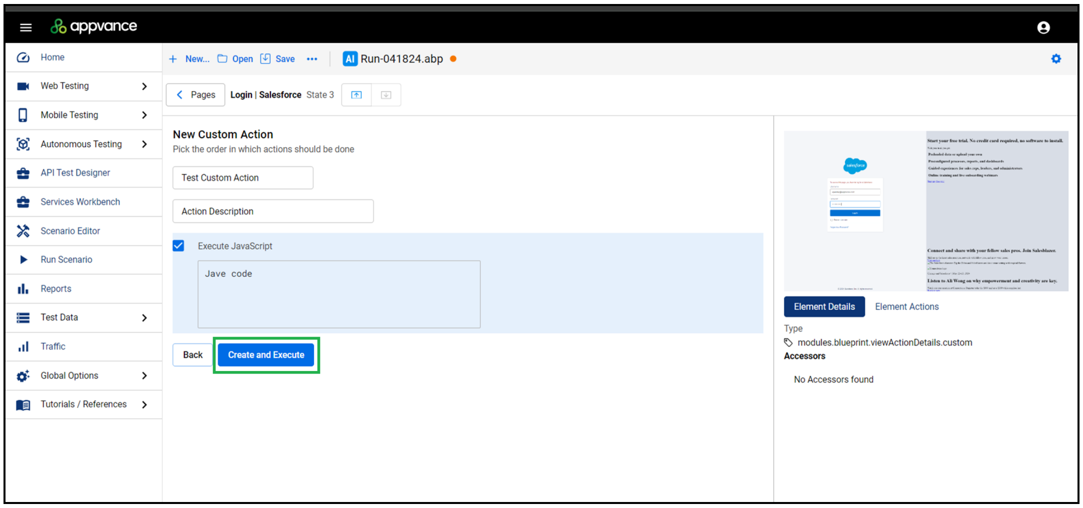


New Custom Action

Select accessors and actions you want to include

- ☐ navigation with class fl small
- ☐ check with id rememberUn — Unknown input
- ☐ action with id Login
- ☐ text with id password — Unknown input
- ☐ text with id username — Unknown input
- ☐ form with id login_form
- ☐ Process SQL
- ☑ Execute JavaScript

11. Enter the custom action name and description and click **Create and Execute**.



New Custom Action

Pick the order in which actions should be done

Test Custom Action

Action Description

☑ Execute JavaScript

```
Jave code
```

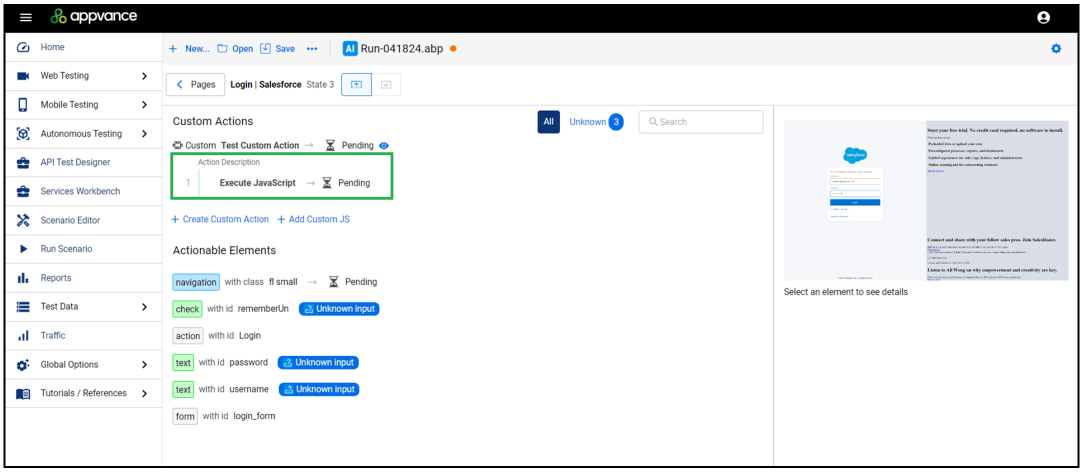
Type

modules.blueprint.viewActionDetails.custom

Accessors

No Accessors found

12. Your new custom action shows as **Pending**.



Custom Actions

Custom Test Custom Action → Pending

Action Description

1 Execute JavaScript → Pending

+ Create Custom Action + Add Custom JS

Actionable Elements

navigation with class fl small → Pending

check with id rememberUn Unknown input

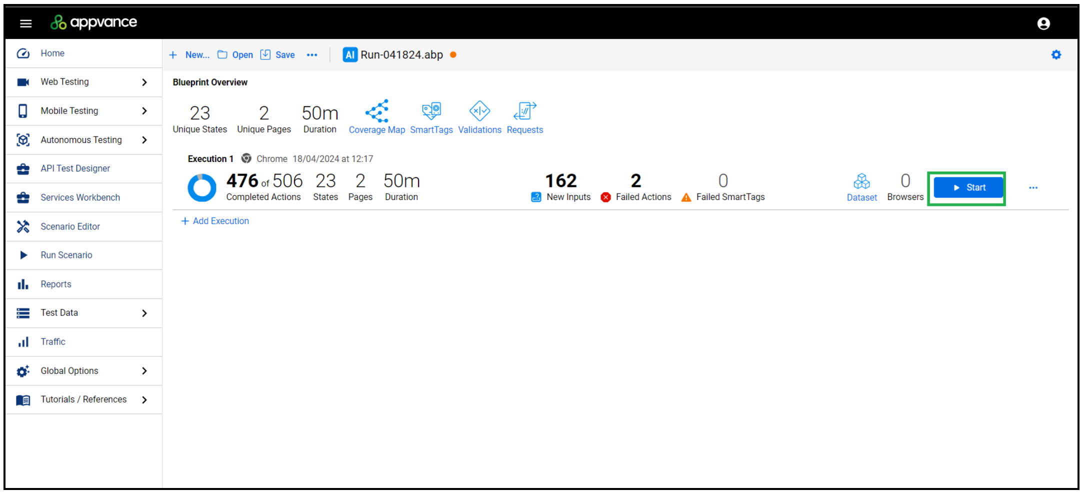
action with id Login

text with id password Unknown input

text with id username Unknown input

form with id login_form

13. Navigate back to the **Blueprint Overview** page and verify that the **Start** button is enabled. This will tell you that the newly added custom actions can be executed by the Blueprint.



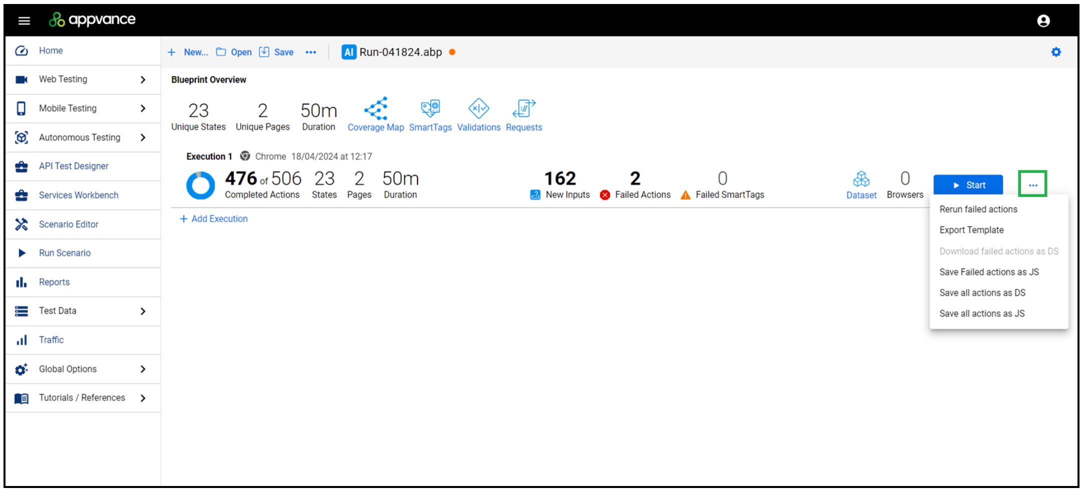
Blueprint Overview

23 Unique States 2 Unique Pages 50m Duration

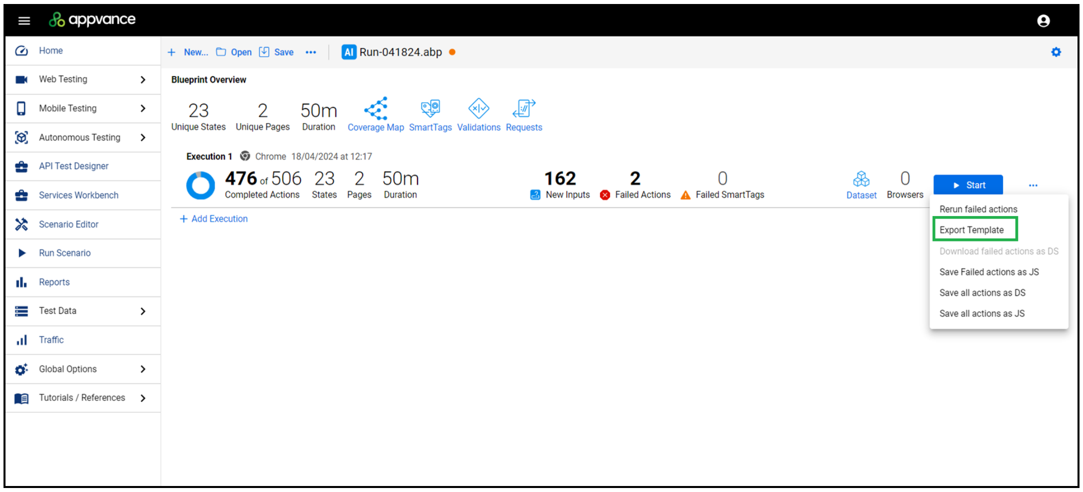
Execution 1 Chrome 18/04/2024 at 12:17

476 of 506 Completed Actions 23 States 2 Pages 50m Duration 162 New Inputs 2 Failed Actions 0 Failed SmartTags 0 Browsers

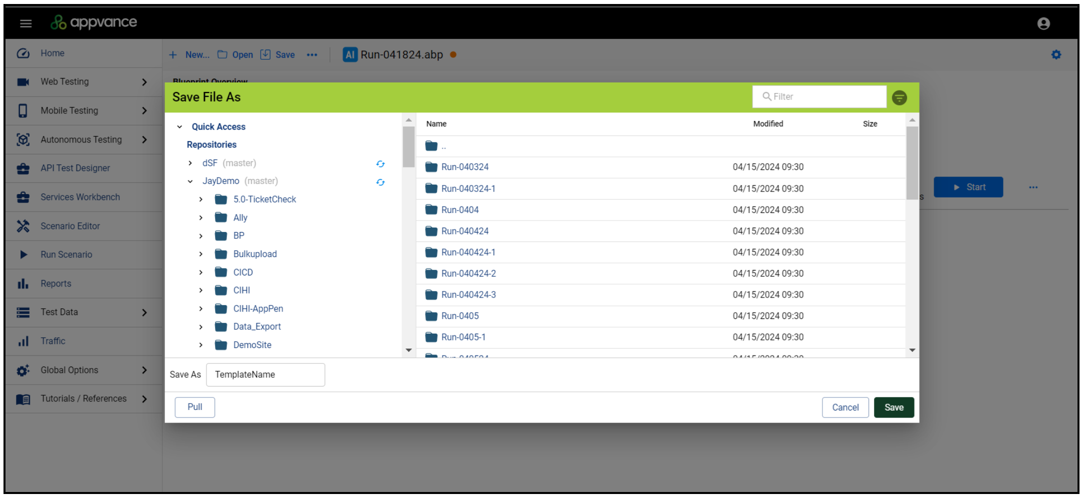
14. Click on additional options menu (the three dots) and check the options that are available.



15. Click **Export Template**.



16. Save the Blueprint template.

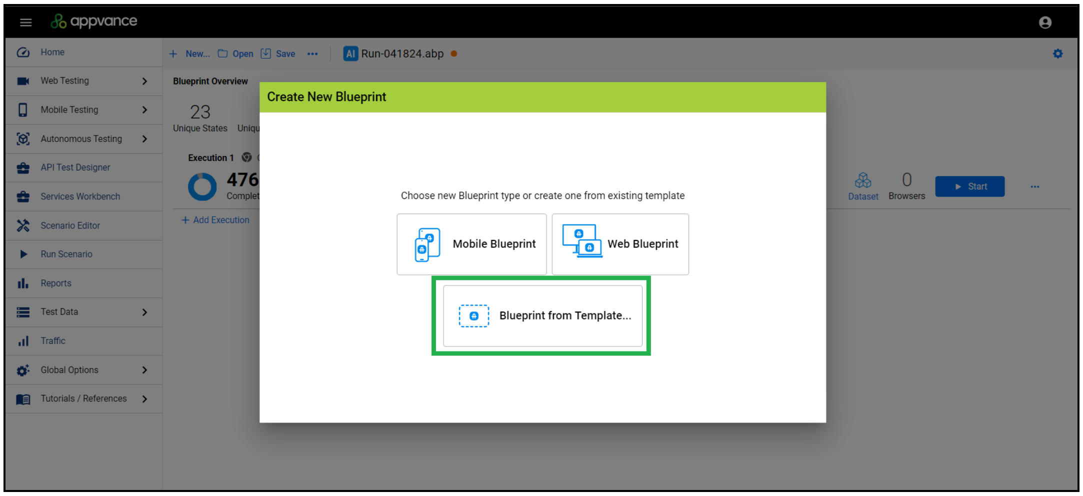


Name	Modified	Size
..		
Run-040324	04/15/2024 09:30	
Run-040324-1	04/15/2024 09:30	
Run-0404	04/15/2024 09:30	
Run-040424	04/15/2024 09:30	
Run-040424-1	04/15/2024 09:30	
Run-040424-2	04/15/2024 09:30	
Run-040424-3	04/15/2024 09:30	
Run-0405	04/15/2024 09:30	
Run-0405-1	04/15/2024 09:30	

Save As: TemplateName

17. Click **New**.

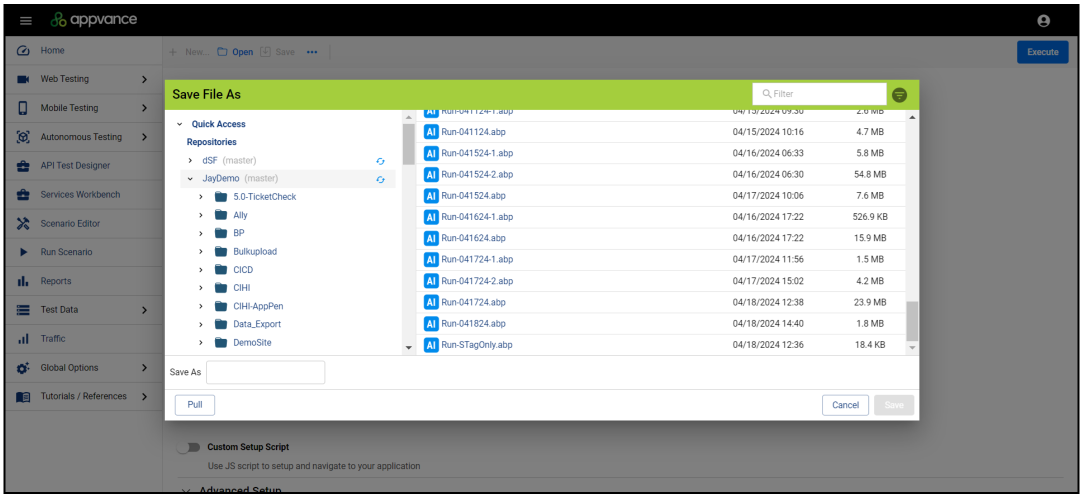
18. Click **Blueprint from Template**.



Choose new Blueprint type or create one from existing template

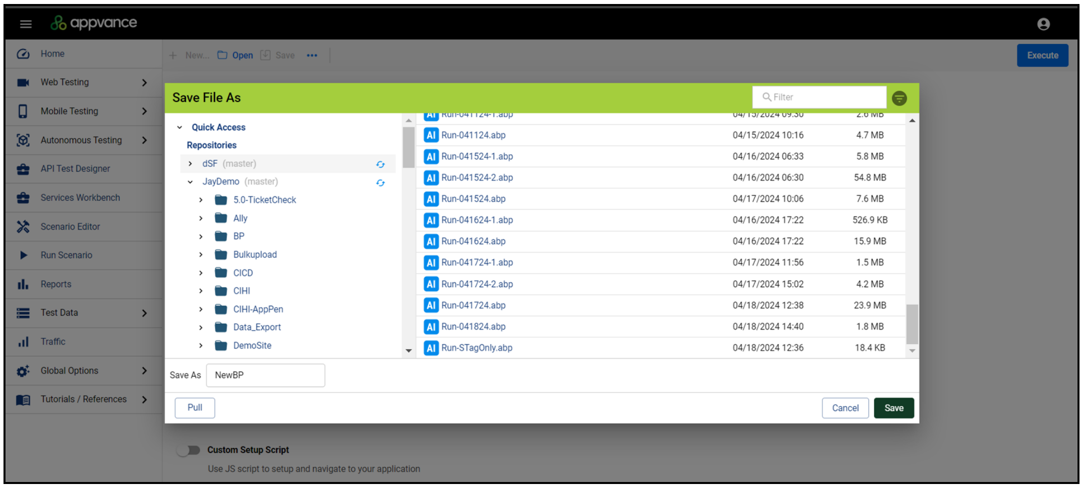
See "Blueprint Templates" on page 62 for more information.

19. Open the Blueprint template file that you just saved.



Run-041124-1.abp	04/15/2024 09:30	2.6 MB
Run-041124.abp	04/15/2024 10:16	4.7 MB
Run-041524-1.abp	04/16/2024 06:33	5.8 MB
Run-041524-2.abp	04/16/2024 06:30	54.8 MB
Run-041524.abp	04/17/2024 10:06	7.6 MB
Run-041624-1.abp	04/16/2024 17:22	526.9 KB
Run-041624.abp	04/16/2024 17:22	15.9 MB
Run-041724-1.abp	04/17/2024 11:56	1.5 MB
Run-041724-2.abp	04/17/2024 15:02	4.2 MB
Run-041724.abp	04/18/2024 12:38	23.9 MB
Run-041824.abp	04/18/2024 14:40	1.8 MB
Run-STagOnly.abp	04/18/2024 12:36	18.4 KB

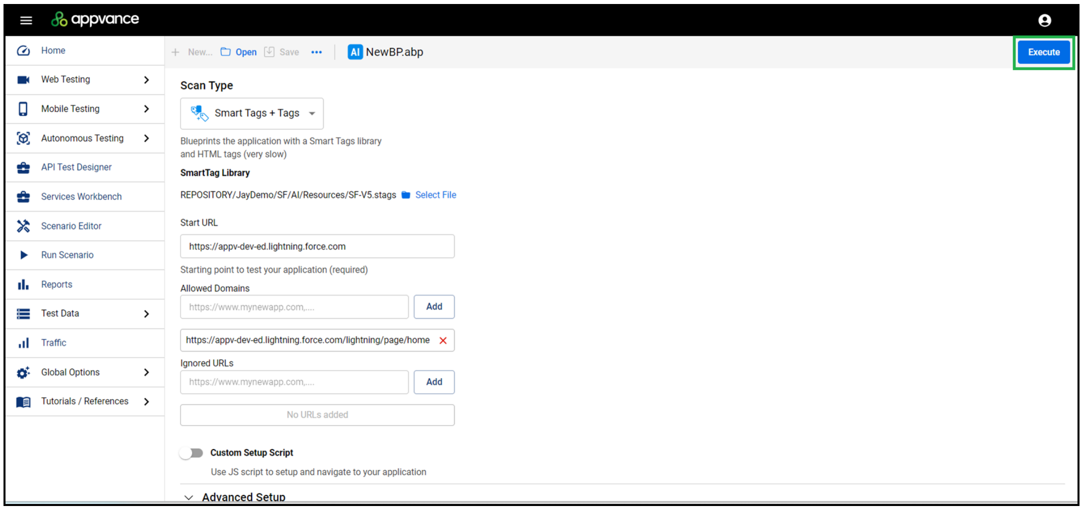
20. After opening the template file you are prompted to save a new blueprint file. Provide the blueprint name and click **Save**.



Run-041124-1.abp	04/15/2024 09:30	2.6 MB
Run-041124.abp	04/15/2024 10:16	4.7 MB
Run-041524-1.abp	04/16/2024 06:33	5.8 MB
Run-041524-2.abp	04/16/2024 06:30	54.8 MB
Run-041524.abp	04/17/2024 10:06	7.6 MB
Run-041624-1.abp	04/16/2024 17:22	526.9 KB
Run-041624.abp	04/16/2024 17:22	15.9 MB
Run-041724-1.abp	04/17/2024 11:56	1.5 MB
Run-041724-2.abp	04/17/2024 15:02	4.2 MB
Run-041724.abp	04/18/2024 12:38	23.9 MB
Run-041824.abp	04/18/2024 14:40	1.8 MB
Run-STagOnly.abp	04/18/2024 12:36	18.4 KB

Save As NewBP

21. The Blueprint Editor opens. Verify the Blueprint settings and details. Click **Execute**.



Scan Type

Smart Tags + Tags

Blueprints the application with a Smart Tags library and HTML tags (very slow)

SmartTag Library

REPOSITORY/JayDemo/SF/AI/Resources/SF-V5.stags

Start URL

https://appv-dev-ed.lightning.force.com

Starting point to test your application (required)

Allowed Domains

https://www.mynewapp.com,...

https://appv-dev-ed.lightning.force.com/lightning/page/home ✕

Ignored URLs

https://www.mynewapp.com,...

