appvance IQ

Visual Accessors for Android New Feature Spotlight

AIQ 4.9 - Beta Release

Document Version 1.1

March 4, 2023

© 2023 Appvance Inc. All rights reserved

TOC

	1
Visual Accessors - New Feature Spotlight	5
Understanding the Image Matching Algorithm	6
How the Algorithm Works	6
Matching Process	6
Recommendations and Considerations	
Visual Accessors for Android	11
Setting up the Environment	13
Appium Server Setup for Android	14
Device Manager Setup	
Android Studio Setup	
Installing Android Studio	
Configuring Android Studio	16
Setting ANDROID_HOME	17
Setting Android Debug Bridge Location	
Adding App to Emulator	21
Required Fields	
Getting parameters from ADB	
Creating a Smart Tag File for Visual Accessors	
Considerations	
Generating Images for Smart Tags	

Considerations	28
Creating a Mobile Blueprint using Visual Accessors	29
Android Mobile App Example	31
Selecting the Mobile Device and App in the Device Manager	32
Creating a Mobile Configuration File	36
Verifying the Mobile Configuration File	39
Testing the Configuration	39
Getting Access to a Repository	41
Taking Images as Possible Smart Tags	43
Creating the Smart Tag File	45
Mobile Blueprint using Visual Accessors	51

Visual Accessors - New Feature Spotlight

The ability to use visual accessors has been added to the Mobile Test Designer. Visual accessors allow Mobile Designer to incorporate images into your test scripts. Smart Tags Workbench and Mobile Blueprint have been enhanced so that images are defined as Smart Tags and can be added to a Mobile Blueprint. This new functionality is available for both Android and iOS mobile devices.

Mobile Designer enables script recording and playback, a simpler approach to mobile test design than traditional mobile scripting. Using visual accessors allows Mobile Designer to incorporate screenshots into scripts.

This documentation details how the image matching algorithm works, recommendations for getting the best results when using this feature, as well as information about setting up and configuring this new functionality.

> Visual accessors for mobile testing is new functionality that will be introduced in AIQ 4.9. This feature and this documentation are currently in Beta.

Understanding the Image Matching Algorithm

The following section details how the image-matching algorithm determines if the target and source images are a match. There are also some recommendations and considerations for getting the best results when using the visual accessor functionality.

How the Algorithm Works

Naturally, different images will have different numbers of distinguishing features. An example of this is shown below.



TXHZMAW

Desired Features



Q086S9

Low Features

The images in the "Desired Features" set have a significant number of distinguishing features that set them apart from the other images in that set.

The images in the "Low Features" set have a smaller number of distinguishing features that set them apart from other images in that set.

Matching Process

• The first thing the algorithm does is compute the distinguishing feature points for each image (target and source).

 Next, the algorithm applies a matcher to find similar features in both images. For the matcher, a possible set of different algorithms can be applied. In this case, it is a matcher that rejects outlier points that are chosen to reduce the probability of getting an incorrect match.



- Using the set of matched points, the algorithm constructs a region with the same dimensions as the target rectangle bounding.
- The similarity between the target and sub-region source is calculated with several parameters to determine if both are similar. If they are found to be

similar, then the bounding box is returned.



Recommendations and Considerations

Here are some recommendations and considerations for getting the best results when using the visual accessor functionality.

 Multiple reference images for the same target. Along with the originalsized image, you should have several images from the same target with different sizes than the original size. Reference images should also include both images taken from the Mobile Designer IDE along with screen captures. Having reference images from both capture sources is recommended because some bit depth differences can affect results when the target image has limited distinguishing features

- Use images with less compression and better quality. The preferred image format is PNG, but formats such as JPG/JPEG will give good results. PNG is preferred because there is less information loss due to compression. Images with more detail will have a greater number of distinguishing features.
- Test your targets and possible sources. Before creating a full test run, test the set of targets and sources to validate if the correct targets are chosen. You can write a script to perform that task.
- Add context to images to help differentiate them from other possible matches. This can happen if the cropping area is too limited and the number of distinguishing features is limited.

For example, an image with the word "ear" could have multiple matches depending on how the image is cropped. If the image is cropped exactly covering the word, the algorithm will also add "bear", "spear", "tear", "boar" and more as matches. Even "ear" could have "oar" or "eor" as possible matches. So, when possible include some additional context in your image to help with exact matching.

Visual Accessors for Android

The following sections contain information about setting up and configuring this new functionality.

- "Setting up the Environment" on page 13
 - "Appium Server Setup for Android" on page 14
 - "Device Manager Setup" on page 15
 - "Android Studio Setup" on page 16
 - "Adding App to Emulator" on page 21
 - "Required Fields" on page 22
- "Creating a Smart Tag File for Visual Accessors" on page 25
- "Generating Images for Smart Tags" on page 27
- "Creating a Mobile Blueprint using Visual Accessors" on page 29

The following sections contain an end-to-end example of using the visual accessor functionality with an example mobile app.

- "Android Mobile App Example" on page 31
 - "Selecting the Mobile Device and App in the Device Manager" on page 32
 - "Creating a Mobile Configuration File" on page 36

- "Getting Access to a Repository" on page 41
- "Taking Images as Possible Smart Tags" on page 43
- "Creating the Smart Tag File" on page 45
- "Mobile Blueprint using Visual Accessors" on page 51

This is the documentation for the visual accessors feature for testing Android mobile applications. The documentation for testing iOS mobile applications can be found on the <u>Visual</u> <u>Accessors for Mobile Testing</u> page.

Setting up the Environment

AIQ Mobile Designer requires an Appium server and a Device Manager to interact with mobile apps. The Device Manager functions as an emulator or bridge interface for a physical device connected to a computer.

- "Appium Server Setup for Android" on the next page
- "Device Manager Setup" on page 15

Appium Server Setup for Android

Appium is an open source test automation framework for use with native, hybrid, and mobile web apps.

It drives iOS, Android, and Windows apps using the WebDriver protocol.

- 1. Download Appium Server from https://github.com/appium/appium-desktop/releases/tag/v1.22.3-4
- 2. Choose the appropriate installer for your environment.
- 3. Launch the installer.



4. Follow the installation steps and select the default settings.

Device Manager Setup

The Device Manager you use depends on your platform.

- Android applications can use Android Studio IDE. You can download it from: <u>https://developer.android.com/studio.</u>
- IOS applications use Xcode IDE. You can download it from, the Mac App Store or from https://apps.apple.com/us/app/xcode/id497799835?mt=12.

Android Studio Setup

Installing Android Studio

- 1. Download the appropriate installer for your environment from https://developer.android.com/studio.
- 2. Launch the installer.

本 Android Studio Setup	– 🗆 X
	Welcome to Android Studio Setup
X	Setup will guide you through the installation of Android Studio. It is recommended that you close all other applications before starting Setup. This will make it possible to update
	relevant system files without having to reboot your computer.
	Click Next to continue.
Android	
Studio	
	< Back Next > Cancel

3. Follow the installation steps and select the default settings.

Configuring Android Studio

Setting ANDROID_HOME

Once Android Studio IDE is installed, add Global Environment Variable ANDROID_HOME:

- 1. For Windows:
 - a. Navigate to Environment Variables > Settings > System Variables > New.
 - b. Add <code>ANDROID_HOME</code> with value of <code>C://Users/Your_User-</code>

```
/Appdata/Local/Android/ SDK
```

The above value assumes you accepted the default path during installation. If you specified a different path, enter that path for the value of ANDROID_ HOME.

- 2. For macOS:
 - a. Open a Terminal and type: echo export "ANDROID_HOME-E=/Users/yourName/Library/Android/sdk" >> ~/.bash_ profile
 - b. Close Terminal and open a new one and type echo \$ANDROID_ HOME to verify that the value exists.
- 3. For Linux:
 - a. Open a terminal in the same directory of installation and sudo nano .bashrc

- b. Them add to the file: export ANDROID_HOME-E=\${HOME}/Android/Sdk export PATH-H=\${PATH}:\${ANDROID_HOME}/platformtools:\${ANDROID_HOME}/tools
- c. Close the terminal and type in a new one echo \$ HOME

Setting Android Debug Bridge Location

Android Debug Bridge (adb) is a command-line tool that lets you communicate with a device. The adb command facilitates a variety of device actions, such as installing and debugging apps. adb provides access to a Unix shell that you can use to run a variety of commands on a device.