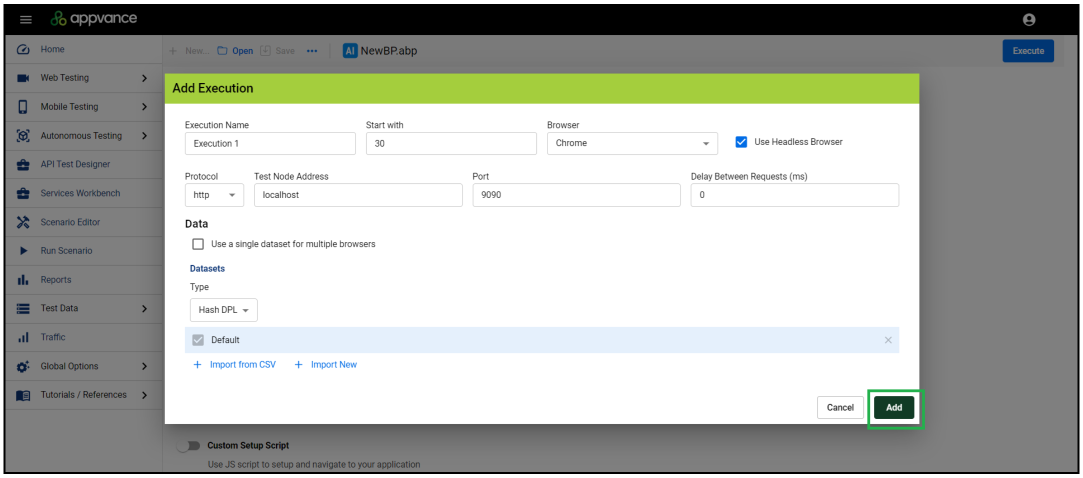
No URLs added

Custom Setup Script

Use JS script to setup and navigate to your application

Advanced Setup

22. Specify the execution details, then click **Add** to start the Blueprint execution.



Add Execution

Execution Name: Execution 1

Start with: 30

Browser: Chrome

☑ Use Headless Browser

Protocol: http

Test Node Address: localhost

Port: 9090

Delay Between Requests (ms): 0

Data

☐ Use a single dataset for multiple browsers

Datasets

Type: Hash DPL

☑ Default

Custom Setup Script

Use JS script to setup and navigate to your application

23. The Blueprint will use the custom actions you created and will be able to login to the Salesforce supplication using the username and passord details that you supplied. This will allow the Blueprint to continue past the point where it stopped in its initial execution. Use this process to train the AI by creating any additional custom actions.

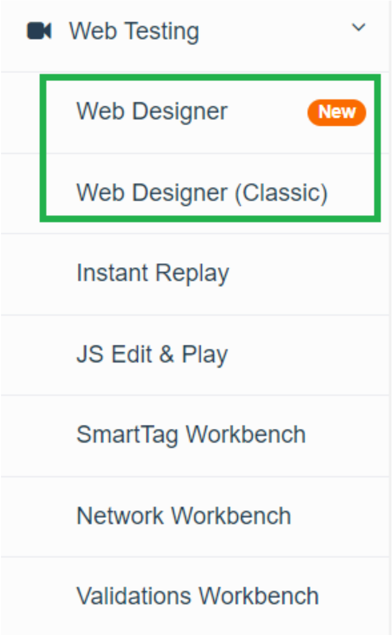
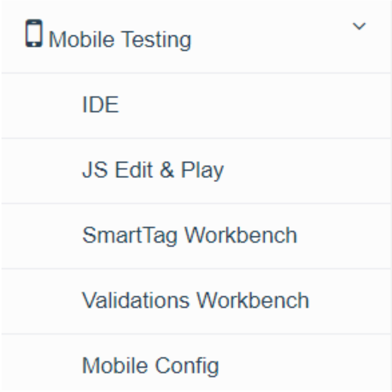
> For more information about working with Blueprints and configuring AI within AIQ, see AIQ AI Implementation Strategy.

Introducing the Web Designer

There are major changes and enhancements to the user interface for creating, maintaining and executing web tests in the 5.0.0 release of AIQ.

> This document is meant as a quick start guide to help acquaint you with the new Web Designer and explain the basic components in the interface.
>
> You can watch a video introducing you to the features and benefits of the new Web Designer.

Here is a comparison between the Web Testing menu options in AIQ 5.0 and previous AIQ releases.

AIQ 5.0.0	AIQ 4.10.x (and older)
 Web Testing Web Designer New Web Designer (Classic) Instant Replay JS Edit & Play SmartTag Workbench Network Workbench Validations Workbench	 Mobile Testing IDE JS Edit & Play SmartTag Workbench Validations Workbench Mobile Config

- **Web Designer** is the new user interface.
- **Web Designer (Classic)** is what was known as IDE in previous AIQ releases.

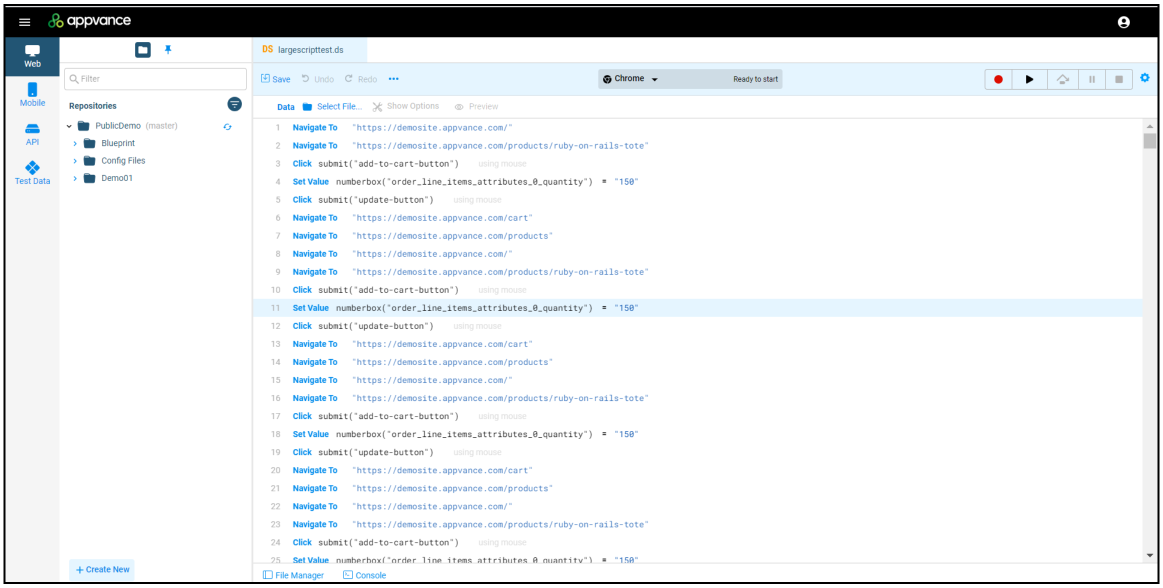
See "Web Testing User Interface Overview" on the facing page for an explanation of the differences between the new and classic interfaces.

> For your convenience, the information about the new Web Designer interface is included in the **AIQ 5.0.0 Release Summary**. This document also contains the AIQ 5.0.0 Release Notes, which includes information about the AIQ 5.0.0 release.

Web Testing User Interface Overview

To access the new Web Designer, from AIQ navigate to **Home** > **Web Testing** > **Web Designer**.

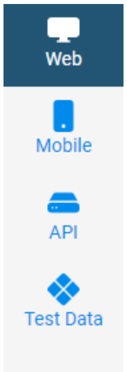
This is an example of the new Web Test Designer interface.

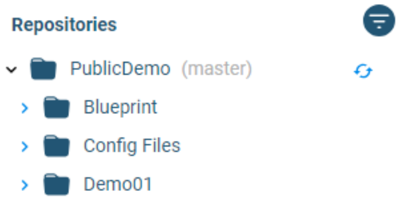
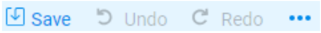
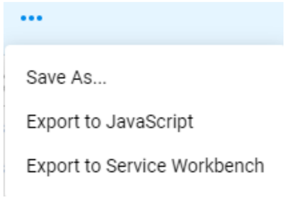


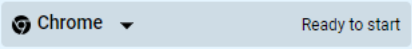
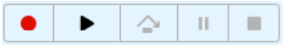
```
Navigate To  "https://demosite.appvance.com/"
Navigate To  "https://demosite.appvance.com/products/ruby-on-rails-tote"
Click  submit("add-to-cart-button")    using mouse
Set Value  numberbox("order_line_items_attributes_0_quantity")  =  "150"
Click  submit("update-button")    using mouse
Navigate To  "https://demosite.appvance.com/cart"
Navigate To  "https://demosite.appvance.com/products"
Navigate To  "https://demosite.appvance.com/"
Navigate To  "https://demosite.appvance.com/products/ruby-on-rails-tote"
Click  submit("add-to-cart-button")    using mouse
Set Value  numberbox("order_line_items_attributes_0_quantity")  =  "150"
Click  submit("update-button")    using mouse
Navigate To  "https://demosite.appvance.com/cart"
Navigate To  "https://demosite.appvance.com/products"
Navigate To  "https://demosite.appvance.com/"
Navigate To  "https://demosite.appvance.com/products/ruby-on-rails-tote"
Click  submit("add-to-cart-button")    using mouse
Set Value  numberbox("order_line_items_attributes_0_quantity")  =  "150"
Click  submit("update-button")    using mouse
Navigate To  "https://demosite.appvance.com/cart"
Navigate To  "https://demosite.appvance.com/products"
Navigate To  "https://demosite.appvance.com/"
Navigate To  "https://demosite.appvance.com/products/ruby-on-rails-tote"
Click  submit("add-to-cart-button")    using mouse
Set Value  numberbox("order_line_items_attributes_0_quantity")  =  "150"
```

Overview of Web Designer Controls and Icons

The following table provides an introduction of the various controls and icons in the new Web Designer interface. The various elements are controls are detailed in the following pages of this document.

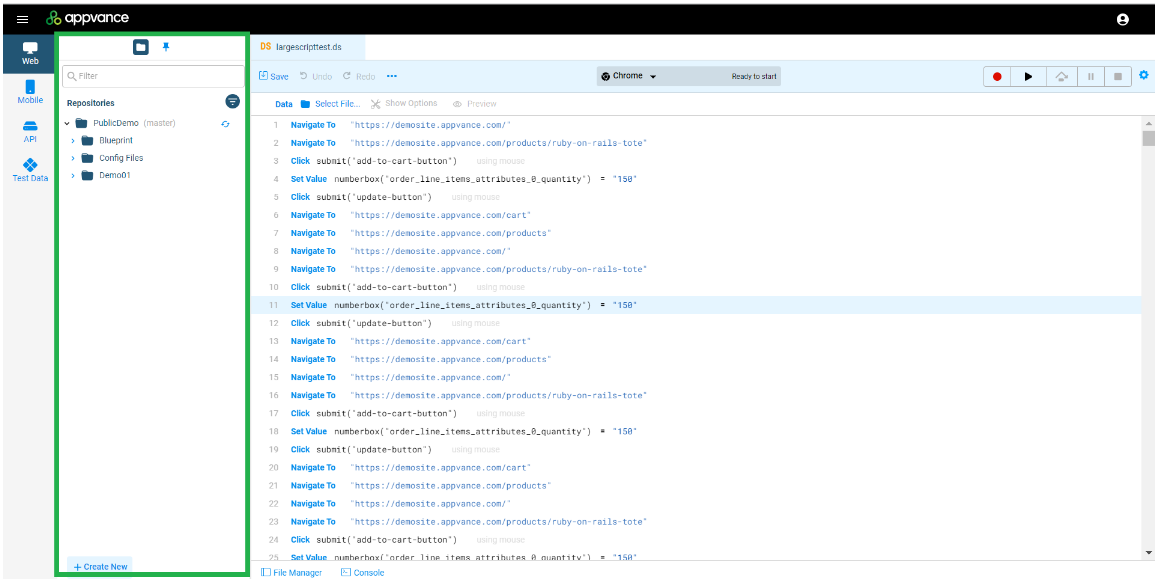
Icon / Control	Function
 Web Mobile API Test Data	Testing Hub. Allows you to easily switch between the Web Designer, Mobile Designer, API Test Designer, and the Test Data console.

Icon / Control	Function
 Repositories PublicDemo (master) Blueprint Config Files Demo01	Repository pane. Displays the contents of your repository. You can sort and filter using various controls.
📁 📌	Repository / Quick Access toggle. Switches the File Manager pane between the repository and quick access views.
 Filter	Filter. Allows you to filter the items displayed in the File Manager.
☰	Sort by Name / Sort by Type. Sorts the contents of the File Manager pane by name or by type.
⟳	Refresh Repository. Refreshes the display of the File Manager pane.
 Save Undo Redo •••	Editing controls menu. • Save • Undo • Redo
 ••• Save As... Export to JavaScript Export to Service Workbench	Additional save and export options menu. • Save as • Export to JavaScript • Export to Services Workbench

Icon / Control	Function
File Manager	File Manager toggle. Shows/hides the File Manager pane.
Log Viewer	Log Viewer toggle. Shows/hides the logs.
	Browser selection control. See "Selecting the Browser" on page 142 for more information.
	Test controls: • Record • Play • Skip Step • Pause • Stop
⚙	Opens the **Test Options** menu. See "Web Designer Test Options" on page 145 for more information.

Repository and File Manager in the Web Designer

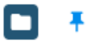
This section details the controls and options available in the Repository and File Manager pane (highlighted) of the Web Designer interface.



```
Navigate To   "https://demosite.appvance.com/"
Navigate To   "https://demosite.appvance.com/products/ruby-on-rails-tote"
Click  submit("add-to-cart-button")     using mouse
Set Value  numberbox("order_line_items_attributes_0_quantity")  =  "150"
Click  submit("update-button")     using mouse
Navigate To   "https://demosite.appvance.com/cart"
Navigate To   "https://demosite.appvance.com/products"
Navigate To   "https://demosite.appvance.com/"
Navigate To   "https://demosite.appvance.com/products/ruby-on-rails-tote"
Click  submit("add-to-cart-button")     using mouse
Set Value  numberbox("order_line_items_attributes_0_quantity")  =  "150"
Click  submit("update-button")     using mouse
Navigate To   "https://demosite.appvance.com/cart"
Navigate To   "https://demosite.appvance.com/products"
Navigate To   "https://demosite.appvance.com/"
Navigate To   "https://demosite.appvance.com/products/ruby-on-rails-tote"
Click  submit("add-to-cart-button")     using mouse
Set Value  numberbox("order_line_items_attributes_0_quantity")  =  "150"
Click  submit("update-button")     using mouse
Navigate To   "https://demosite.appvance.com/cart"
Navigate To   "https://demosite.appvance.com/products"
Navigate To   "https://demosite.appvance.com/"
Navigate To   "https://demosite.appvance.com/products/ruby-on-rails-tote"
Click  submit("add-to-cart-button")     using mouse
Set Value  numberbox("order_line_items_attributes_0_quantity")  =  "150"
```

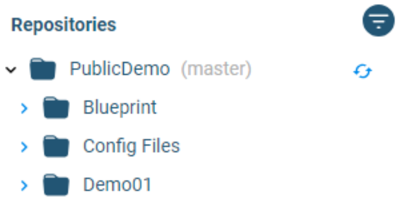
Repository / Quick Access Toggle

Switches the File Manager pane between the repository and quick access views.



Repository Pane

Displays the contents of your repository. You can sort and filter using various controls noted below.



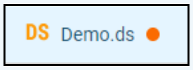
File Type Icons

The following icons display before the file names to indicate the test type.

- DS : Web test file (.ds)
- JS : JavaScript test file (.js)

The following test type icons and files are hidden in the Web Designer because the context is invalid.

- API : API test file (.api)
- 📱 : Mobile test file (.mds)

> There is no auto-save function in the Web Designer. There is a visual indicator, a red dot, that displays to the right of the file name that indicates that you have unsaved changes.
>
> 
>
> DS Demo.ds ●

Filter

Allows you to filter the items displayed based on your search criteria.

Filter

Sort Toggle

Sorts the contents of the Repository File Manager pane by name or by type.

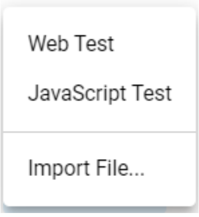
Refresh Repository

Refreshes the display of the File Manager pane.

↻

Create New

Shortcut to create new test files.



Available options are:

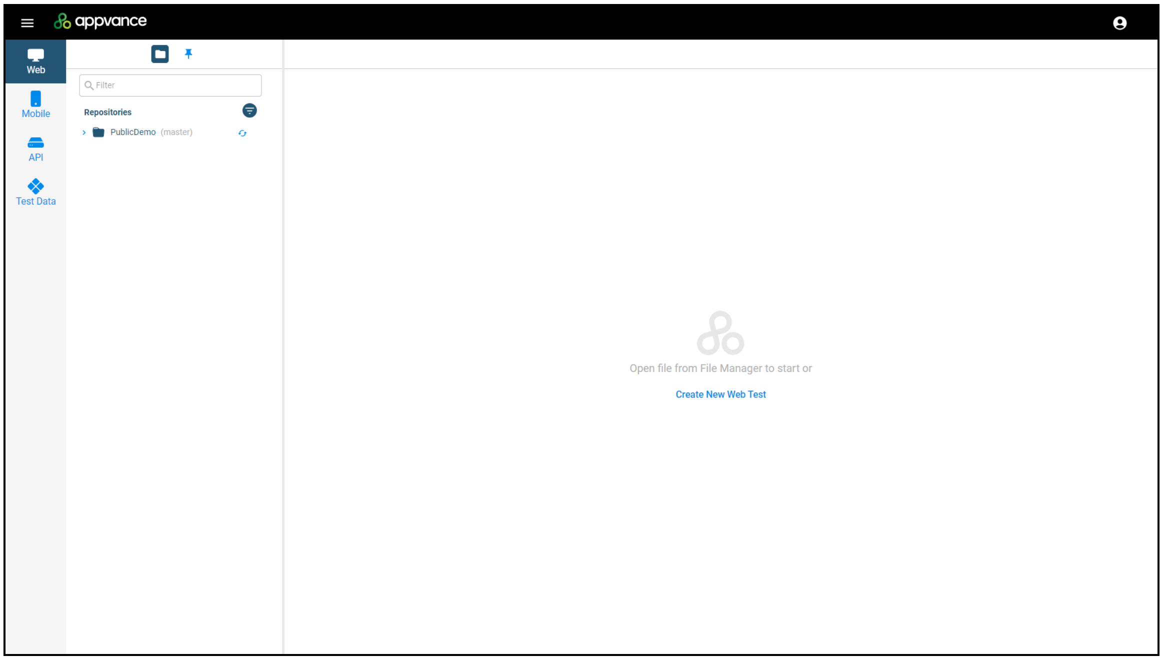
- Web Test
- Javascript Test
- Import File

Initial Screen

The first time you access the new Web Designer interface, you may see a screen similar to the following. Depending on what messages are present in the interface, you may need to perform additional setup or configuration.

Situation 1 : Ready for Use

If your screen looks like this the first time that you open Web Designer it indicates that the minimum configuration requirements have been met and you can begin working immediately.



Open file from File Manager to start or

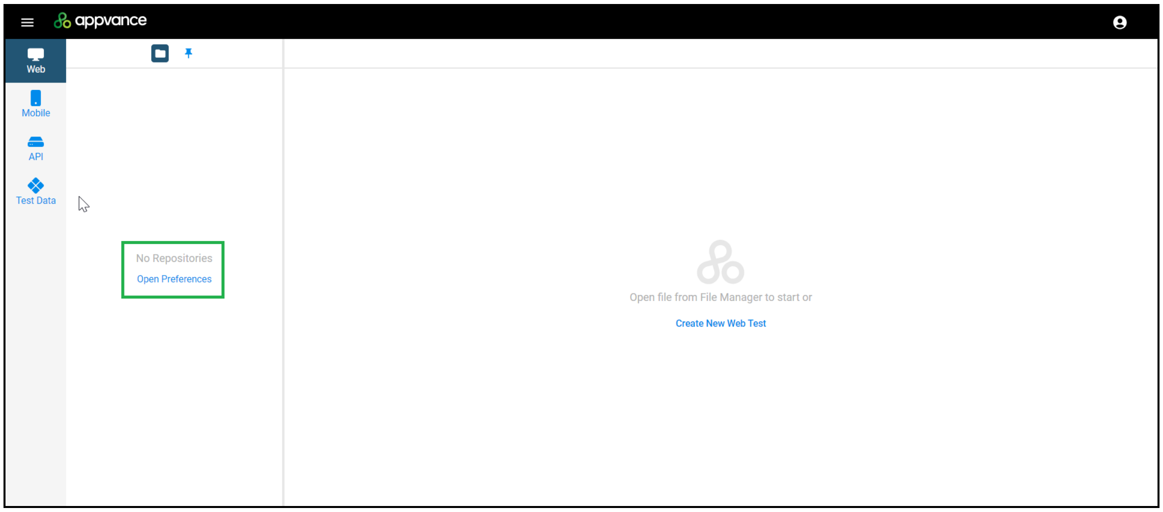
Create New Web Test

Situation 2 : No Repository Configured

If you see the message "No Repositories" (highlighted below) in the File Manager pane it means that you do not have a valid repository configured. You must have a repository configured to store your files. Storing files locally is not supported.

If you are going to use Visual Accessors, you must use a SFTP, FTP, or S3 repository. This is because Images are resource-consuming elements, at the disk, memory, and CPU level, so using an S3 repository will yield the best performance. The images will be saved with the script in your S3 repository which simplifies image management.
Appvance does not recommend using Git to store your images if your are using the Visual Accessor functionality. This is due to Git not being designed to handle large amounts of binary files efficiently.

Click the **Open Preferences** link to configure a repository. See Repository Configuration in Appvance IQ for information on the process.

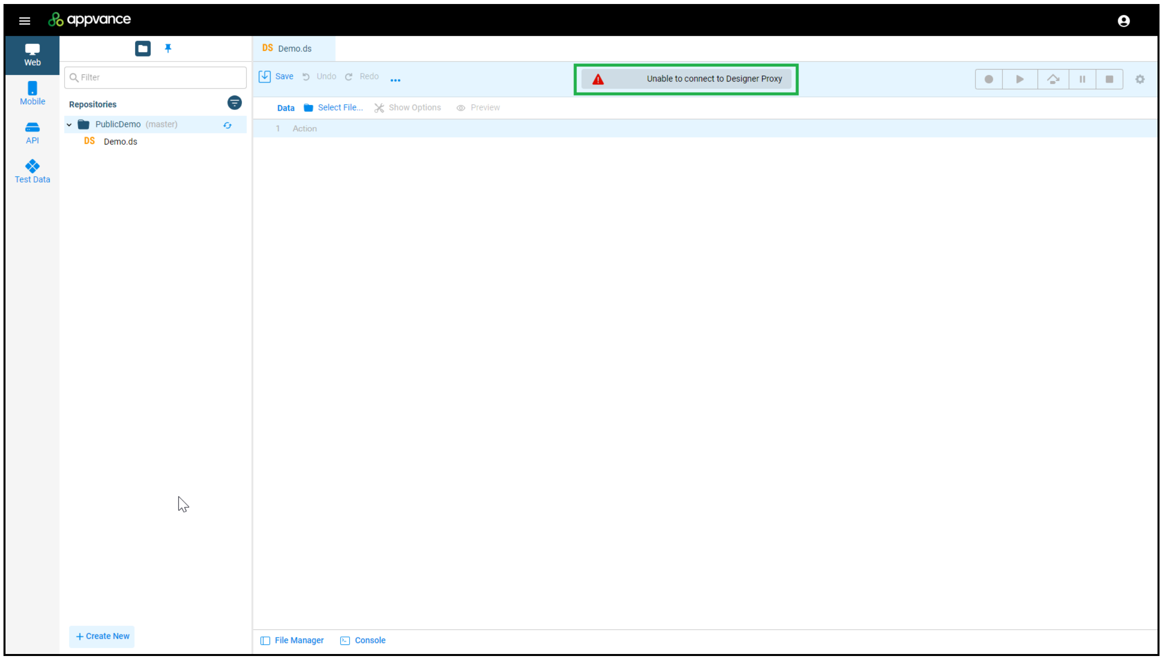


Here is a brief video that shows you the steps to configure a repository and begin using the Web Designer.

Situation 3 : Proxy Mismatch

If you see the message "Unable to Connect to Designer Proxy" (highlighted below) it means that you must update the AIQ proxy before you can continue.

See the section on updating your Designer Proxy for information on the process.

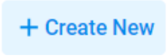


Unable to connect to Designer Proxy

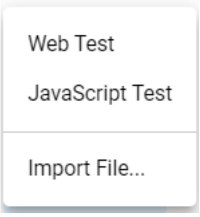
Creating a New Web Test

Here are the basic steps for creating a new web test file.

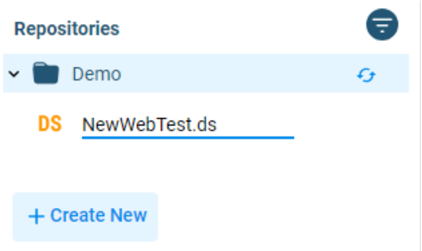
1. Click the **Create New** button in the Repository / File Manager pane



2. Select **Web Test**.



3. Name your new file.



4. You must hit **Enter** or click **Save** before you can edit your new web test file.

5. You can now begin working with your test script.

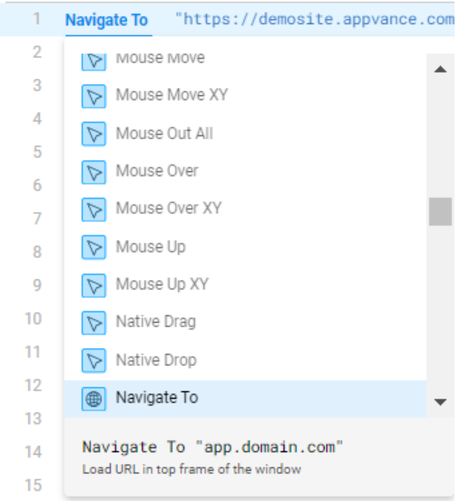
> See the following sections for information about the working with the Web Designer.

Editing Functions and Options

The following functions and options are available when writing or editing scripts. The available functions depend on what column your cursor is in.

Available Actions

The auto assist feature will show you the list of available actions.



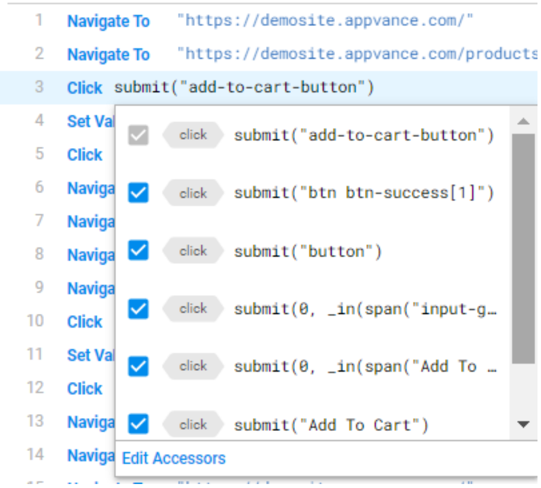
```
Navigate To    "https://demosite.appvance.com
```

```
Navigate To "app.domain.com"
```

Load URL in top frame of the window

- If you click on a column with an existing action, the selected action is highlighted. Information about the action is shown at the bottom of the pop-up.
- If you are adding an action to a new row in your script, right-click to see the available actions pop-up. You can then scroll through the list until you find that action that you want to use. You can also begin typing and the auto complete will show you matching actions.

Editing Accessors



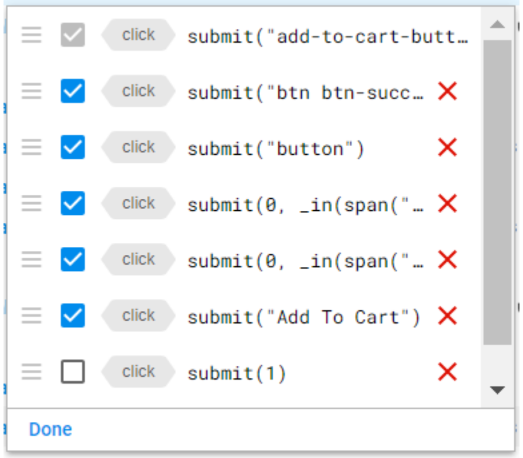
```
1  Navigate To  "https://demosite.appvance.com/"
2  Navigate To  "https://demosite.appvance.com/products
3  Click  submit("add-to-cart-button")
4  Set Va
5  Click
6  Naviga
7  Naviga
8  Naviga
9  Naviga
10 Click
11 Set Va
12 Click
13 Naviga
14 Naviga
```

- ☑ click submit("add-to-cart-button")
- ☑ click submit("btn btn-success[1]")
- ☑ click submit("button")
- ☑ click submit(0, _in(span("input-g…
- ☑ click submit(0, _in(span("Add To …
- ☑ click submit("Add To Cart")

Edit Accessors

- Right-click with the accessor column to see the list of accessors for that step. You can use the individual blue check boxes to toggle individual accessors on and off.

- Click **Edit Accessors** to edit multiple accessors at once.



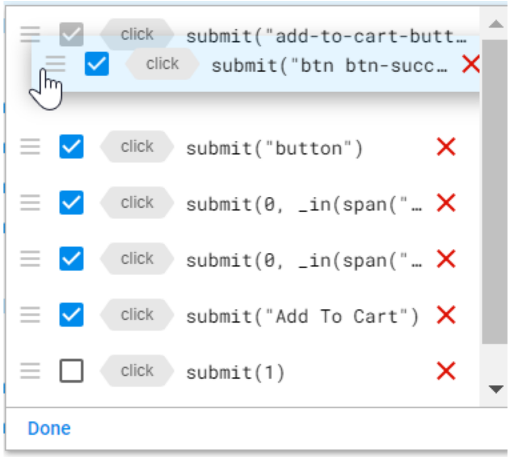
- ☑ click submit("add-to-cart-butt…
- ☑ click submit("btn btn-succ… ✕
- ☑ click submit("button") ✕
- ☑ click submit(0, _in(span("… ✕
- ☑ click submit(0, _in(span("… ✕
- ☑ click submit("Add To Cart") ✕
- ☐ click submit(1) ✕

Done

  - Click the blue checkboxes to toggle individual accessors on or off.

  - Use the red X to delete an accessor from the step.

- You can reorder the list of accessors. Click the three lines on the left, and then you can drag and drop the accessor within the list.



☑ click submit("add-to-cart-butt…

☑ click submit("btn btn-succ… ✕

☑ click submit("button") ✕

☑ click submit(0, _in(span("… ✕

☑ click submit(0, _in(span("… ✕

☑ click submit("Add To Cart") ✕

□ click submit(1) ✕

Done

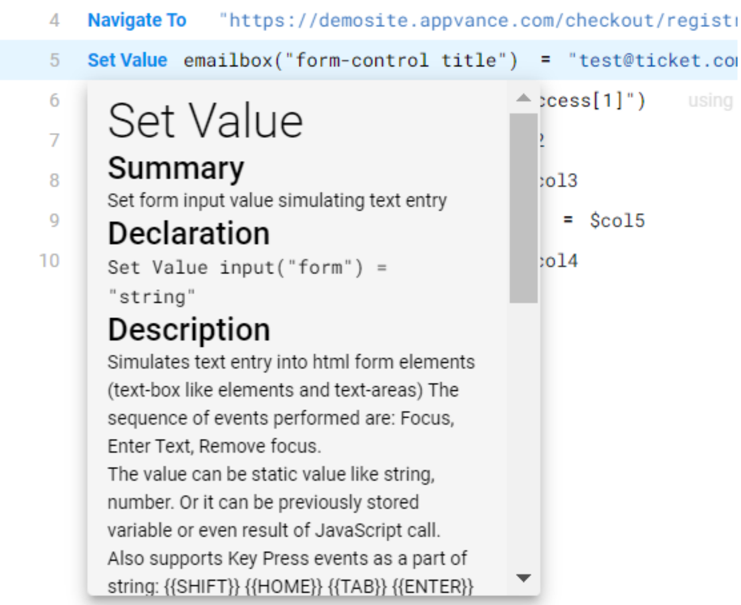
Right Click Menu

The following options are available from the right click menu in the test script editor. Your cursor needs to be in the right most column of the editor to access the right click menu.

- Show Quick Help
- Copy
- Cut
- Paste
- Extract to included DS File
- Add Breakpoint
- Record From Here
- Add Step Below
- Add Step Above
- Add Description
- Disable Step
- Delete

Show Quick Help

This is a new feature in the user interface. It displays information about the selected action.



```
Navigate To  "https://demosite.appvance.com/checkout/regist
Set Value  emailbox("form-control title")  =  "test@ticket.co
                                              :cess[1]")   using
                                              :ol3
                                                 =  $col5
                                              :ol4
```

Set Value

Summary

Set form input value simulating text entry

Declaration

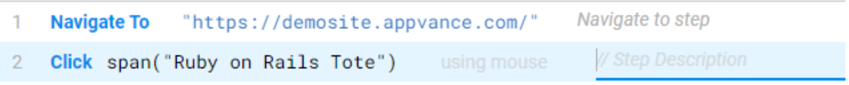
```
Set Value input("form") =
"string"
```

Description

Simulates text entry into html form elements (text-box like elements and text-areas) The sequence of events performed are: Focus, Enter Text, Remove focus.
The value can be static value like string, number. Or it can be previously stored variable or even result of JavaScript call.
Also supports Key Press events as a part of string: {{SHIFT}} {{HOME}} {{TAB}} {{ENTER}}

Add Description

You can add descriptions for any and all steps in your test script. This lets you document the important details such as a specific search operation or a login details. Script descriptions are extremely useful for Appvance Support if they are called upon to help with script debugging.

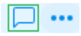


```
Navigate To  "https://demosite.appvance.com/"        Navigate to step
Click  span("Ruby on Rails Tote")   using mouse      // Step Description
```

In all AIQ reports, descriptions will be replaced as the step names.

Show / Hide Descriptions

When a line of the script is highlighted, you can click the Description icon to toggle the display of comments for that line.



Removing Descriptions

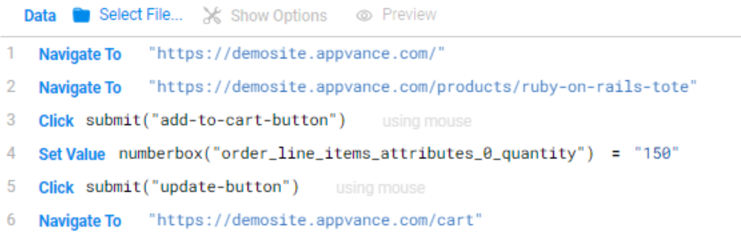
When a line of the script that has a description is highlighted, there is a **Remove Description** option available fro the right-click menu. You can also click the three dots to the right of the Description icon and select **Remove Description** from the menu.

Adding DPL Files

Data Product Libraries (DPLs) deliver operational data to tests at test runtime. For instance, a recorded Appvance Designer Script test of a Web-based login page retrieves the user name and password values from a comma-separated value (CSV) data file as AIQ executes the test.

Tests can use DPL provided data not only as input values but also to validate the application's function.

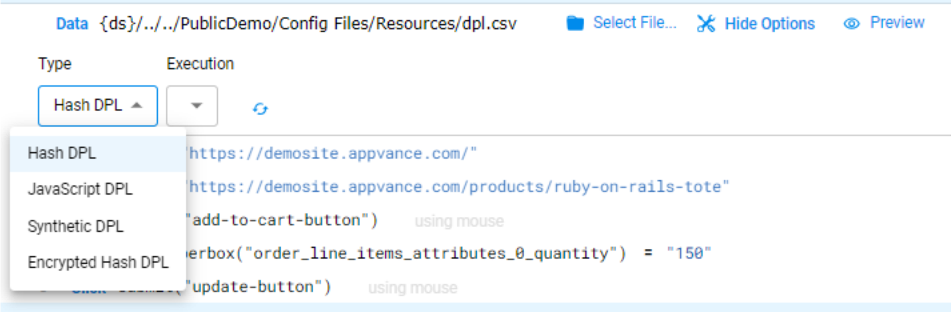
1. In the **Data** area, click **Select File**.



```
Navigate To  "https://demosite.appvance.com/"
Navigate To  "https://demosite.appvance.com/products/ruby-on-rails-tote"
Click  submit("add-to-cart-button")   using mouse
Set Value  numberbox("order_line_items_attributes_0_quantity")  =  "150"
Click  submit("update-button")   using mouse
Navigate To  "https://demosite.appvance.com/cart"
```

2. Browse to your data file and select it.

3. When the file has been added you can select the DPL type.



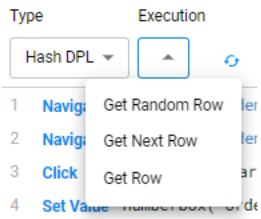
Data {ds}/../../PublicDemo/Config Files/Resources/dpl.csv

Type: Hash DPL

Execution

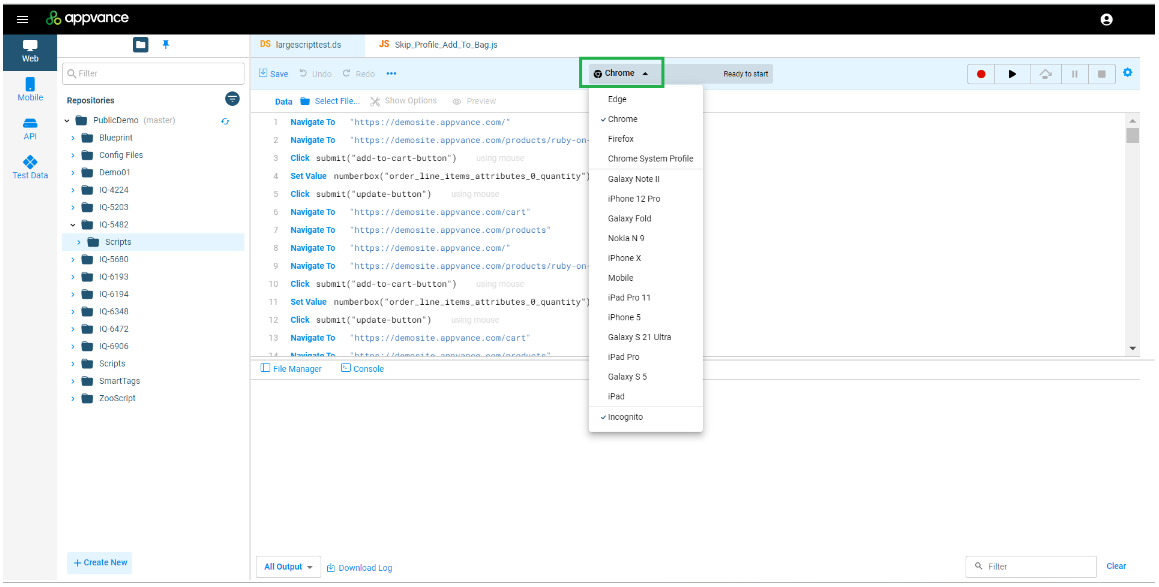
```
"https://demosite.appvance.com/"
"https://demosite.appvance.com/products/ruby-on-rails-tote"
"add-to-cart-button")   using mouse
erbox("order_line_items_attributes_0_quantity")  =  "150"
"update-button")   using mouse
```

4. You can select the Execution options.



Selecting the Browser

Click the browser drop down to select the browser that will be used for the test.



```
Navigate To   "https://demosite.appvance.com/"
Navigate To   "https://demosite.appvance.com/products/ruby-on-
Click  submit("add-to-cart-button")     using mouse
Set Value  numberbox("order_line_items_attributes_0_quantity")
Click  submit("update-button")     using mouse
Navigate To   "https://demosite.appvance.com/cart"
Navigate To   "https://demosite.appvance.com/products"
Navigate To   "https://demosite.appvance.com/"
Navigate To   "https://demosite.appvance.com/products/ruby-on-
Click  submit("add-to-cart-button")     using mouse
Set Value  numberbox("order_line_items_attributes_0_quantity")
Click  submit("update-button")     using mouse
Navigate To   "https://demosite.appvance.com/cart"
Navigate To   "https://demosite.appvance.com/products"
```

Supported Web Browsers

- Edge
- Chrome
- Firefox
- Chrome System Profile

Supported Mobile Browsers

- Galaxy Note II
- iPhone 12 Pro
- Galaxy Fold
- Nokia N 9
- iPhone X
- Mobile
- iPad Pro 11

- iPhone 5
- Galaxy S 21 Ultra
- iPad Pro
- Galaxy S 5
- iPad

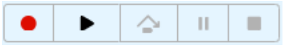
This list is subject to change as new devices are released. This documentation may not reflect the current list. See the UI for the current list of supported mobile devices/browsers.

Supported Modes

- Incognito

Web Designer Test Controls

The following controls are available when executing scripts.

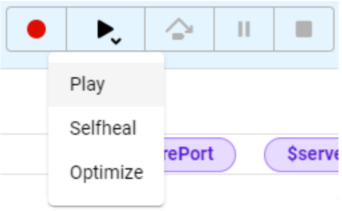


Record

Record a script. Is enabled once a script has been open or saved.

Be sure that you have selected the appropriate browser and other options before you record your script.

Play



There are three options available from the **Play** button:

- **Play** - Used to replay recorded scripts. Is enabled once a script has been open or saved.
- **Selfheal** - Self-healing has the power to remove invalid locators/identifiers, has the power to add any valid locators/identifiers, all it needs is one valid locator to add or remove the other locators.
- **Optimize** - Like self healing, Optimize script is a powerful feature of Appvance IQ's Test designer where it helps you to pick the right accessor with the given list of accessors for any step by rerunning the script for 5 times.

See Script Optimization and Self-Healing for more information on those functions.

Be sure that you have selected the appropriate browser and other options before you play, self-heal or optimize your script.

Step Over

Step Over is a useful tool for debugging the scripts. You can place a breakpoint and use Step Over to replay the script step by step.

Pause

Pause can be used to pause the recording when certain steps are not required to be recorded.

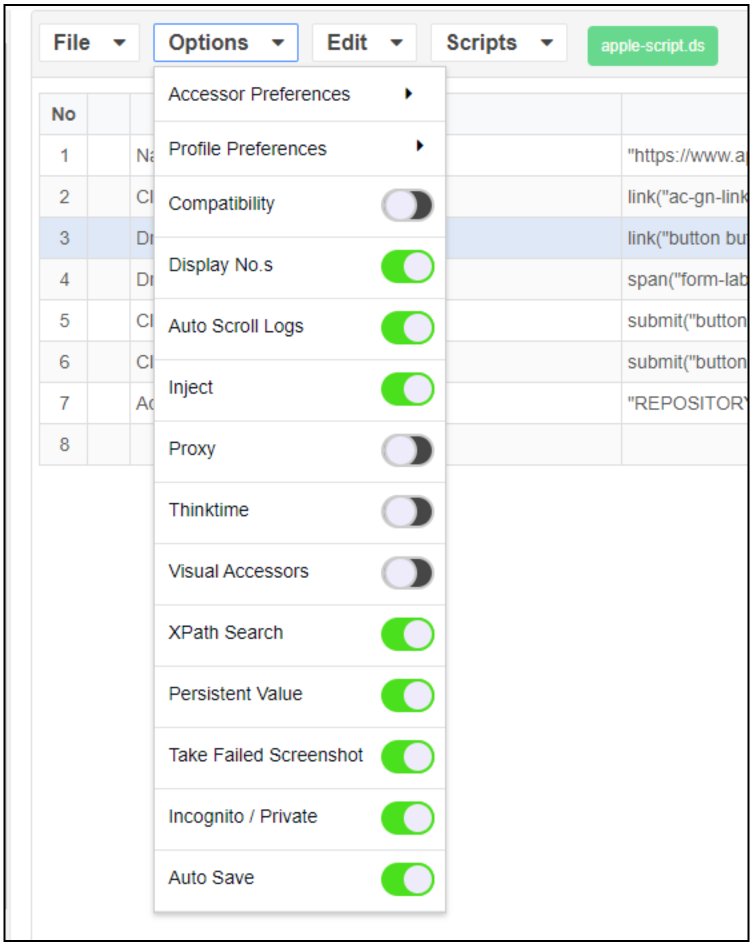
Stop

Stop the recording.

Web Designer Test Options

The implementation of Test Options is different between the Web Designer (Classic) and Web Designer interfaces.