- 1. Open Android Studio IDE.
 - b List ipe is many loss Con Extra But Ap ise 103 Bins But M Aprices (Standardschuldungsschu
- 2. In the Tools menu, select SDK Manager.

3. Copy the SDK Location. "path_copied"

🛎 Elle Edit View Navigate Code Befactor							
MyApplication2) app) src) main) java) cor							Q 🗢
별 💻 Android 👻 😌 호 😤 🏚							
² ✓ Ing app ■ → ■ manifecte							
Com.example.myapplication						Actions	
SecondFragment SecondFragment Concexample.myapplication (andre Concexample.myapplication (test)		Each Android SDK Platform package includes the Android s default. Once installed, the IDE will automatically check for individual SDK components.	slatform and sources pertaining updates. Check "show package i		1.33.x86_64 • risu Google APis x8	33 6 🕨 🖿	
Dir corn: eximple.myapplication BuidConfig End drawable Dir layout End southy_main.comi	Updates Process Elevation Passwords Android SDK Memory Settings			Not installed Installed Installed Not installed Installed			
🚜 content_main.xml		Android 10.0 (Q)		Not installed Not installed			
a fragment secondami ≻ ■ menu		Android 8.1 (Oreo)		Not installed Not installed			
> D m/pmap 2 > D may/gation	Keymap > Editor	Android 7.1.1 (Nougat)		Not installed Not installed			
Values colorsami	> Build, Execution, Deployment Kotlin	Android 6.0 (Marshmallow) Android 5.1 (Lollipop) Android 5.0 (Lollipop) Android 5.0 (Lollipop)		Not installed Not installed Not installed			
2 annens (4) 2 astrings.com 5 > Im themes (2)	> Tools Advanced Settings			Not installed Not installed			
8 → Ditixmi Terminal: Local × + →		Android 4.3 (Jerly Bean)		Not installed			
2 3 Instale la versión más reciente							
PS C:\Users\Diego\AndroidStudioP							
Version Control III TODO O Problems ^ daemon started successfully (today 11:08)						ventiog 🔍 Layou 4 LF UTF-8 4 sp	t inspector aces 🏾 🔒

- 4. Copy the SDK Location. "path_copied"
 - a. For Windows:
 - i. Open a cmd console
 - ii. Type:cd path_copied/platform-tools
 - iii. Type:adb help
 - iv. Verify that it works.
 - b. For Mac:
 - i. Open terminal
 - ii. Navigate to the directory: ~/Library/Android-/sdk/platform-tools/

- iii. Type: adb help
- iv. Verify that it works.
- c. For Linux:
 - i. Perform the same command for the default directory in ~/Android/Sdk/platform-tools/adb or a Terminal type whereis adb and look for the directory

Adding App to Emulator

In Android Studio, the Device Manager interface helps you to configure an Emulator or Physical Device.

- 1. Choose between a Virtual or Physical Device.
- 2. Click Create device and follow the instructions.



- 3. When you finish the process, deploy the app by clicking the Play icon.
- 4. Install the app by using drag and drop from the apk or installing it from the Play Store. If you are using an emulator, be sure to install an Android version with Google Play Services.

Required Fields

The minimum required parameters to set up a configuration file from the app are:

- appPackage
- appActivity
- udid
- platformVersion

The parameters can be obtained depending on the source of the app (self-app, downloaded from the app store)

Refer to "Creating a Mobile Configuration File" on page 36 for information on how to build a file configuration.

Getting parameters from ADB

Parameters can be found using adb:

- 1. Navigate to the adb path. See "Android Studio Setup" on page 16 for information on the adb location.
- 2. Install the app in the emulator or physical device (the emulator can be downloaded from the app store or just drag and drop apk to the emulator).
- 3. Open the app in the Emulator/Physical Device
- 4. In the console/Terminal type: adb shell pm list packages to obtain the package name "appPackage" (search for it in the list with a name similar to the app, for instance, com.mobile.myapp).