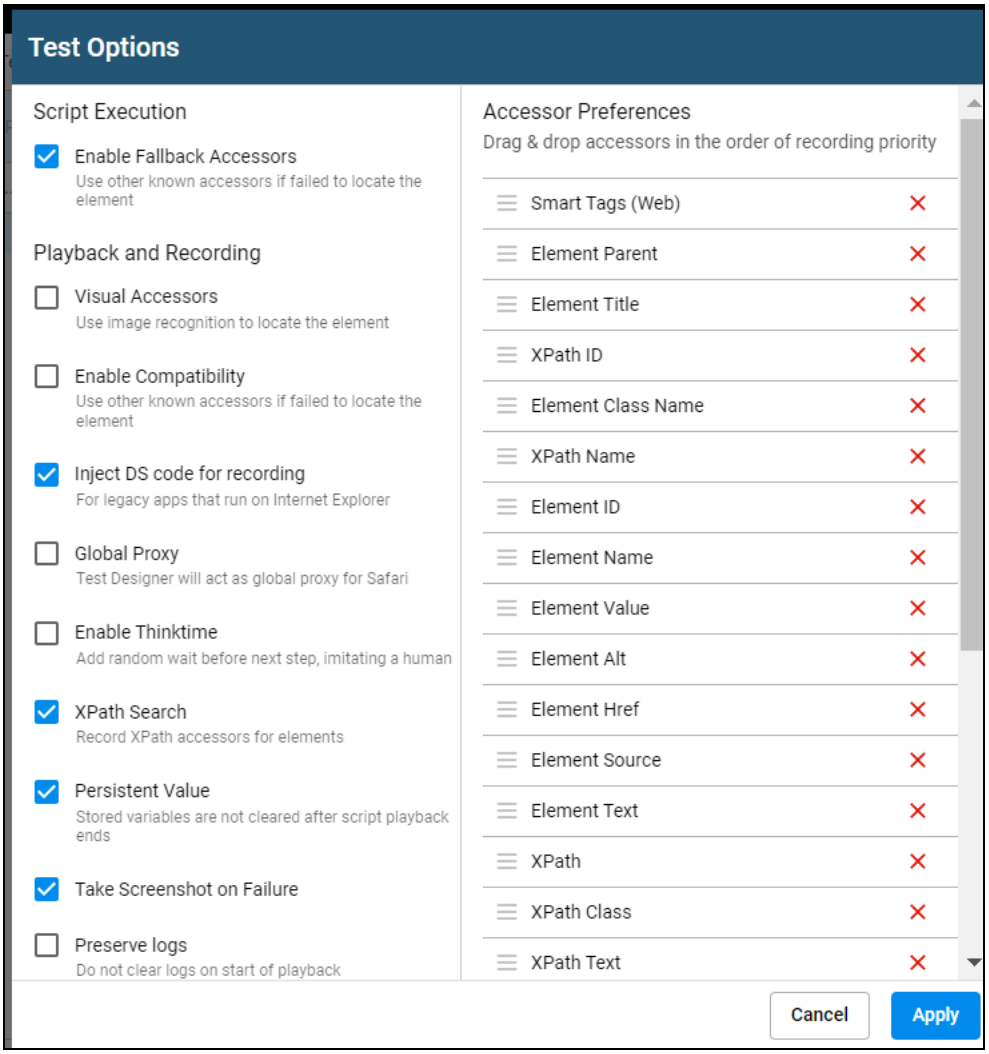
- Test Options available in Web Designer (Classic)



Compatibility ○
Display No.s ●
Auto Scroll Logs ●
Inject ●
Proxy ○
Thinktime ○
Visual Accessors ○
XPath Search ●
Persistent Value ●
Take Failed Screenshot ●
Incognito / Private ●
Auto Save ●

No		
1	Na	"https://www.a
2	Cl	link("ac-gn-link
3	Dr	link("button bu
4	Dr	span("form-lab
5	Cl	submit("button
6	Cl	submit("button
7	Ac	"REPOSITORY
8		

- Test Options available in the new Web Designer interface.



Test Options

Script Execution

☑ Enable Fallback Accessors
Use other known accessors if failed to locate the element

Playback and Recording

☐ Visual Accessors
Use image recognition to locate the element

☐ Enable Compatibility
Use other known accessors if failed to locate the element

☑ Inject DS code for recording
For legacy apps that run on Internet Explorer

☐ Global Proxy
Test Designer will act as global proxy for Safari

☐ Enable Thinktime
Add random wait before next step, imitating a human

☑ XPath Search
Record XPath accessors for elements

☑ Persistent Value
Stored variables are not cleared after script playback ends

☑ Take Screenshot on Failure

☐ Preserve logs
Do not clear logs on start of playback

Accessor Preferences
Drag & drop accessors in the order of recording priority

- Smart Tags (Web) ✕
- Element Parent ✕
- Element Title ✕
- XPath ID ✕
- Element Class Name ✕
- XPath Name ✕
- Element ID ✕
- Element Name ✕
- Element Value ✕
- Element Alt ✕
- Element Href ✕
- Element Source ✕
- Element Text ✕
- XPath ✕
- XPath Class ✕
- XPath Text ✕

See Removed Test Options for information on options that were available in Web Designer (Classic) but are not available in the new Web Designer interface.

Available test options are:

- "Enable Fallback Accessors" on the next page
- "Visual Accessors" on page 149
- "Enable Compatibility" on page 149

- "Inject DS Code for Recording" on the facing page
- "Global Proxy" on the facing page
- "Enable Thinktime" on page 150
- "XPath Search" on page 150
- "Persistent Value" on page 150
- "Take Screenshot on Failure" on page 150
- "Preserve Logs" on page 150
- "Accessor Preferences" on page 151

Script Execution Options

The following options pertain to execution of scripts.

Enable Fallback Accessors

You can enable the Fallback Accessors option so that AIQ keeps test running if an accessor fails or has changed, by automatically choosing other available accessors.

- If Fallback Accessors are enabled, AIQ will try one time for each accessor. If an accessor fails then it will scroll down and again try again each accessor one time and scroll until it reaches the end of the page or reaches maximum number scrolls.

- The scrolling logic will scroll up to ten times, depending on what it locates. If it is not locating elements, it will stop before it reaches ten. If it is locating elements, it will continue until ten.

- If Fallback Accessors disabled, playback will function as it currently does. Tests may fail if an accessor fails or has changed.

Playback and Recording Options

The following options pertain to playback and recording of scripts.

Visual Accessors

Visual Accessors are image based accessors which can be used to enhance your test scripts in situations where traditional code-based accessors will fail. AIQ will capture an image of the website and then crop the surrounding boundaries of the element that you are working on. During playback it takes the image and applies a feature based matching algorithm which creates a pattern from the image and uses that to locate the image on the webpage.

Using a Visual Accessors means that AIQ is not only looking at just the code elements in a page, the AI is also looking at the visual elements on the page.

See Visual Accessors for Web Testing for more detailed usage information.

Enable Compatibility

Specific functionality that is used for legacy apps that run on Internet Explorer, enable it when you need to record legacy IE apps and should be disabled for non-legacy apps which run on Google Chrome and Firefox browsers but are not compulsory. Internet Explorer is no longer a supported browser in AIQ.

Inject DS Code for Recording

The Inject option, when enabled, will inject DS code that will allow you to do UX recording. Web Designer can also record HTTP History calls when it is disabled.

Global Proxy

Enabling proxy makes Web Designer a global proxy, useful for recording on Edge browser on Windows and Mac operating systems, should be enabled when recording with Edge and Safari browsers because they need the global proxy settings and it can be disabled when working with Google Chrome or Firefox browsers but not compulsory to do so.

Enable Thinktime

Think time options can be enabled or disabled depending on the requirements. Enabling think times will add wait for statements in between every action performed from the designer script.

XPath Search

There are several ways to identify an element on the web page, XPath search is one among them. When you record a use case with this option enabled, under the Accessor list you will notice the XPath accessors of that particular element as well. Disabling this option will not add any Xpath references to that element being identified.

Persistent Value

Determines if you want to use persistent values, also known as persistent variables, which are values that are retained in memory between calls.

> Stored variables are not cleared after script playback ends.

Take Screenshot on Failure

Determines if a screenshot is taken even if the script fails before we reach that step where a snapshot was to be captured..

Preserve Logs

Determines if the logs should be cleared every time a script is played. This option is disabled by default.

Removed Test Options

The following options are available in the Web Designer (Classic) interface but not in the new Web Designer implementation.

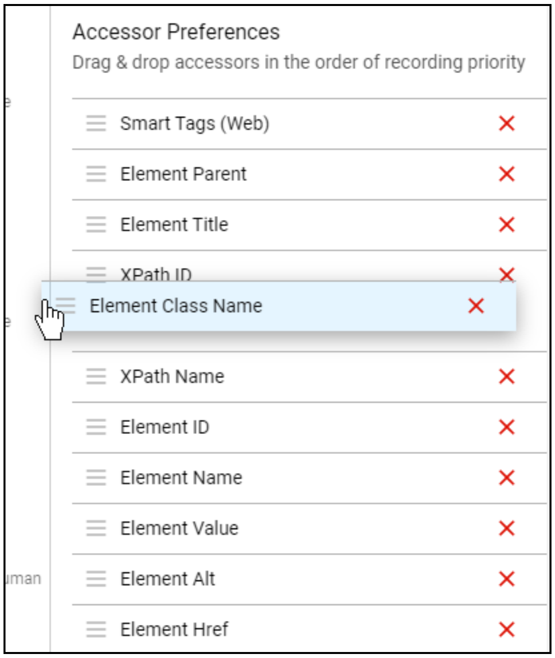
Profile Preferences

The **Profile Preferences** option in the Web Browser (Classic) user interface has been removed from the new Web Designer user interface. The use case for having those roles has become obsolete with the AI improvements in AIQ 5.x.

Accessor Preferences

After setting the fallback accessors and recording a script, if you still feel that the order of accessors that were recorded needs to be changed, you can still drag and drop those accessors and change the order of those accessors based on what you feel has more priority among the list of accessors.

After changing the preferences, you can save the preferences, and based on that recording of the elements from the web page would happen.



Accessor Preferences

Drag & drop accessors in the order of recording priority

- Smart Tags (Web) ✕
- Element Parent ✕
- Element Title ✕
- XPath ID ✕
- Element Class Name ✕
- XPath Name ✕
- Element ID ✕
- Element Name ✕
- Element Value ✕
- Element Alt ✕
- Element Href ✕

From this menu you can specify what has to be recorded, based on what attributes and the elements are on the page.

You can drag and drop these accessor preferences into the order in which you want them. Click Reset Preferences which will go back to defaults.

Above will be the list and the priorities by default, User can drag and drop between these choices and it will be saved as the accessor preferences to record the scripts, there is also an option to Reset the preferences which will go back to the list as mentioned above.

If an element on the web page has all the above-mentioned attributes, then all those will be recorded and listed in the accessor drop-down (But an element will likely have all those attributes).

If for example the accessor preference chosen is Element ID and that element does not have an element ID, then the next choice from the accessor preferences shall be chosen and recorded with that preference, applies to any choice made.

Adding Visual Accessors

Here is an example of adding a Visual Accessor to a test script.

Prerequisites

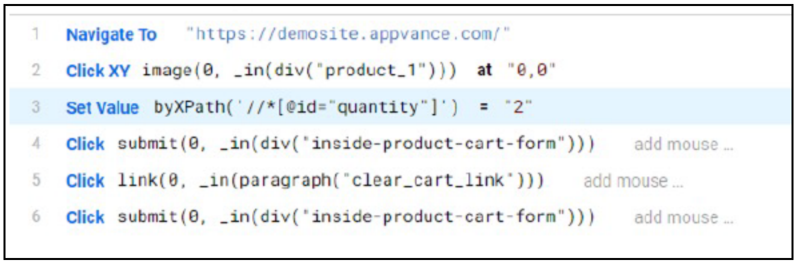
The following options must be enabled:

- Visual Accessors
- XPath Search
- Enable Fallback Accessors
- Shadow DOM

See "Web Designer Test Options" on page 145 for more information on these settings.

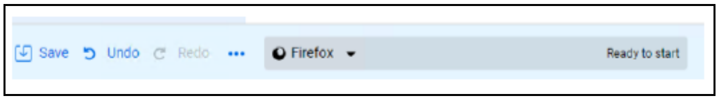
Capture Visual Accessor

1. Right click anywhere in your test script.

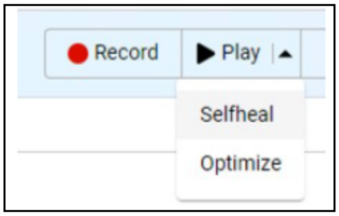


```
Navigate To  "https://demosite.appvance.com/"
Click XY  image(0, _in(div("product_1")))  at  "0,0"
Set Value  byXPath('//*[@id="quantity"]')  =  "2"
Click  submit(0, _in(div("inside-product-cart-form")))    add mouse ...
Click  link(0, _in(paragraph("clear_cart_link")))    add mouse ...
Click  submit(0, _in(div("inside-product-cart-form")))    add mouse ...
```

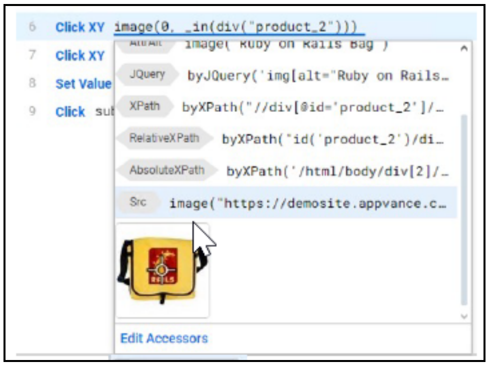
2. Select the browser in the drop down.



3. Select **Optimize**.



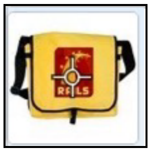
4. Right click in the action and select **Edit Accessor**. Scroll through the list until you see the Visual Accessor.



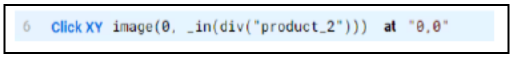
```
Click XY image(0, _in(div("product_2")))
Click XY
Set Value
Click  sul
```

```
AltAlt           image("Ruby on Rails Bag")
JQuery           byJQuery('img[alt="Ruby on Rails…
XPath            byXPath("//div[@id='product_2']/…
RelativeXPath    byXPath("id('product_2')/di…
AbsoluteXPath    byXPath('/html/body/div[2]/…
Src              image("https://demosite.appvance.c…
```

5. Select the Visual Accessor, and check that it is valid.



6. The accessor in your test script is now `image`.



```
Click XY image(0, _in(div("product_2"))) at "0,0"
```

New User Interfaces FAQ

Here are some quick hints and reminders on how to perform certain actions and functions in the new Web Designer user interface.

Can I store my files locally?

No. Storing files locally is no longer supported. You must have a repository configured to store your files.

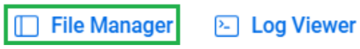
If you do not have a repository configured, Web Designer will prompt you to create one before you can continue working. See Repository Configuration in Appvance IQ for information on the adding a repository.

How do I configure options?

Click the gear icon on the far right of the playback controls panel. See "Web Designer Test Options" on page 145 for more information.

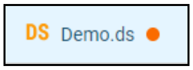
How do I hide the file browser?

In the lower pane click **File Manager** to toggle the file browser panel display.



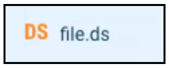
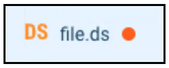
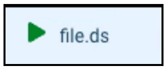
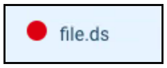
Are files in the Web Designer auto-saved?

No, there is no auto-save function in the Web Designer. There is a visual indicator, a red dot, that displays to the right of the file name that indicates that you have unsaved changes.



How do I tell what is happening with my script?

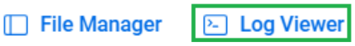
There are some visual indicators on the file tab that show you the current state of your test script. These indicators show you that your script has unsaved changes, is currently being played or currently being recorded.

Icon	Meaning
	Test script is ready, all changes have been saved.
	Test script has unsaved changes.
	Test script is currently playing. The play icon replaces the icon that shows the test script type.
	Test script is currently recording. The recording icon replaces the icon that shows the test script type.

You can play or record a test script with unsaved changes, so you may see the red dot indicating unsaved changes when playing or recording a test script.

How do I see the logs and other information?

Click **Console** and then select the type of information that you want to see.



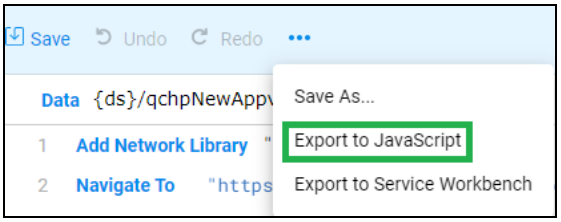
You can select:

- All Output (default)
- Log
- Success
- Info

- Warnings
- Error

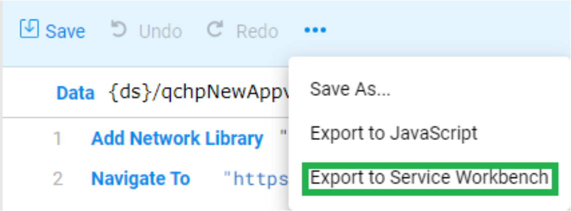
How do I export to Javascript?

Click the additional options control (**...**) and select **Export to Javascript**.



How do I export to Services Workbench?

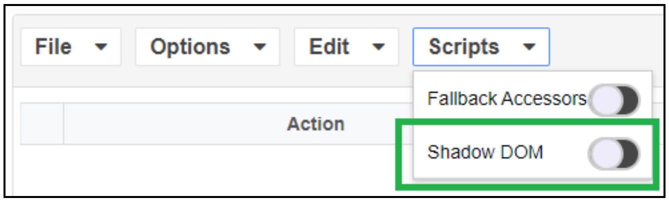
Click the additional options control (**...**) and select **Export to Services Workbench**.



This option is only available in the new Web Designer interface.

What happened to the Shadow DOM setting in the script options?

The Shadow DOM setting that was available in previous AIQ releases has been removed from the user interface. This option is now handled globally in a configuration file.



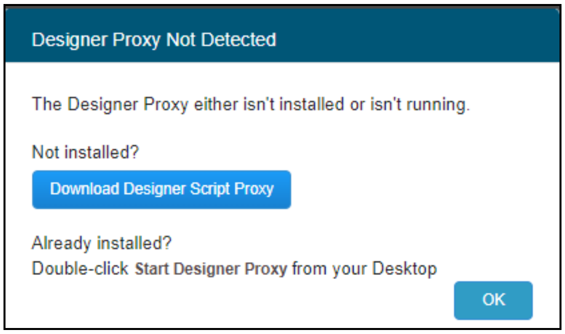
A Shadow DOM setting (`forcedShadowDom`) has been added to the `global.json` file. The default value is `true`. The Shadow DOM setting in the script options in the Web Designer will be ignored if the `forcedShadowDom` is set to true. Even if that option is set to false in the Test Designer IDE, scripts will be run with the option set to true.

Generally speaking, there is no longer any reason for having the Shadow DOM option turned off. Setting it off will not help or prevent any problems or failures. Having it set to off can actually cause failures in instances where a script tries to perform a click action on a element that is under Shadow DOM.

In previous releases, there may have been concerns of some potential performance issues when the setting was on versus the setting being off. Those issues have been addressed.

How do I download an updated Designer Proxy?

In the previous implementation of the web testing interface (now called "Web Designer (Classic)"), the following message would display as soon as you opened the web testing IDE if your version of the Designer Proxy was not compatible.

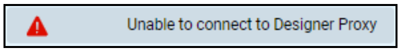

Designer Proxy Not Detected

The Designer Proxy either isn't installed or isn't running.

Not installed?

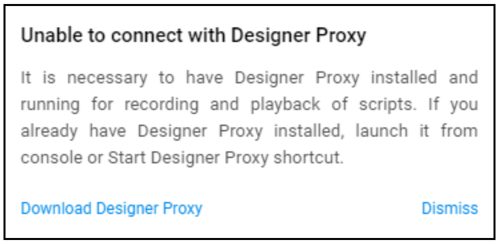
Already installed?
Double-click Start Designer Proxy from your Desktop

In the new Web Designer, you will see the following message in the row of controls in place of the browser selection control.



Unable to connect to Designer Proxy

Click the warning message to launch this dialog. Click the **Download Designer Proxy** link to begin the update process.



Unable to connect with Designer Proxy

It is necessary to have Designer Proxy installed and running for recording and playback of scripts. If you already have Designer Proxy installed, launch it from console or Start Designer Proxy shortcut.

See Upgrading Appvance Test Designer Client for more information and instructions.

Working with Include Files

Extract to included DS file

Why are there two Designer Proxies installed on my computer?

You may notice that you have two separate Designer Proxies installed in different locations.

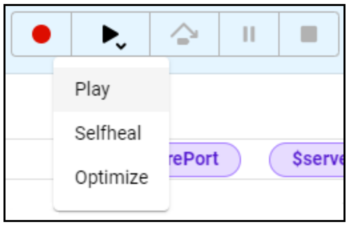
- The Designer Proxy for the new Web Designer implementation is installed at `C:\Users\<user>\WebDesigner`.

- The Designer Proxy for the pre-AIQ 5.0 implementation was installed at

`C:\Users\<user>\TestDesigner`. This instance of the Designer Proxy is not used by AIQ 5.0,

Where do I select to self-heal or optimize a script?

The script self-healing and script optimize options are now sub-options of the **Play** function.



In previous releases and in the current Web Designer (Classic) interface, these options are available from the right-click menu.

For more information see:

- "Web Designer Test Controls" on page 144
- Script Optimization and Self-Healing

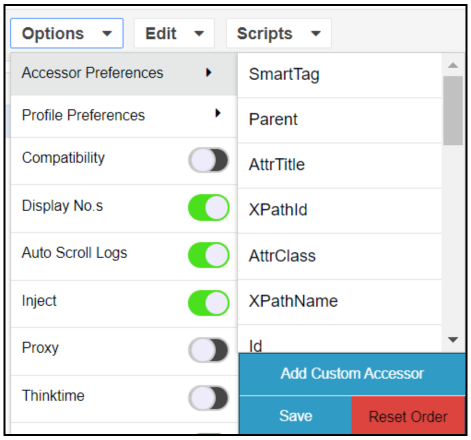
Mobile Testing Enhancements

The following enhancements have been made to the Mobile Designer in AIQ release 5.0.0.

- "Accessor Preferences for Mobile" below
- "After Set Value Option" on the next page
- "Zoom In and Zoom Out" on page 165
- "Zoom In and Zoom Out" on page 165
- "“Go Back” Button" on page 168
- "LambdaTest in Mobile Configuration Options" on page 168

Accessor Preferences for Mobile

From this menu you can specify what has to be recorded, based on what attributes and the elements are on the page.



Compatibility ○
Display No.s ●
Auto Scroll Logs ●
Inject ●
Proxy ○
Thinktime ○

After setting the fallback accessors and recording a script, if you still feel that the order of accessors that were recorded needs to be changed, you can still drag and drop those accessors and change the order of those accessors based on what you feel has more priority among the list of accessors.

After changing the preferences, you can save the preferences, and based on that recording of the elements from the web page would happen.

From this menu you can specify what has to be recorded, based on what attributes and the elements are on the page.

You can drag and drop these accessor preferences into the order in which you want them. Click **Reset Order** which will go back to defaults.

Above will be the list and the priorities by default, You can drag and drop between these choices and it will be saved as the accessor preferences to record the scripts, there is also an option to Reset the preferences which will go back to the list as mentioned above.

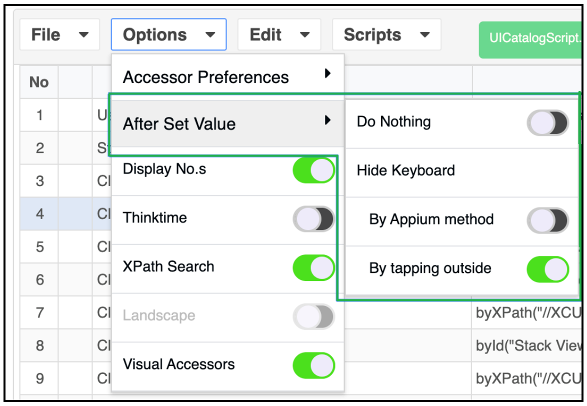
If an element on the web page has all the above-mentioned attributes, then all those will be recorded and listed in the accessor drop-down, but an element will likely have all those attributes.

If for example the accessor preference chosen is Element ID and that element does not have an element ID, then the next choice from the accessor preferences shall be chosen and recorded with that preference, applies to any choice made.

After Set Value Option

This option allows you to tell AIQ what to do after using the `setValue` instruction.

> This option was added in the AIQ 5.0.0 release.



Display No.s ●
Thinktime ○
XPath Search ●
Landscape ○
Visual Accessors ●

Do Nothing ○
Hide Keyboard
By Appium method ○
By tapping outside ●

There are three options:

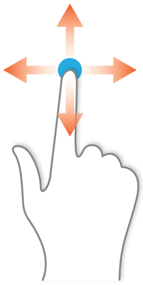
- **Do Nothing** - AIQ won’t perform any actions after setting a value. The keyboard would remain visible after that.

- **Hide Keyboard** - By Appium method : Default option. AIQ will use the default Appium driver method to hide the keyboard. Note, for some apps this could trigger the action for the Done / Enter button of the keyboard, depending on how the app was implemented.

- **Hide Keyboard** - By tapping outside : AIQ will perform a click in a margin of the screen in order to hide the keyboard. This would only dismiss the keyboard if the app implementation allows that behavior.

Drag

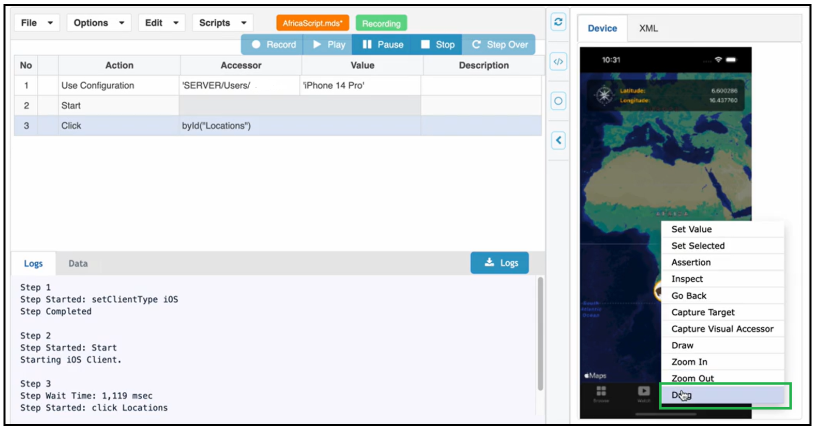
You can perform Drag functions. These function mirrors users performing the pinch and zoom gestures. The following example shows this function in Google Maps.



This was added in the AIQ 5.0.0 release.

1. In the Device tab, select a point.

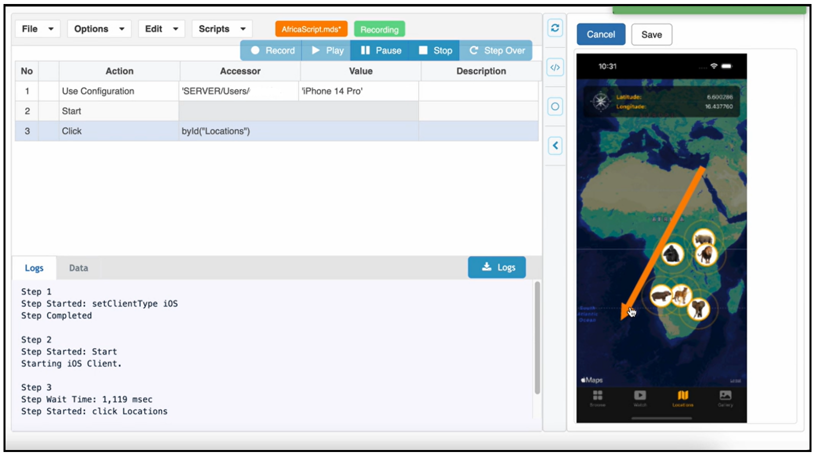
2. Right click and select **Drag**.



No		Action	Accessor	Value	Description
1		Use Configuration	'SERVER/Users/	'iPhone 14 Pro'	
2		Start			
3		Click	byId("Locations")		

```
Step 1
Step Started: setClientType iOS
Step Completed

Step 2
Step Started: Start
Starting iOS Client.

Step 3
Step Wait Time: 1,119 msec
Step Started: click Locations
```

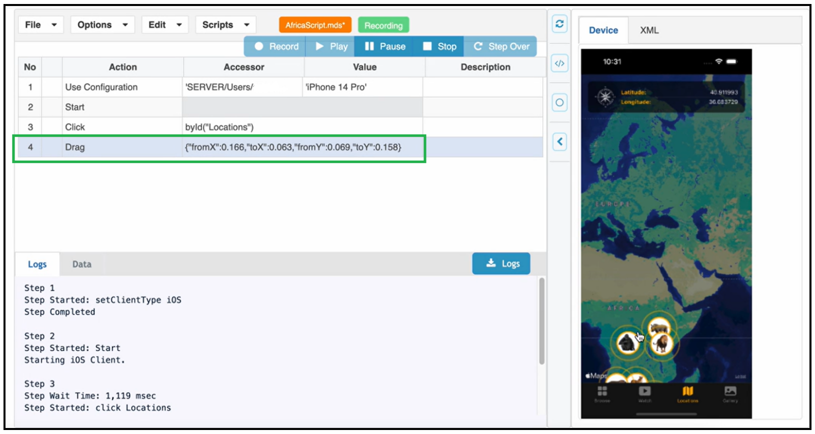
3. Draw the drag arrow and click Save.



No		Action	Accessor	Value	Description
1		Use Configuration	'SERVER/Users/	'iPhone 14 Pro'	
2		Start			
3		Click	byId("Locations")		

```
Step 1
Step Started: setClientType iOS
Step Completed

Step 2
Step Started: Start
Starting iOS Client.

Step 3
Step Wait Time: 1,119 msec
Step Started: click Locations
```

4. The **Drag** action is performed and recorded.

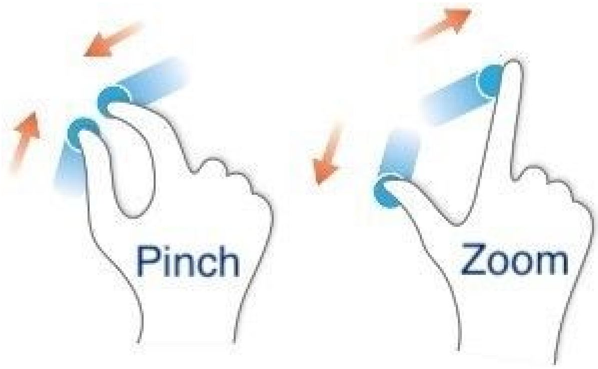


No		Action	Accessor	Value	Description
1		Use Configuration	'SERVER/Users/	'iPhone 14 Pro'	
2		Start			
3		Click	byId("Locations")		
4		Drag	{"fromX":0.166,"toX":0.063,"fromY":0.069,"toY":0.158}		

```
Step 1
Step Started: setClientType iOS
Step Completed

Step 2
Step Started: Start
Starting iOS Client.

Step 3
Step Wait Time: 1,119 msec
Step Started: click Locations
```

Zoom In and Zoom Out

You can perform Zoom In and Zoom Out functions. These function mirrors users performing the pinch and zoom gestures. The following example shows this function in Google Maps.

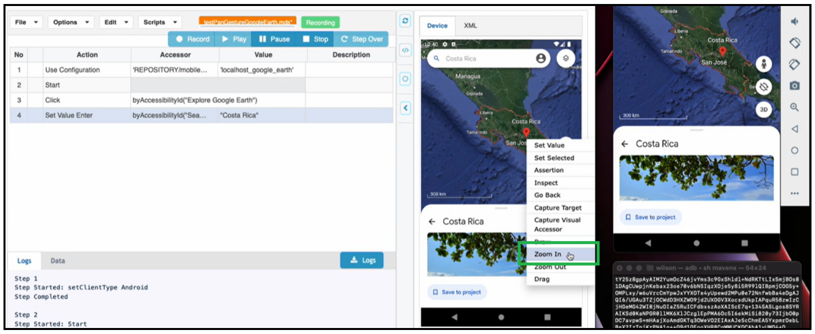


This was added in the AIQ 5.0.0 release.

Zoom In

1. In the **Device** tab, select a region.

2. Right click and select **Zoom In**.



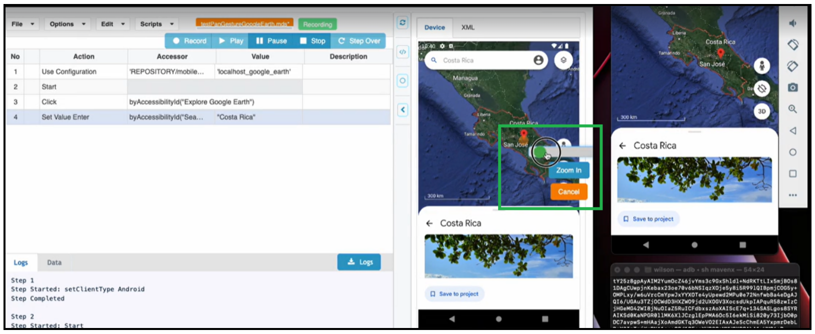
No		Action	Accessor	Value	Description
1		Use Configuration	'REPOSITORY/mobile...	'localhost_google_earth'	
2		Start			
3		Click	byAccessibilityId("Explore Google Earth")		
4		Set Value Enter	byAccessibilityId("Sea...	"Costa Rica"	

Logs | Data | Logs

```
Step 1
Step Started: setClientType Android
Step Completed

Step 2
Step Started: Start
```

```
tY25z8gpAyAIM2YumOcZ46jvYms3c9GxShldl+NdRKTtLIx5mjBOs8
1DAgCUwpjnKebax23oe70v6bNSIqzXOje5y8i5R991QIBpmjCOG5y+
OMPLxy/w6uVrcCmYpwJxYYXOTe4yUpewd2MPu0e72NnfwbBa4eDgAJ
QI6/UGAu3TZjOCWdD3HXZWO9jd2UXOGV3XocsdUkpIAPquR58zwIzC
jHGeMG42WI8jNuOIaZ5RuICFdbxszAoXAIScE7q+134SASLgos8SYR
AIKSd0KaNPGR0llMK6X1JCzglEpPMA6Oc5I6ekMiSi820y73IjbO0p
DC7avpwS+mHAajXoAmdGKTq3OWeVO2EIAxAJeScChmEA5YxpmrDebL
```

3. Adjust the slider as needed and click **Zoom In**.



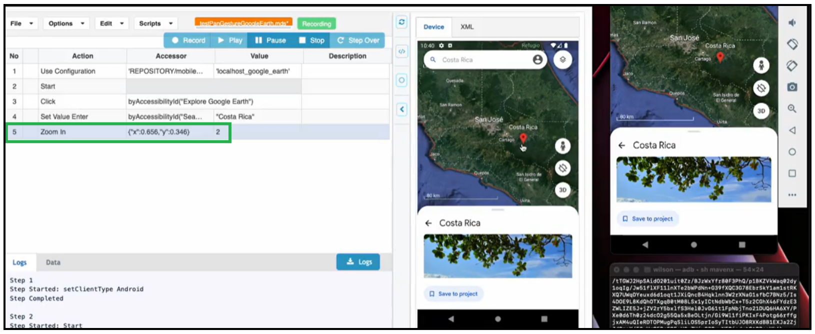
No		Action	Accessor	Value	Description
1		Use Configuration	'REPOSITORY/mobile...	'localhost_google_earth'	
2		Start			
3		Click	byAccessibilityId("Explore Google Earth")		
4		Set Value Enter	byAccessibilityId("Sea...	"Costa Rica"	

Logs | Data | Logs

```
Step 1
Step Started: setClientType Android
Step Completed

Step 2
Step Started: Start
```

```
tY25z8gpAyAIM2YumOcZ46jvYms3c9GxShldl+NdRKTtLIx5mjBOs8
1DAgCUwpjnKebax23oe70v6bNSIqzXOje5y8i5R991QIBpmjCOG5y+
OMPLxy/w6uVrcCmYpwJxYYXOTe4yUpewd2MPu0e72NnfwbBa4eDgAJ
QI6/UGAu3TZjOCWdD3HXZWO9jd2UXOGV3XocsdUkpIAPquR58zwIzC
jHGeMG42WI8jNuOIaZ5RuICFdbxszAoXAIScE7q+134SASLgos8SYR
AIKSd0KaNPGR0llMK6X1JCzglEpPMA6Oc5I6ekMiSi820y73IjbO0p
DC7avpwS+mHAajXoAmdGKTq3OWeVO2EIAxAJeScChmEA5YxpmrDebL
```

4. The **Zoom In** action is performed and recorded.



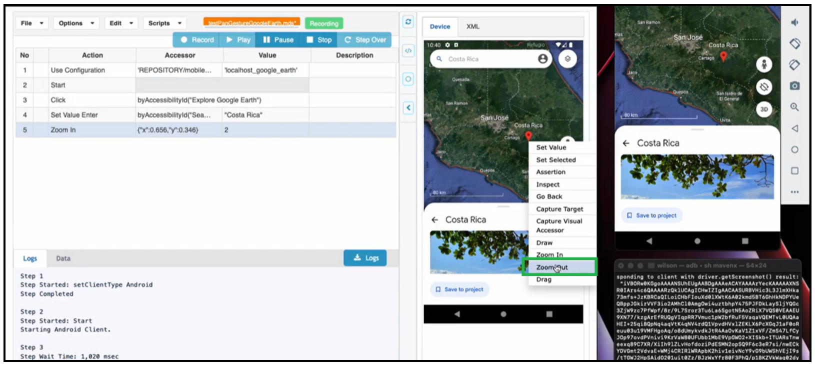
No	Action	Accessor	Value	Description
1	Use Configuration	'REPOSITORY/mobile...	'localhost_google_earth'	
2	Start			
3	Click	byAccessibilityId("Explore Google Earth")		
4	Set Value Enter	byAccessibilityId("Sea...	"Costa Rica"	
5	Zoom In	{"x":0.656,"y":0.346}	2	

```
Step 1
Step Started: setClientType Android
Step Completed

Step 2
Step Started: Start
```

Zoom Out

1. In the **Device** tab, select a region.

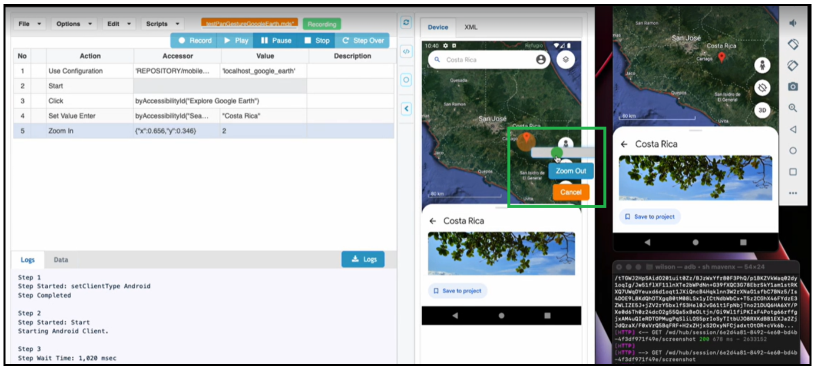
2. Right click and select **Zoom Out**.



No	Action	Accessor	Value	Description
1	Use Configuration	'REPOSITORY/mobile...	'localhost_google_earth'	
2	Start			
3	Click	byAccessibilityId("Explore Google Earth")		
4	Set Value Enter	byAccessibilityId("Sea...	"Costa Rica"	
5	Zoom In	{"x":0.656,"y":0.346}	2	

```
Step 1
Step Started: setClientType Android
Step Completed

Step 2
Step Started: Start
Starting Android Client.

Step 3
Step Wait Time: 1,020 msec
```

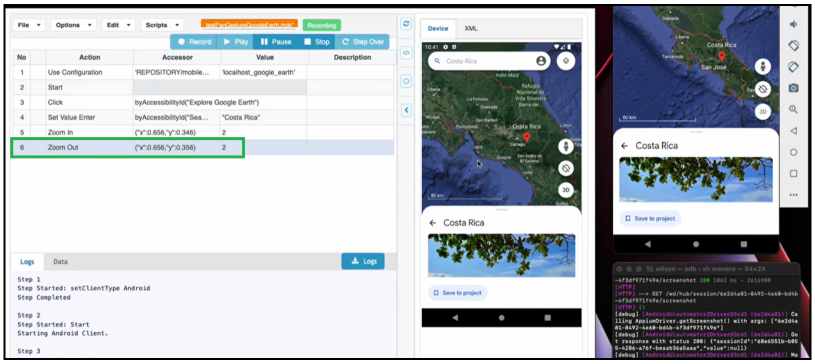
3. Adjust the slider as needed and click **Zoom Out**.



No	Action	Accessor	Value	Description
1	Use Configuration	'REPOSITORY/mobile...	'localhost_google_earth'	
2	Start			
3	Click	byAccessibilityId("Explore Google Earth")		
4	Set Value Enter	byAccessibilityId("Sea...	"Costa Rica"	
5	Zoom In	{"x":0.656,"y":0.346}	2	

```
Step 1
Step Started: setClientType Android
Step Completed

Step 2
Step Started: Start
Starting Android Client.

Step 3
Step Wait Time: 1,020 msec
```

4. The **Zoom Out** action is performed and recorded.



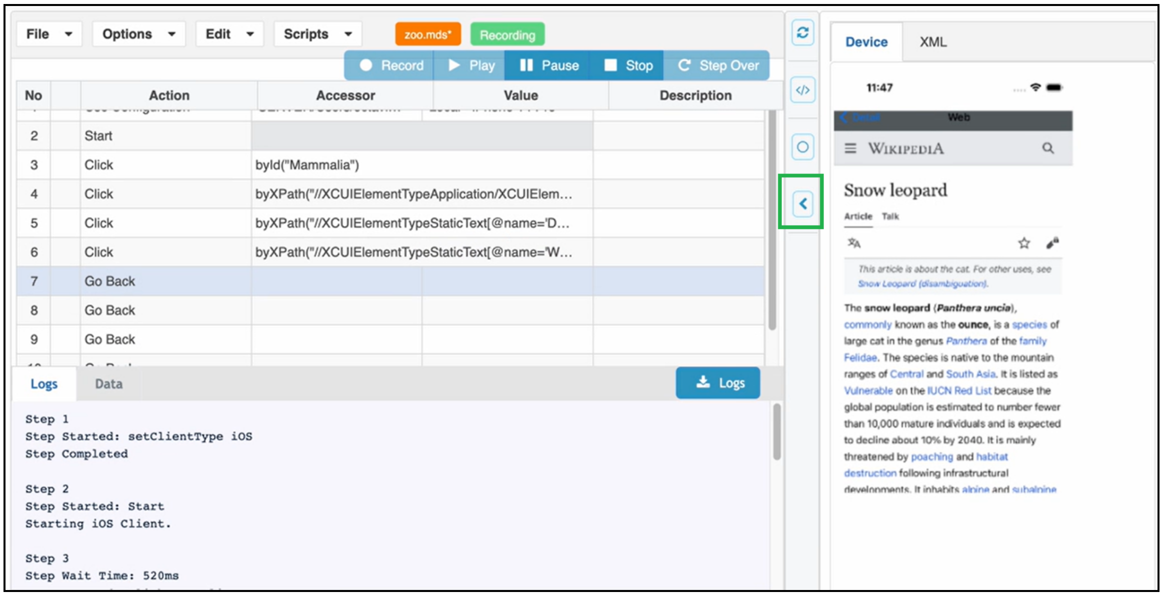
No		Action	Accessor	Value	Description
1		Use Configuration	'REPOSITORY/mobile…	'localhost_google_earth'	
2		Start			
3		Click	byAccessibilityId("Explore Google Earth")		
4		Set Value Enter	byAccessibilityId("Sea…	"Costa Rica"	
5		Zoom In	{"x":0.656,"y":0.346}	2	
6		Zoom Out	{"x":0.656,"y":0.356}	2	

```
Step 1
Step Started: setClientType Android
Step Completed

Step 2
Step Started: Start
Starting Android Client.

Step 3
```

```
-4f3df971f49e/screenshot 200 1062 ms - 2616908
[HTTP]
[HTTP] --> GET /wd/hub/session/6e2d4a81-8492-4e60-bd4b
-4f3df971f49e/screenshot
[HTTP] {}
[debug] [AndroidUiautomator2Driver@3cd1 (6e2d4a81)] Ca
lling AppiumDriver.getScreenshot() with args: ["6e2d4a
81-8492-4e60-bd4b-4f3df971f49e"]
[debug] [AndroidUiautomator2Driver@3cd1 (6e2d4a81)] Go
t response with status 200: {"sessionId":"68e6551b-b05
5-4286-a76f-beaab36a5aaa","value":null}
[debug] [AndroidUiautomator2Driver@3cd1 (6e2d4a81)] Ma
```

“Go Back” Button

Added a back button to the preview pane that and will perform a goBack action. This is available for both iOS and Android.