5. For main activity "appActivity" type: adb shell dumpsys package com.mobile.myapp

Example result: adb shell dumpsys package com.overdrive.mobile.android.libby

Activity Resolver Table: Schemes: dewey-oauth: 3969bfc com.overdrive.mobile.android.libby/com.overdrive.mobile.android.nautilus.ui.Activity_Main filter f2f78a6 Action: "android.intent.action.VIEW" Category: "android.intent.category.DEFAULT" Category: "android.intent.category.BROWSABLE"

extractNativeLibs=false primaryCpuAbi=null secondaryCpuAbi=null cpuAbiOverride=null versionCode=520000 minSdk=21 targetSdk=33 minExtensionVersions=[] versionName=5.2.0 usesNonSdkApi=false splits=[base] apkSigningVersion=2

.....

UDID can be found just by typing: adb devices

Example: adb devices

List of devices attached emulator-5554 device

Creating a Smart Tag File for Visual Accessors

Visual accessors use a new type of Smart Tag called Image Reference. It can be provided from a URL or a repository.

- 1. In AIQ navigate to **Mobile Designer > Smart Tags Workbench**.
- 2. In the Category field, select Image Ref.
- 3. Enter a **Smart Tag Name**.

Appvance IQ : Mobile Smart Tag: x	Appvance IQ : Output Logs X +						~	-	0	
→ C	lobileDesigner/SmartTags.html?path=					07	© @ ¢	*		D
appvance.ai									4	5
Dashboard	File *		Summary	Smart Tags	Actions	Extractions	Validation	s	ettings	
Test Designer K										
tobile Designer	Smart Tags 🕂	Category				Add	Duplicate			
105		 Image Ref 	Disable							
iDC	Smart Tag Name	Smart Tag Type								
JS Edit & Play		Navigation		~						
Smart Tags Workbench	Note: Smart Tag name should begin with a letter a	nd not contain spaces.								
Validations Workbench	Smart Tag Description									
Mobile Config	Add a description for your Smart Tag									
I Scripting K										
ervice Suite <	Scan Mode Select an option below (recommended). In the	a ovent no ontions are selected a r	ew Smart Tao acti	on will be added i	work time a	s Smart Tag ie i	dentified			
cenario Editor		e event no opeons are senected, a r	ew omant hag acts	at to	overy unite e	r officiant mights in	dentined.			
lay	One Per Page Scan only first matching element for each acce. Example 4 Home butten present	Once Per App Scan all matching elements from	n only one	Singleton Scan an element	which only	exists once				
nalyze	across many pages.	exists across many pages.	rwnich	elements.	cation. Exar	mpie: Login				
PL Options <										
raffic	Images 🕂									
Slobal Options <	Image URL									
ost:8080/UI/MobileDesigner/SmartTags.htm	d?path=#		Browse	Show Image	emove					

- 4. Click on the plus next to Images.
- 5. You can enter a URL for the target in the **URL** field, or use the **Browse** button to upload an image from an existing repository.
- 6. To add more Smart Tags in the same file, click **Add** in the top right section and repeat the same steps. When all Smart Tags are created, save them in a file by clicking on the file menu.

Considerations

- Smart Tag images can have as many targets as necessary, just click on the plus icon next to **Images** to add more fields.
- if just a URL is provided it must be public access.
- If you are using a repository, it must be configured in AIQ, Navigate to
 Global Options > Preferences > Repositories.