No	Action	Accessor	Value	Description
2	Start			
3	Click	byId("Mammalia")		
4	Click	byXPath("//XCUIElementTypeApplication/XCUIElem…		
5	Click	byXPath("//XCUIElementTypeStaticText[@name='D…		
6	Click	byXPath("//XCUIElementTypeStaticText[@name='W…		
7	Go Back			
8	Go Back			
9	Go Back			

```
Step 1
Step Started: setClientType iOS
Step Completed

Step 2
Step Started: Start
Starting iOS Client.

Step 3
Step Wait Time: 520ms
```

Snow leopard

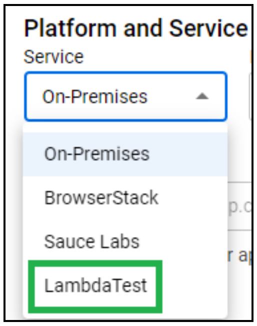
Article Talk

This article is about the cat. For other uses, see Snow Leopard (disambiguation).

The **snow leopard** (***Panthera uncia***), commonly known as the **ounce**, is a species of large cat in the genus *Panthera* of the family Felidae. The species is native to the mountain ranges of Central and South Asia. It is listed as Vulnerable on the IUCN Red List because the global population is estimated to number fewer than 10,000 mature individuals and is expected to decline about 10% by 2040. It is mainly threatened by poaching and habitat destruction following infrastructural developments. It inhabits alpine and subalpine

LambdaTest in Mobile Configuration Options

LambdaTest has been added to the available services for a Mobile Configuration.



Platform and Service

Service

On-Premises

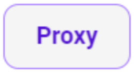
See Mobile Configuration for more information on mobile configurations.

See the LambdaTest website for more information on the LambdaTest platform.

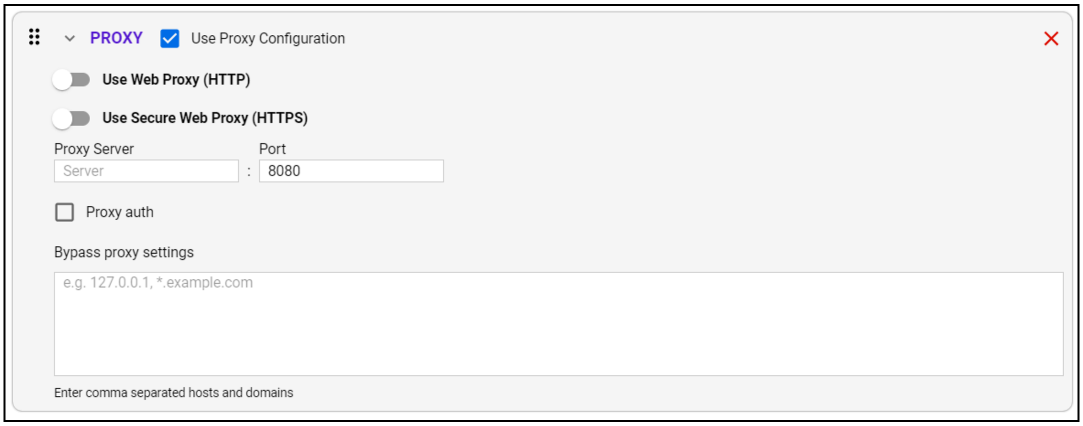
Add a Proxy in the API Test Designer

You can now configure proxy information in the API Test Designer.

1. Select and drag the **Proxy** building block into the workspace.



2. Check the **Use Proxy Configuration** box and click the down arrow to expand the configuration options.



PROXY ☑ Use Proxy Configuration ✕

Use Web Proxy (HTTP)

Use Secure Web Proxy (HTTPS)

Proxy Server: Server

Port: 8080

□ Proxy auth

Bypass proxy settings

e.g. 127.0.0.1, *.example.com

Enter comma separated hosts and domains

3. Configure the proxy options as needed.

   - Use Web Proxy (HTTP)
   - Use Secure Web Proxy (HTTPS)
   - Proxy Server
   - Port

- Proxy Auth
- Bypass Proxy Settings

> For more information on API testing, see API Testing with AIQ.

Azure Monitor Integration

Appvance supports an Application Performance Monitoring (APM) integration with Microsoft's Azure Monitor.

This integration was added in the AIQ 5.0.0 release.

For more information see Microsoft's overview of Azure Monitor.

Configure AIQ Integration with Azure Monitor

1. Navigate to **Global Options** > **Preferences** > **APM Integration**.

2. Select **Azure Monitor**.

3. Enter the URL of your Azure Monitor portal and a name for the integration.

4. Select a Time Range to set the time to analyze along the execution. For example, if you select 30 minutes, it will analyze 5 minutes before and 5 minutes after a point selected in the execution.

5. Click **Save**.

6. Confirm the integration is saved in the dropdown at the top left corner.

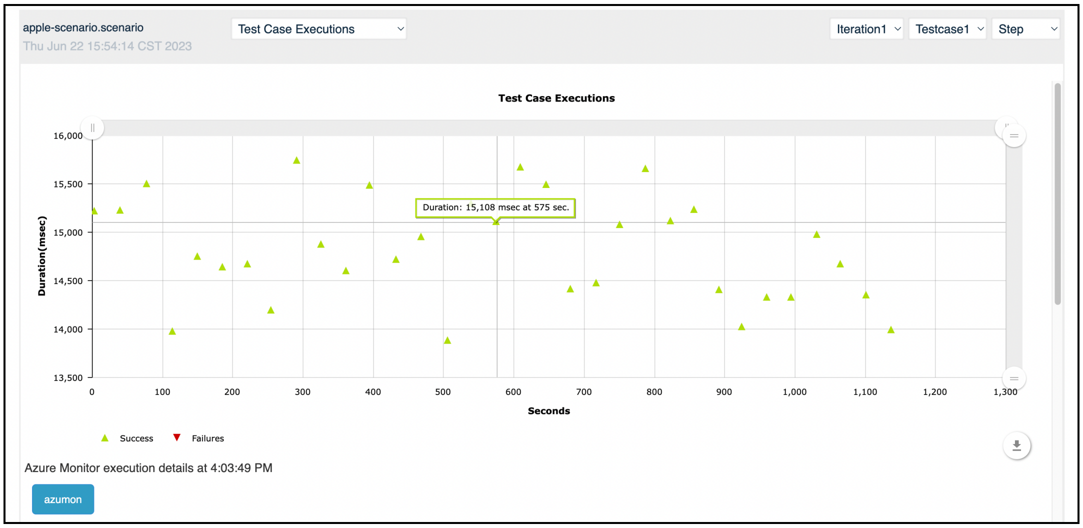
Seeing the Data in Azure Monitor

1. In AIQ run the scenario.

2. When the scenario is complete, navigate to **Reports** > **Test Case Executions**.

3. Click the link to see the report.

4. Click any of the green triangles. Make note of the time index. Example: Azure Monitor execution details at 4:03:49 PM.



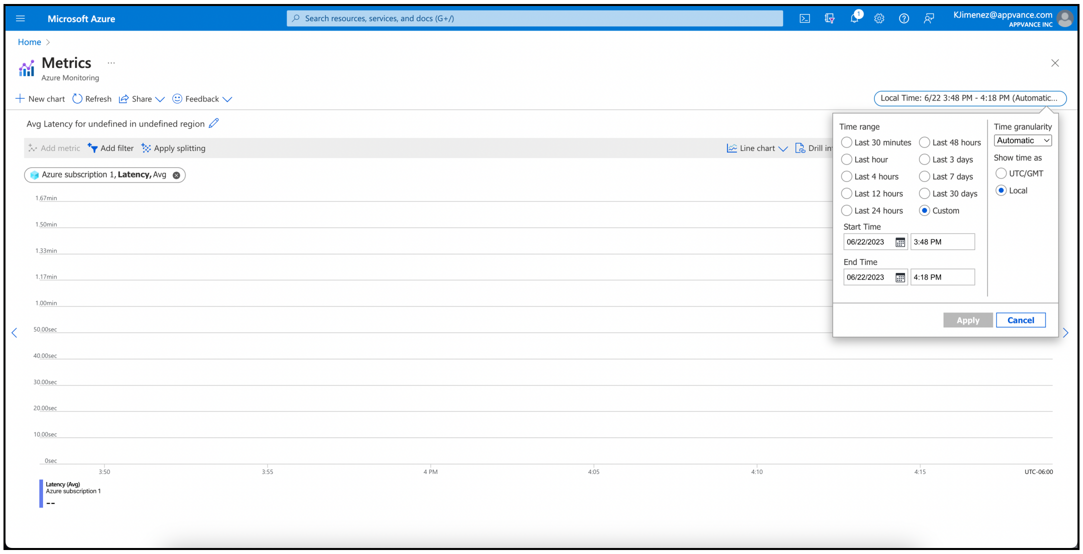
apple-scenario.scenario
Thu Jun 22 15:54:14 CST 2023

Test Case Executions | Iteration1 | Testcase1 | Step

Azure Monitor execution details at 4:03:49 PM

5. Click the blue button to open a new window with the Azure Monitor report between the time specified in the APM integration settings. Example: 30 minutes should show the report between 3:48 pm and 4:18

pm.



Local Time: 6/22 3:48 PM - 4:18 PM (Automatic...

Time range

○ Last 30 minutes ○ Last 48 hours
○ Last hour ○ Last 3 days
○ Last 4 hours ○ Last 7 days
○ Last 12 hours ○ Last 30 days
○ Last 24 hours ● Custom

Start Time: 06/22/2023 3:48 PM

End Time: 06/22/2023 4:18 PM

Time granularity: Automatic

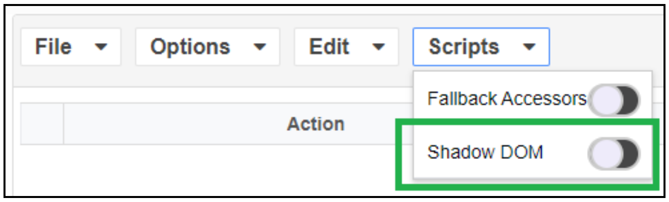
Show time as

○ UTC/GMT
● Local

Shadow DOM Setting

This applies to the Web Designer (Classic) interface. This option is not present in the new Web Designer interface.

A Shadow DOM setting (forcedShadowDom) has been added to the global.json file. The default value is true. The Shadow DOM setting in the script options in the Test Designer IDE will be ignored if the forcedShadowDom is set to true. Even if that option is set to false in the Test Designer IDE, scripts will be run with the option set to true.



This change was made in the AIQ 4.10.6 release and is propagated to AIQ 5.0.0.

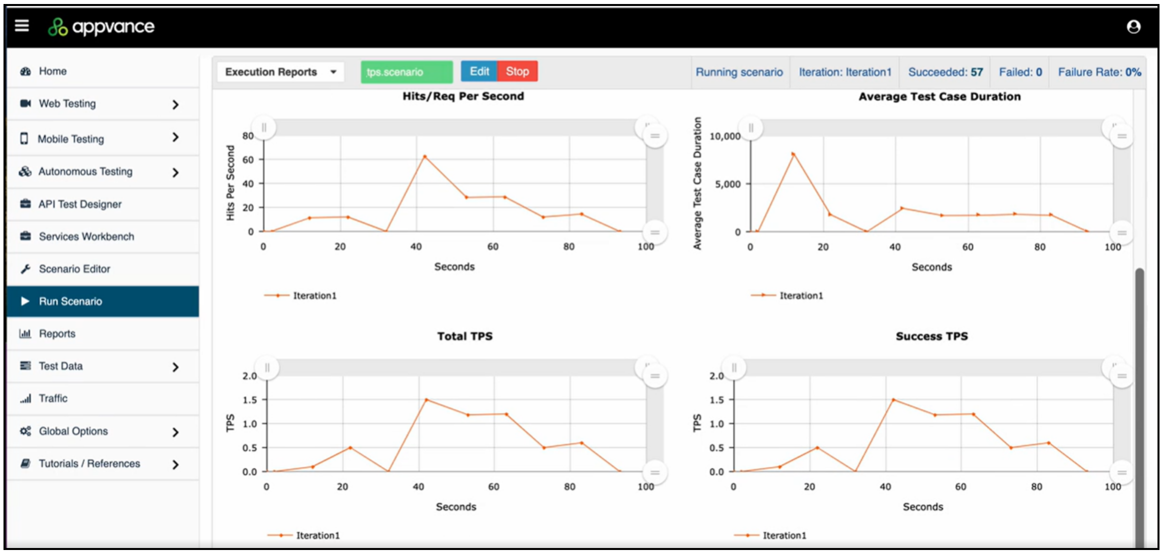
Generally speaking, there is no longer any reason for having the Shadow DOM option turned off. Setting it off will not help or prevent any problems or failures. Having it set to off can actually cause failures in instances where a script tries to perform a click action on a element that is under Shadow DOM.

In previous releases, there may have been concerns of some potential performance issues when the setting was on versus the setting being off. Those issues have been addressed.

See About Shadow DOM Settings for more information.

Requests per Second and Transactions Per Second

When running load or performance tests, you see the requests per second (RPS), transactions per second (TPS), or a successful number of transactions per second (Success TPS) as part of the execution reports.



The graphic above shows the RPS, TPS, and Success TPS details for a scenario execution. While the graphs for RPS and TPS are similar, they are not identical. Requests per second (RPS) is a different measurement than transactions per second (TPS). This is because a single transaction may consist of several requests. For example, a single login operation may contain five HTTP requests that can be grouped together as a single transaction.

Functional Changes

These are the various small changes to functionality that were made in the 5.0.0 release of AIQ. The changes are grouped by functional area.

General

Warning if Test Node and Controller Versions are Different

When one or more test nodes Using location have a different version than the controller, the user will be notified by the warning message: “The Test Node you are trying to use has a different version than the controller you are currently on. This may cause performance and reliability issues.“ in the Play Controller.

Updated Error Messages

Updated various error messages to be more specific in describing the error.

Added Encryption to Cloud Config Files

Added an encryption flag (`-Encryption flag (Yes/No)`) in the Cloud Config file. When the Encryption flag is Yes, the AIQ Test Node will be created with encryption using the default AWS KMS Key for that account.

Administration

Additional Session Information in User Management

You can now see when a users session was create and when the last accessed the AIQ server. To view this information navigate to **Global Options** > **Admin Options** > **User Management** tab. The **Session Created** and **Last Access** columns display the respective information.

Test Designer / Web Designer

Auto Save Option Added to IDE

Added an Auto Save option to the Options menu. Files will be auto saved every 30 seconds.

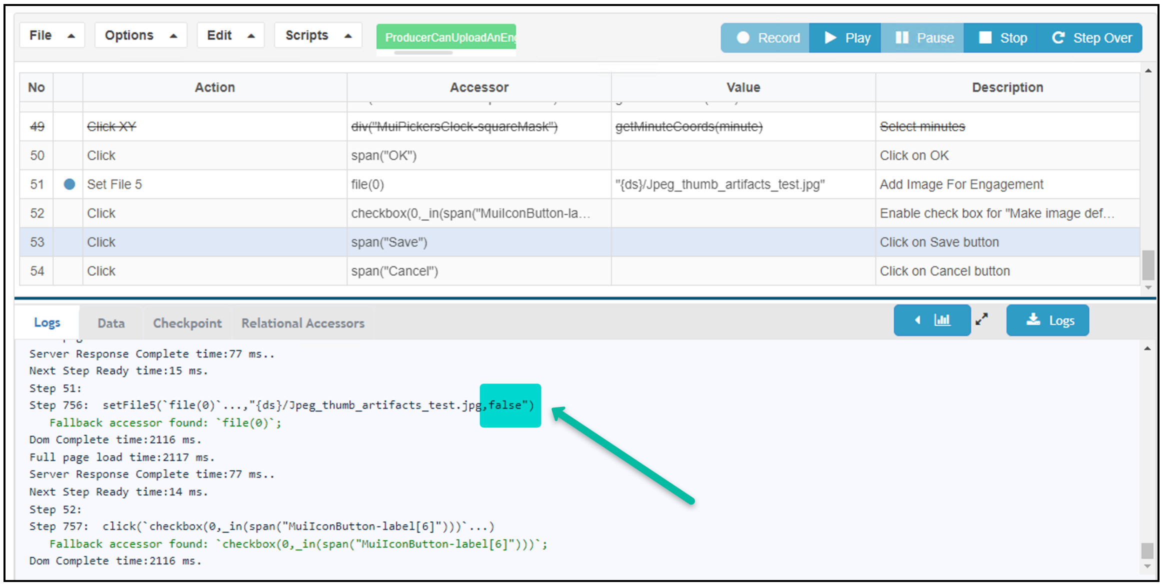
This option is only available in the Web Designer (Classic) interface. The new Web Designer interface auto saves your files by default.

Drag and Drop Data

You can drag and drop data from the Data page directly into a test script.

Removed “False” Indicator from Set File

“False” was an internal value and confusing to users. It has been removed.

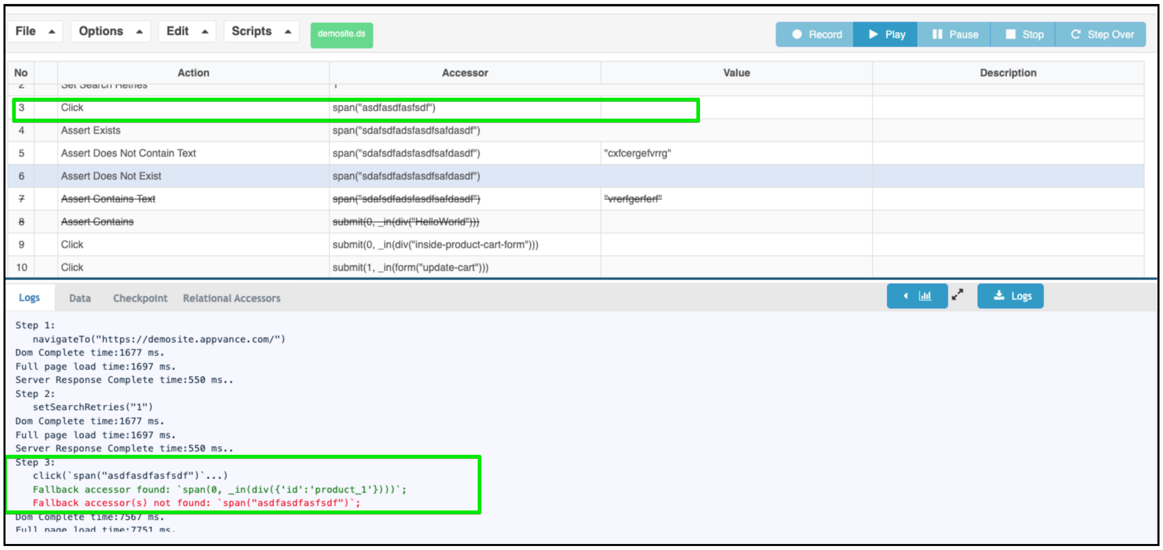


No		Action	Accessor	Value	Description
~~49~~		~~Click XY~~	~~div("MuiPickersClock-squareMask")~~	~~getMinuteCoords(minute)~~	~~Select minutes~~
50		Click	span("OK")		Click on OK
51	●	Set File 5	file(0)	"{ds}/Jpeg_thumb_artifacts_test.jpg"	Add Image For Engagement
52		Click	checkbox(0,_in(span("MuiIconButton-la…		Enable check box for "Make image def…
53		Click	span("Save")		Click on Save button
54		Click	span("Cancel")		Click on Cancel button

```
Server Response Complete time:77 ms..
Next Step Ready time:15 ms.
Step 51:
Step 756:  setFile5(`file(0)`...,"{ds}/Jpeg_thumb_artifacts_test.jpg,false")
   Fallback accessor found: `file(0)`;
Dom Complete time:2116 ms.
Full page load time:2117 ms.
Server Response Complete time:77 ms..
Next Step Ready time:14 ms.
Step 52:
Step 757:  click(`checkbox(0,_in(span("MuiIconButton-label[6]")))`...)
   Fallback accessor found: `checkbox(0,_in(span("MuiIconButton-label[6]")))`;
Dom Complete time:2116 ms.
```

Logging of Fallback Accessors

Improved the logging of fallback accessors to report which accessors were evaluated and clearly showing which accessor was found.



No	Action	Accessor	Value	Description
2	Set Search Retries	1		
3	Click	span("asdfasdfasfsdf")		
4	Assert Exists	span("sdafsdfadsfasdfsafdasdf")		
5	Assert Does Not Contain Text	span("sdafsdfadsfasdfsafdasdf")	"cxfcergefvrrg"	
6	Assert Does Not Exist	span("sdafsdfadsfasdfsafdasdf")		
~~7~~	~~Assert Contains Text~~	~~span("sdafsdfadsfasdfsafdasdf")~~	~~"vrerfgerferf"~~	
~~8~~	~~Assert Contains~~	~~submit(0, _in(div("HelloWorld")))~~		
9	Click	submit(0, _in(div("inside-product-cart-form")))		
10	Click	submit(1, _in(form("update-cart")))		

```
Step 1:
   navigateTo("https://demosite.appvance.com/")
Dom Complete time:1677 ms.
Full page load time:1697 ms.
Server Response Complete time:550 ms..
Step 2:
   setSearchRetries("1")
Dom Complete time:1677 ms.
Full page load time:1697 ms.
Server Response Complete time:550 ms..
Step 3:
   click(`span("asdfasdfasfsdf")`...)
   Fallback accessor found: `span(0, _in(div({'id':'product_1'})))`;
   Fallback accessor(s) not found: `span("asdfasdfasfsdf")`;
Dom Complete time:7567 ms.
Full page load time:7751 ms.
```

- Found fallback accessors are noted in green in the log.
- Not found fallback accessors are noted in red in the log.

Mobile Testing

Added clickPoint and Matching Cropped Area to Step Image

The selected clickpoint and the and the matching cropped area are now displayed in the step image.

API Testing

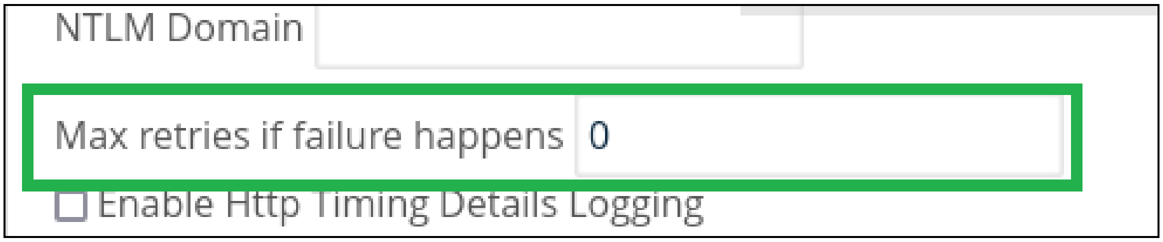
API Credentials and Validations

Credentials validation was added at the API level.

Scenario Editor

Retry Logic in Scenario Execution

Retry logic has been added as a global option for scenario executions in the Scenario Editor. This option allows you to set the number of times a scenario will be rerun if it fails during the initial run.



NTLM Domain

Max retries if failure happens 0

□ Enable Http Timing Details Logging

For example, if you specify “3” as the maximum number of retries, the scenario will be rerun a maximum of three times if the first execution fails.

See Maximum Number of Retries Setting for Scenarios for more information.

Services Workbench

New Function to Create Subfolders in Inbox

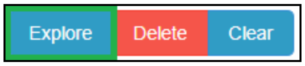
Added the ability to set the work folder to a sub folder under the inbox. You can access subfolders and execute all the current functions like read, delete, reply, etc.

The command is : `graphService.setMailFolder("parent_folder", "child_folder");`

Administration

Explore Function in Repository Configuration

The **Explore** button on the Repository configuration page (**Global Options** > **Preferences** > **Repository**) now allows you to explore any of the connection types. Previously, this option was only available for GIT repositories.



For more information on repositories, see Repositories.

AIQ 4.10.x Enhancements

Depending on what AIQ release you are upgrading from to AIQ 5.0.0, there may be additional functional enhancements included in this release. For your convenience, information about the enhancements in targeted 4.10.x releases has been added to this document.

> Refer to the release notes for all the AIQ releases since your current release for more details.

- "Release 4.10.1" below
- "Release 4.10.2" on page 190
- "Release 4.10.3" on page 191
- "Release 4.10.4" on page 195
- "Release 4.10.6" on page 197

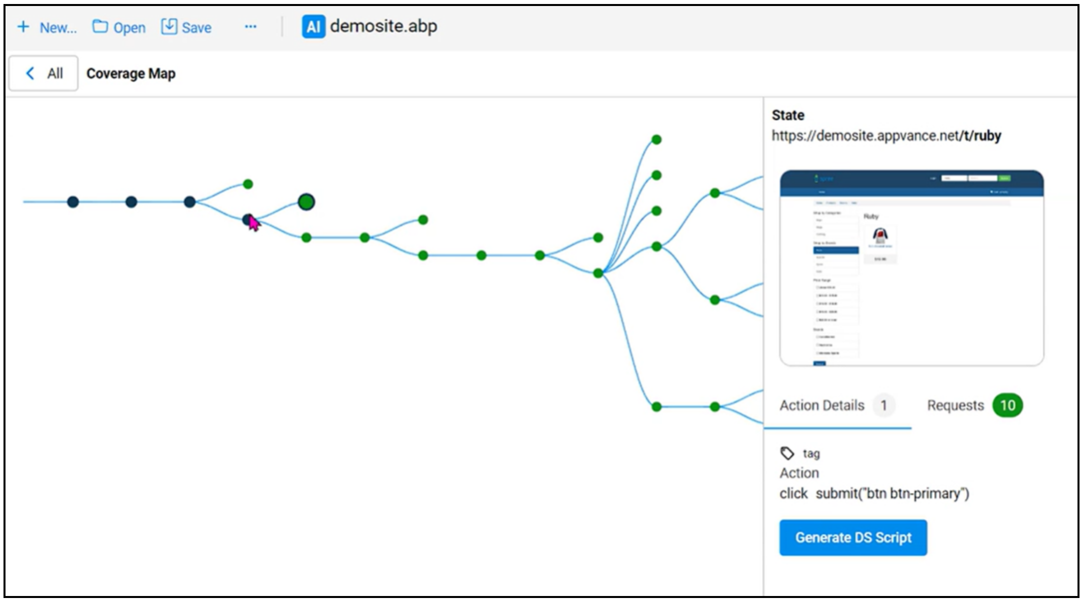
Release 4.10.1

AIQ 4.10.1 contained the following functional enhancements.

Creating a Script from a Coverage Map

You can now click on a path in the Coverage Map and generate the test script for that path directly from the Coverage Map. This makes it easy for you to locate and run the test script for a particular flow rather than having to locate the script for a particular path from the list of all of your downloaded scripts.

1. From the **Coverage Map**, click on a node.



State
https://demosite.appvance.net/t/ruby

tag
Action
click submit("btn btn-primary")

2. Click the **Generate DS Script** button.

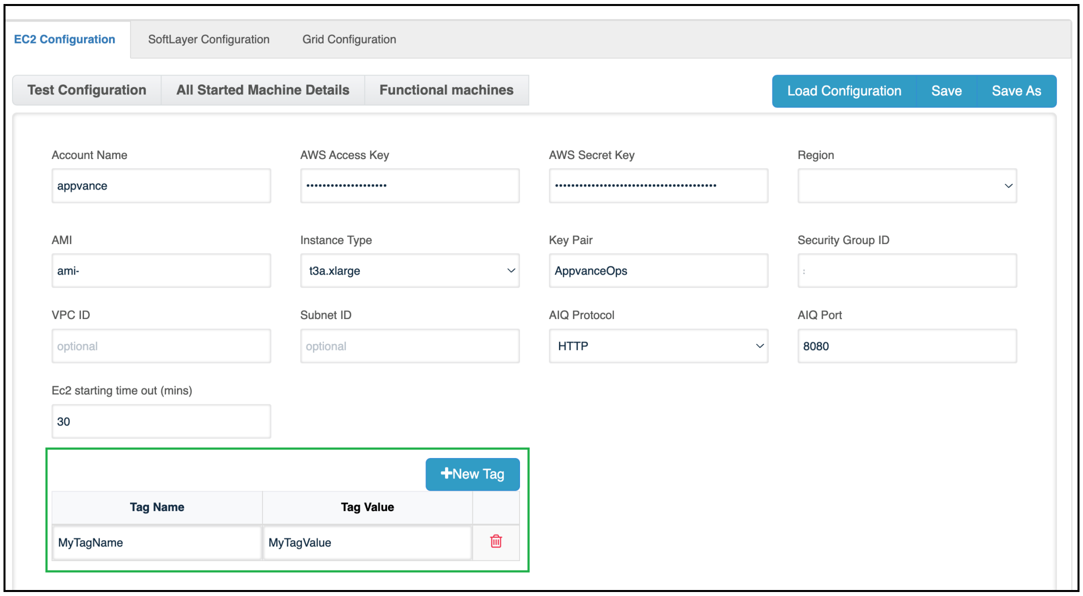
3. The script and blueprint files (.stags; .netx, .valid, DPL) are downloaded as a zip file from Coverage Map.

4. You can upload the .zip file to your repository and open the script in Test Designer.

Tag Instances in the Cloud Config Editor

You can create a custom tag in the Cloud Config Editor. For example, you can assign a tag to all instances using a particular configuration as use that tag for tracking resources, ownership, etc.

1. In AIQ, navigate to **Global Options** > **Cloud Config** > **EC2 Configuration**.



Account Name appvance	AWS Access Key ••••••••••••••••••••	AWS Secret Key ••	Region
AMI ami-	Instance Type t3a.xlarge	Key Pair AppvanceOps	Security Group ID :
VPC ID optional	Subnet ID optional	AIQ Protocol HTTP	AIQ Port 8080
Ec2 starting time out (mins) 30			

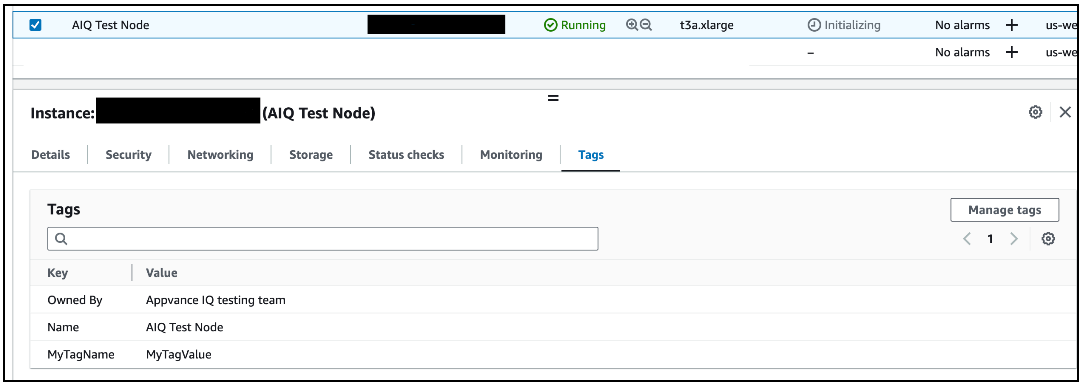
Tag Name	Tag Value
MyTagName	MyTagValue

2. Click **New Tag** to add a custom tag to the configuration.

3. Start the scenario.

4. Open the AIQ Console, select your EC2 instance.

5. On the **Tags** tab and check that the tag you created in the cloud config is there.



☑ AIQ Test Node [redacted] Running t3a.xlarge Initializing No alarms + us-we
– No alarms + us-we

Instance: [redacted] **(AIQ Test Node)**

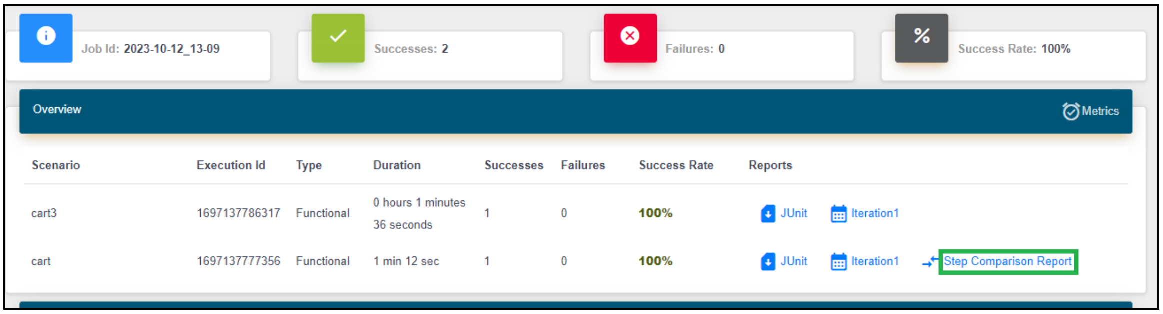
Tags

Key	Value
Owned By	Appvance IQ testing team
Name	AIQ Test Node
MyTagName	MyTagValue

Step Times Comparison Report

Step times comparison reporting has been added for CICD test executions. These reports show you a comparison of step times between a recent execution against a previous execution for a given test case.

Click the **Step Comparison Report** to see the reports.



Job Id: 2023-10-12_13-09

Successes: 2

Failures: 0

Success Rate: 100%

Scenario	Execution Id	Type	Duration	Successes	Failures	Success Rate	Reports
cart3	1697137786317	Functional	0 hours 1 minutes 36 seconds	1	0	100%	JUnit Iteration1
cart	1697137777356	Functional	1 min 12 sec	1	0	100%	JUnit Iteration1 Step Comparison Report

Options

- You can specify a base line execution to compare against previous base line executions.

- You can select which steps should be graphed and then generate the graph with only those selected steps.

Considerations

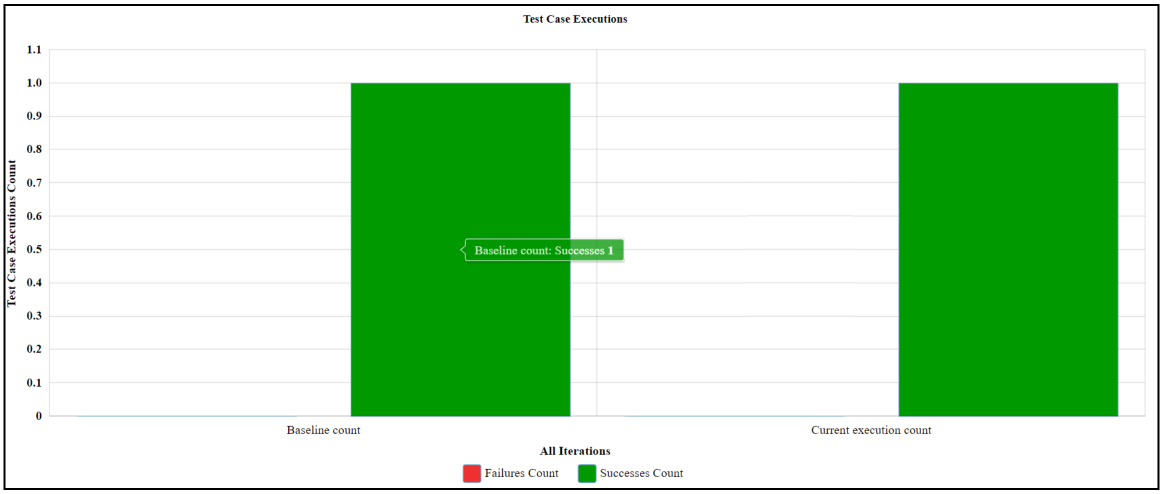
- Test executions must have the same test case name.

- If the test case names do not match, the comparison report will only compare the headers and will be unable to compare any step detail.

- Only step names that have the same step name or description can be compared.

- Mismatched step names will be ignored. However, a list of the step names that were ignored will be created.
- Add descriptions to all steps that you want to compare.
- Step description ensure that even when the step accessors change, the step name will be consistent.
- Scenarios with the **TestCase Only** option selected in the Scenario Editor will not have summary data, which will result in an error for those executions on the Step Times Comparison report.
- Scenarios with zero successes will not have duration summary data for steps, and won't display on the Step Times Comparison report. However, a warning will be displayed that durations were ignored for failed test cases and steps.

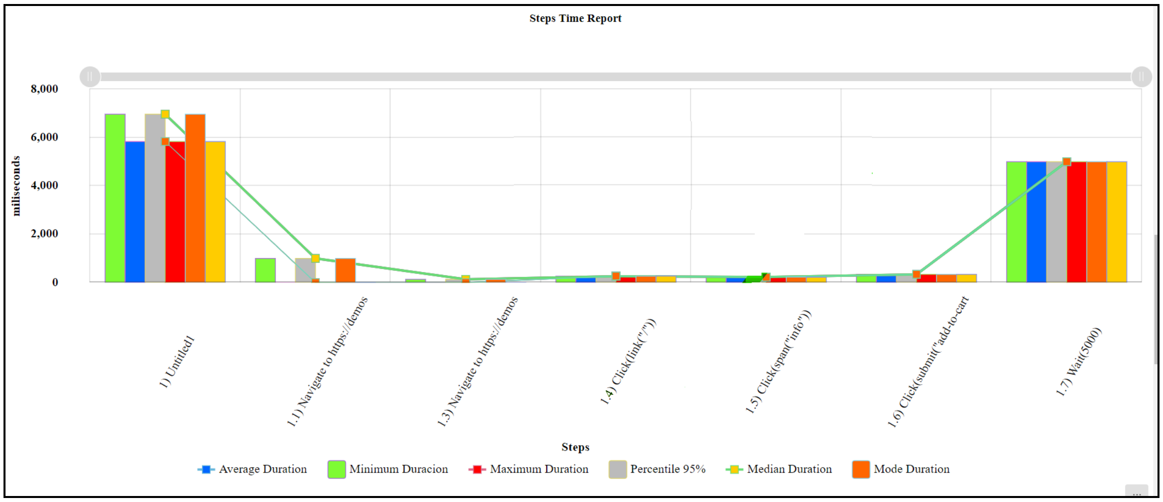
Test Case Execution - Sample Report



The default list of metrics on the Test Execution Report include:

- Failures Count
- Successes Count

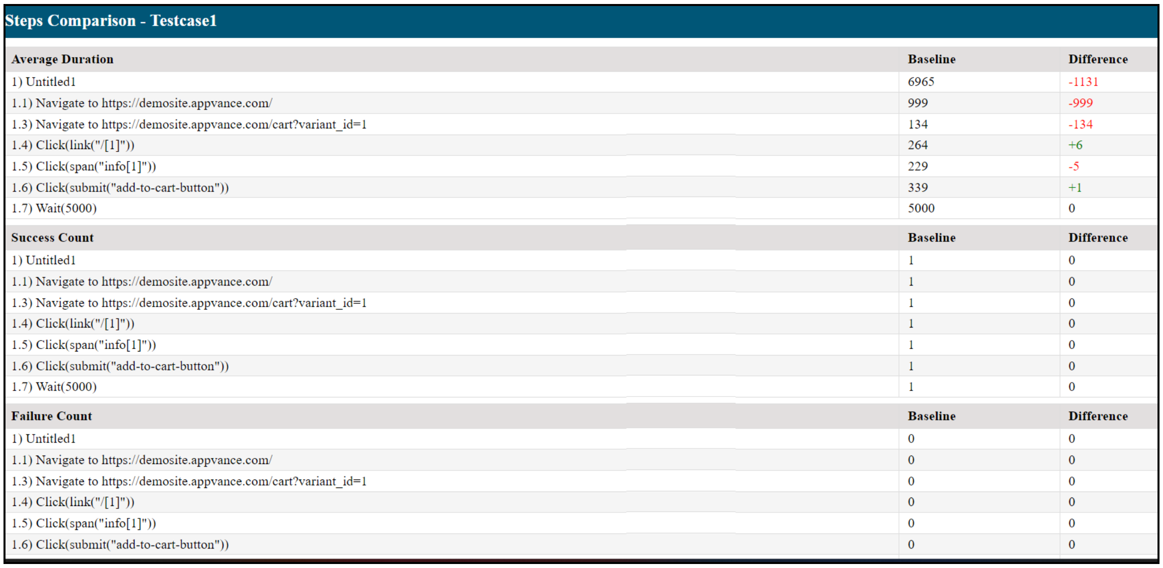
Step Times Report - Sample Report



The default list of metrics on the Step Times Report include:

- Average Duration
- Minimum Duration
- Maximum Duration
- 95% Percentile Duration
- Median Duration
- Mode Duration

Step Comparison Report - Sample Report



Steps Comparison - Testcase1

Average Duration	**Baseline**	**Difference**
1) Untitled1	6965	-1131
1.1) Navigate to https://demosite.appvance.com/	999	-999
1.3) Navigate to https://demosite.appvance.com/cart?variant_id=1	134	-134
1.4) Click(link("/[1]"))	264	+6
1.5) Click(span("info[1]"))	229	-5
1.6) Click(submit("add-to-cart-button"))	339	+1
1.7) Wait(5000)	5000	0
Success Count	**Baseline**	**Difference**
1) Untitled1	1	0
1.1) Navigate to https://demosite.appvance.com/	1	0
1.3) Navigate to https://demosite.appvance.com/cart?variant_id=1	1	0
1.4) Click(link("/[1]"))	1	0
1.5) Click(span("info[1]"))	1	0
1.6) Click(submit("add-to-cart-button"))	1	0
1.7) Wait(5000)	1	0
Failure Count	**Baseline**	**Difference**
1) Untitled1	0	0
1.1) Navigate to https://demosite.appvance.com/	0	0
1.3) Navigate to https://demosite.appvance.com/cart?variant_id=1	0	0
1.4) Click(link("/[1]"))	0	0
1.5) Click(span("info[1]"))	0	0
1.6) Click(submit("add-to-cart-button"))	0	0

The default list of metrics on the Steps Comparison Report include:

- Average Duration
- Successes Count
- Failures Count
- Maximum Duration
- Minimum Duration

Images in Mobile SmartTags

The following enhanced functionality is now available in the Mobile SmartTag Workbench.

Add or Remove Images

You can now modify an existing SmartTag to remove or add images.

Support for Relative Paths for Images

Added support for relative paths for image URLs for SmartTags. You can convert an image URL between absolute and relative paths.

- Image URL with an absolute path. Click Relative to convert the image URL to a relative path.

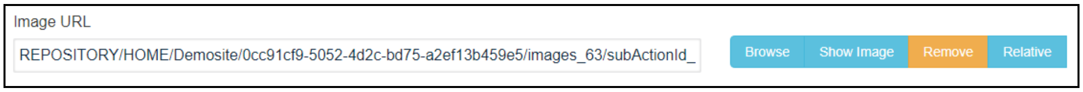


Image URL

REPOSITORY/HOME/Demosite/0cc91cf9-5052-4d2c-bd75-a2ef13b459e5/images_63/subActionId_

- Image URL with a relative path. Click Absolute to convert the image URL to an absolute path.

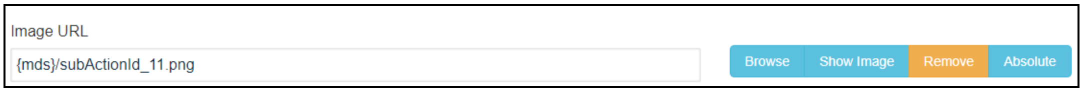
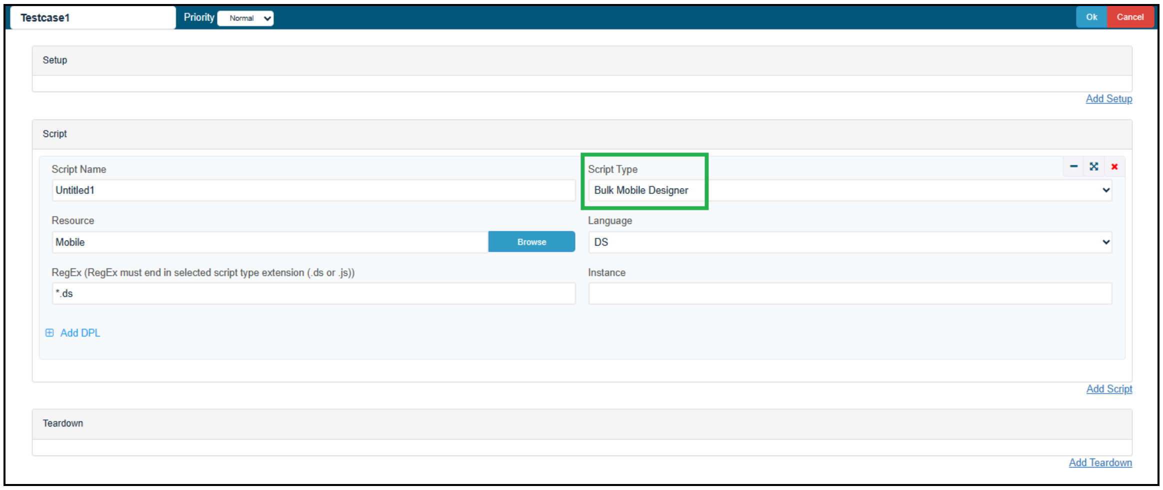


Image URL

{mds}/subActionId_11.png

Bulk Mobile Designer Test Cases

There is a new option in the Scenario Editor to create a bulk scenario for mobile testing.



Testcase1

Priority: Normal

Setup

Script

Script Name: Untitled1

Script Type: Bulk Mobile Designer

Resource: Mobile

Language: DS

RegEx (RegEx must end in selected script type extension (.ds or .js)): *.ds

Instance

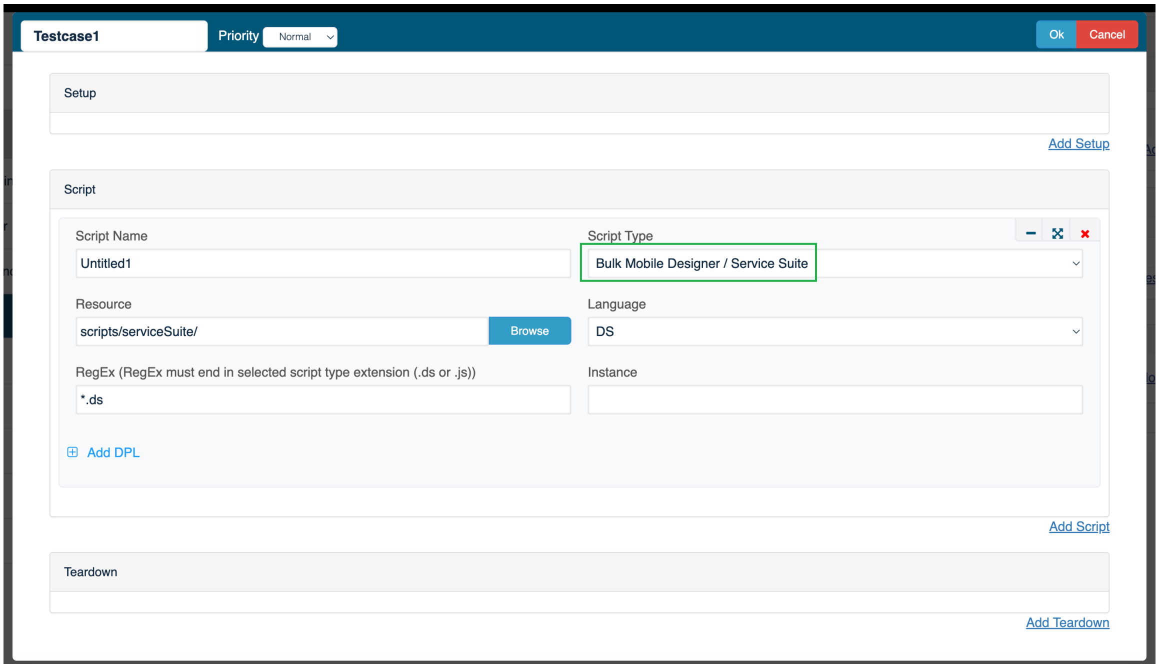
Teardown

Release 4.10.2

AIQ 4.10.2 contained the following functional enhancements.

Bulk Service Suite Test Cases

There is a new option in the Scenario Editor to create a bulk service suite test case.



Testcase1

Priority: Normal

Setup

Script

Script Name: Untitled1

Script Type: Bulk Mobile Designer / Service Suite

Resource: scripts/serviceSuite/

Language: DS

RegEx (RegEx must end in selected script type extension (.ds or .js)): *.ds

Instance:

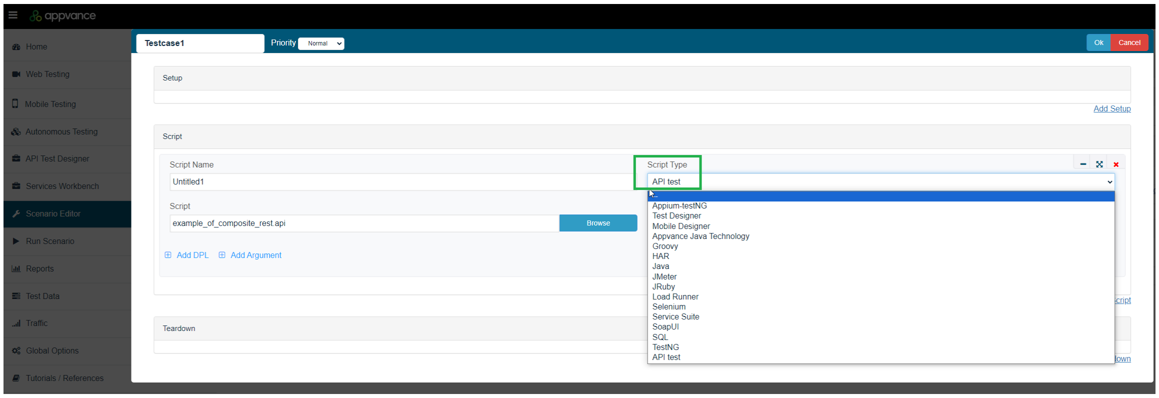
Teardown

Release 4.10.3

AIQ 4.10.3 contained the following functional enhancements.

Run a Performance Scenario for an API Editor Script

Performance Scenarios created using Scenario Editor can now include API Test cases.


Testcase1

Priority: Normal

Script Name: Untitled1

Script Type: API test

Script: example_of_composite_rest.api

Types of Tests for general information on Performance Tests.

Passing Parameters to the JSON Body of a Request from a Previous Response

You can now customize JSON when it is sent in the body of a request, by passing data name from previous response in double braces. The format is: `{{param}}`

This update also impacts URL parameters. All scripts that used {param} must be updated to `{{param}}`.

Capture Browser URL Information when a Script Fails

You can now capture the browser URL from script failures, and that information will display on reports.

To enable this function:

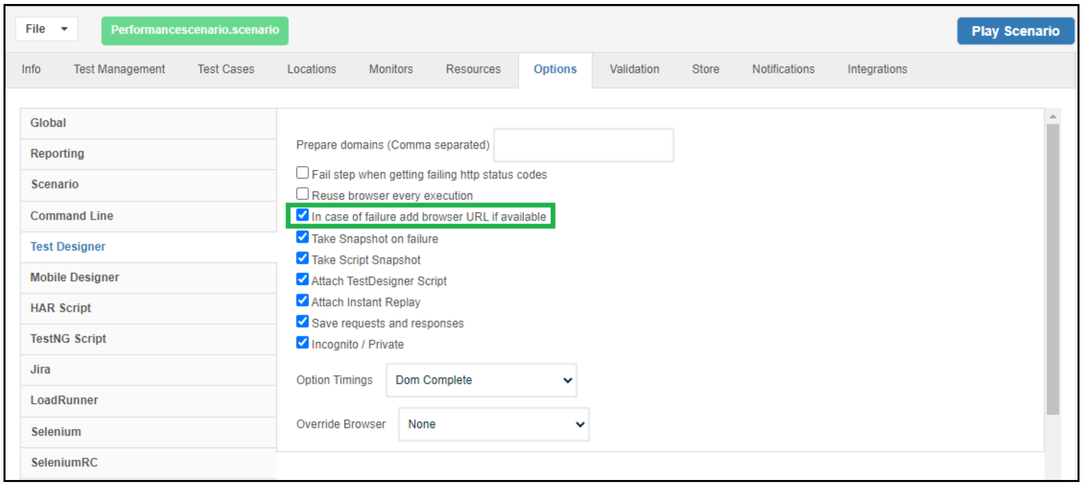
1. Navigate to the Scenario Editor.

2. Open a scenario.

3. Navigate to the **Options** tab.

4. Select **Test Designer**.

5. Select the **In case of failure add browser URL if available** option.



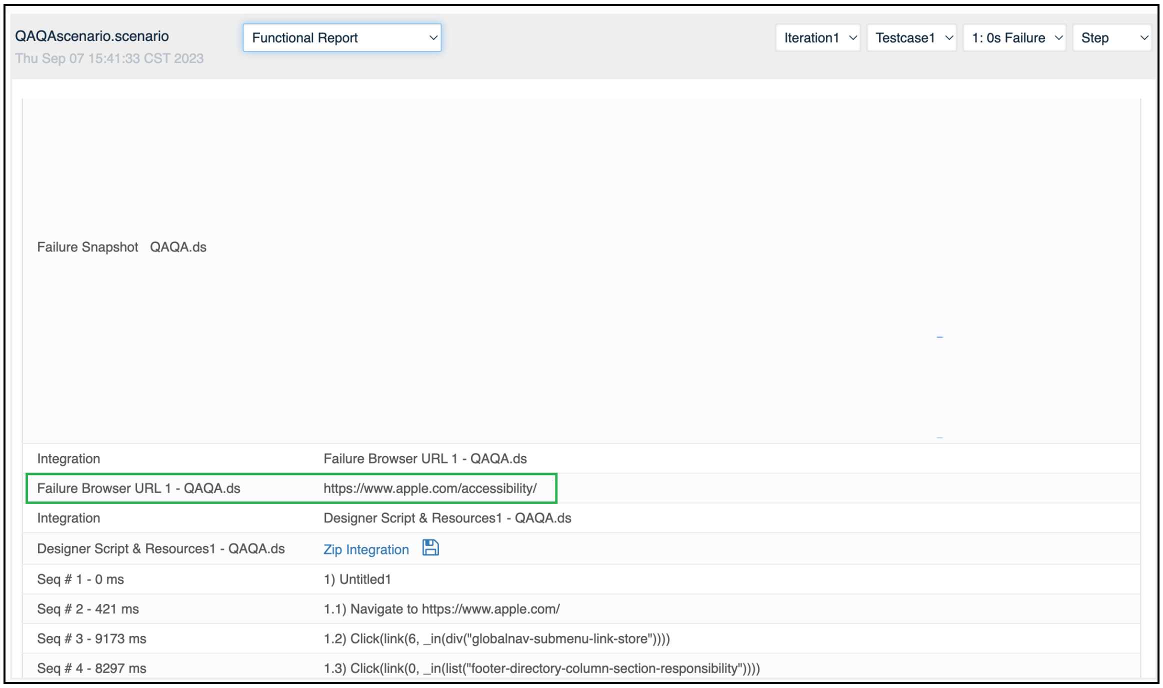
Prepare domains (Comma separated)

☐ Fail step when getting failing http status codes
☐ Reuse browser every execution
☑ In case of failure add browser URL if available
☑ Take Snapshot on failure
☑ Take Script Snapshot
☑ Attach TestDesigner Script
☑ Attach Instant Replay
☑ Save requests and responses
☑ Incognito / Private

Option Timings: Dom Complete

Override Browser: None

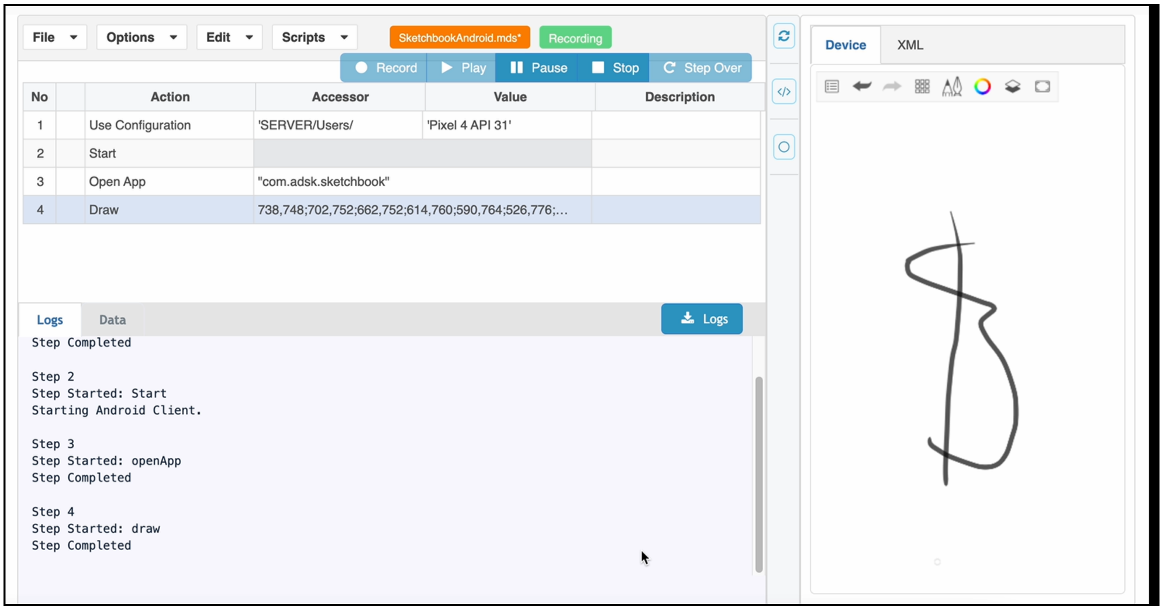
Example report showing browser failure URL



Failure Snapshot QAQA.ds	
Integration	Failure Browser URL 1 - QAQA.ds
Failure Browser URL 1 - QAQA.ds	https://www.apple.com/accessibility/
Integration	Designer Script & Resources1 - QAQA.ds
Designer Script & Resources1 - QAQA.ds	Zip Integration
Seq # 1 - 0 ms	1) Untitled1
Seq # 2 - 421 ms	1.1) Navigate to https://www.apple.com/
Seq # 3 - 9173 ms	1.2) Click(link(6, _in(div("globalnav-submenu-link-store"))))
Seq # 4 - 8297 ms	1.3) Click(link(0, _in(list("footer-directory-column-section-responsibility"))))

Electronic Signatures in Mobile Scripts in JS Edit

You can now capture electronic signatures during Mobile testing. The following screen shot shows a very basic example using the draw function of a sketchbook app.

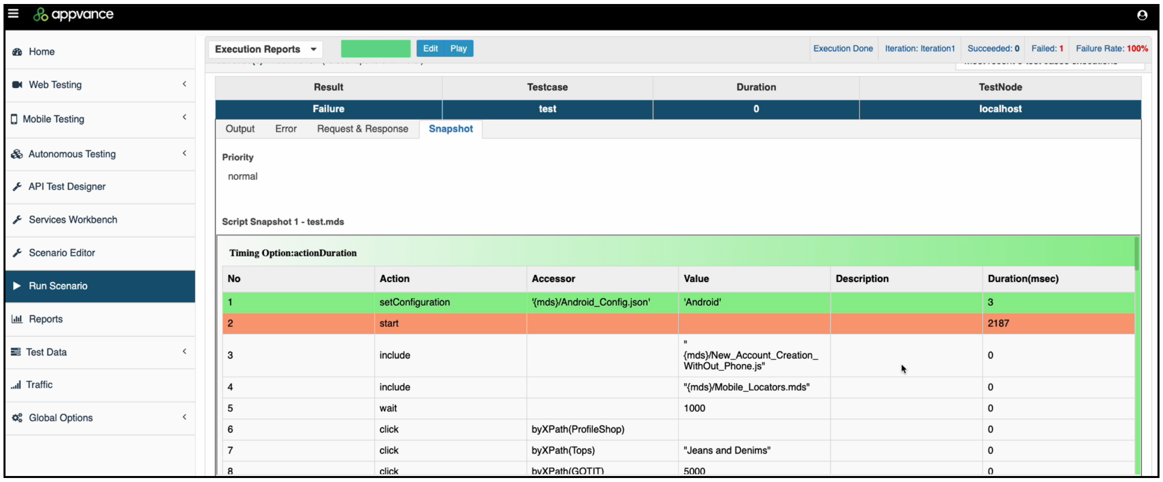


No	Action	Accessor	Value	Description
1	Use Configuration	'SERVER/Users/	'Pixel 4 API 31'	
2	Start			
3	Open App	"com.adsk.sketchbook"		
4	Draw	738,748;702,752;662,752;614,760;590,764;526,776;...		

```
Step Completed

Step 2
Step Started: Start
Starting Android Client.

Step 3
Step Started: openApp
Step Completed

Step 4
Step Started: draw
Step Completed
```

Mobile Designer Dashboard Report Enhancements

Script Snapshots are now available for Mobile Testing. Steps that pass will be displayed in green, while steps that fail will display in red.



Result	Testcase	Duration	TestNode
Failure	test	0	localhost

Priority

normal

Script Snapshot 1 - test.mds

Timing Option:actionDuration

No	Action	Accessor	Value	Description	Duration(msec)
1	setConfiguration	'{mds}/Android_Config.json'	'Android'		3
2	start				2187
3	include		" {mds}/New_Account_Creation_WithOut_Phone.js"		0
4	include		"{mds}/Mobile_Locators.mds"		0
5	wait		1000		0
6	click	byXPath(ProfileShop)			0
7	click	byXPath(Tops)	"Jeans and Denims"		0
8	click	byXPath(GOTIT)	5000		0

To enable this function:

1. Navigate to the Scenario Editor.
2. Open a scenario.
3. Navigate to the **Options** tab.
4. Select **Mobile Designer**.
5. Select the **Take Script Snapshot** option.

Release 4.10.4

AIQ 4.10.4 contained the following functional enhancements.

AWS OpenSearch Integration

You can now sign index requests for the AWS OpenSearch service using AWS authentication.

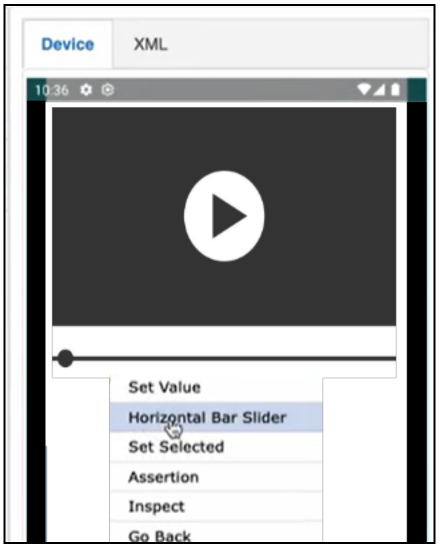
> For more information, see the “Using AWS Authentication protocol” section in the ElasticSearch OpenSearch Integration topic.

Mobile Video Playback

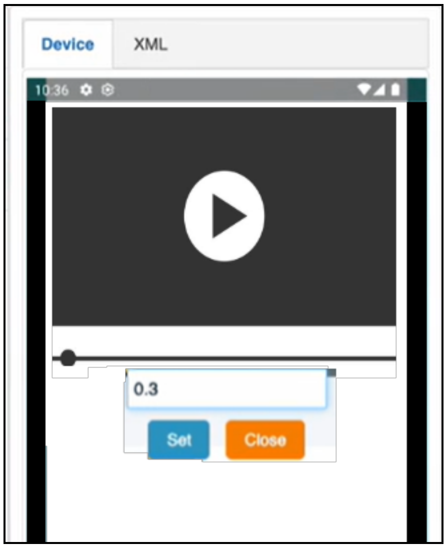
When playing a video on a mobile device, you can now adjust the video playback slider in your test script.

1. Click on the video in the **Device** tab.
2. Right click to display the available options.

3. Select **Horizontal Bar Slider**.

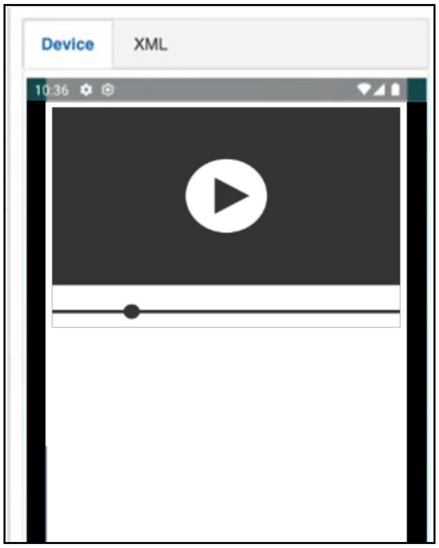


4. Enter a percentage value as a decimal. For example, if you enter .3 the slider will move to the time index that corresponds to 30% of the total video length from the point where the video was paused. To move backwards, enter a negative value.



5. Click **Set**.

6. The video playback slider is adjusted.



Functional Report Enhancements

Enhanced the following aspects of the downloadable Functional Report:

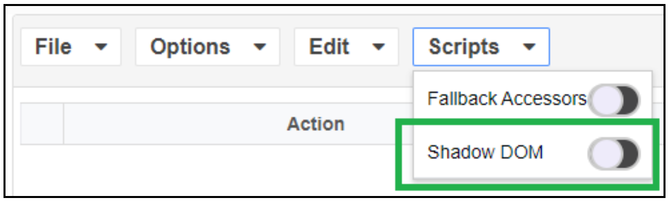
- Test case output log
- Screenshots
- Improved highlighting the clickable elements

Release 4.10.6

AIQ 4.10.6 contained the following functional enhancements.

Shadow DOM Setting

A Shadow DOM setting (`forcedShadowDom`) has been added to the `global.json` file. The default value is `true`. The Shadow DOM setting in the script options in the Test Designer IDE will be ignored if the `forcedShadowDom` is set to true. Even if that option is set to false in the Test Designer IDE, scripts will be run with the option set to true.



Generally speaking, there is no longer any reason for having the Shadow DOM option turned off. Setting it off will not help or prevent any problems or failures. Having it set to off can actually cause failures in instances where a script tries to perform a click action on a element that is under Shadow DOM.

In previous releases, there may have been concerns of some potential performance issues when the setting was on versus the setting being off. Those issues have been addressed in the AIQ 4.10.6 release. The option will be removed completely from the AIQ user interface in a future release.