asi Appvance IQ : Preferences	×	a localhost	× +						\sim		0
← → C (① localhos	st:8080/UI/p	references.html						on <u>B</u> j	iể ☆	* 0	0
≡ appvance.ai	I								-	â appvan	œ
8 Dashboard	^	Repository Reports	APM Integration	Git Credentials	EC2 Credentia	Is AIQ Credentials					
Test Designer	<										_
Mobile Designer	۰.	Select	✓ Refresh	ast updated (yyyy-	MM-dd):		Pull selected repository	Checkout Branch	Clone	e Reposito	ry
& Al Scripting	<	Туре		Name (Case ins	ensitive)	Branch:		Explore	Delete	Clear	
I Service Suite	•	GIT	~	test_name							1
▶ Play		URL					Visibility				
<u>al</u> Analyze		https://myurl.git					Private ~				
DPL Options	<	Username		Password / SSI	H (If your account	uses 2FA provide the acc	ess token instead of the password)				
al Traffic		my_user_name		Password / To	ken 👻						
0 Global Options	~										
Output Logs	- 1	Proxy Options						1			
AIQ Status	- 1	Select an option	~								
Preferences	- L	Default Commit N	lessage			Status:					
Cloud Config											
Monitor Config		Stacktrace:									

Generating Images for Smart Tags

The Mobile Designer IDE allows you to play and record actions, besides cropping images from the device screen.

For taking images for your Smart Tags:

- 1. Open an emulator and start an Appium Server.
- 2. In AIQ navigate to **Mobile Designer**.
- 3. Click Record.
- 4. Load a file configuration. See Mobile Blueprint Enhancements for information on file configurations.
- 5. Click and navigate through the application and crop the images by drawing rectangles.



6. Click **Save** and save the images in your repository or local drive.

Considerations

- It could be helpful to add multiple images from the same target taken using different methods at different sizes (preferably bigger than the original size).
- You can use the emulator to take a screenshot and crop images by using an image editor or just you could use basic cropping functionality from the OS.
- Images must be saved in a repository or a public URL.

Creating a Mobile Blueprint using Visual Accessors

- 1. In Appvance IQ navigate to **AI Scripting > Create Blueprint**.
- 2. Click **Mobile** to create a mobile blueprint.

← → C ⁴ (0) Incellingt 90000	as Apprante IQ UserGuide - Mobil: X + Edimential Conta hard	✓ - 0 Х с № ю с • П 0 ;
≡ appvance.ai	Annahusin Ceanstruis	appvance 🗸
& Dashboard	File V Options V New Bicasprint	0 0 0 0 0 00:00:00 Browsers Pages Found Actions Found Remaining Actions Duration (hh.mm.ss)
Test Designer	Basic Execution Management	
Mobile Designer	Mobile	Web
& Al Scripting Y		
Create Blueprint		
Compare Blueprints		
Blueprint Template Editor		
Generate Scripts		
Coverage Map	Version: 300	
Service Suite		
🗲 Scenario Editor		
► Play		
Lal Analyze		
DPL Options <		
d Traffic		
Of Global Options <		

- 3. Load your device configuration file. Click **Browse**, search for the file and load it.
- 4. In Scan Type choose Smart Tags or Smart Tags + Tags and click Next.
- 5. Create a new execution and give it a name.
- 6. After clicking **Create**, the Blueprint will be running.

ecution Name:	a time		
Default	×	Variable	Value
		Not Set	Set a value
		Not Set	Set a value
	*	Not Set	Set a value
otal Selected Datasets: 1		Not Set	Set a value
Select All Deselect All		Not Set	Set a value
/pe		Not Set	Set a value
Hash DPL - Reads CSV, provid V		Not Set	Set a value

7. Once the Blueprint finishes, the execution results display.

🚥 Appvance IQ : Blueprint 🛛 🗙	🚥 Appvance IQ U	UserGuide - Mobile 🗙 🏻	+									~	-	0	×
\leftrightarrow \rightarrow C (localhost:8080/UI/	Blueprint/Create	html									04	Bi le ≴	*	•) :
≡ appvance.ai													4	appva	nce
4 Dashboard	File 👻	Options -	New Blueprin					1 Devices	10 Pages Four	nd Actions	ound Remai	0 ning Actions	00: Duration	:07:04 (hh:mm:	55)
Test Designer (Basic	Execution Managen	ent (Mobile) Requ	ests Reports	Validatio	ns Smart	Tags					N	lot Runr	ning
Mobile Designer		ataset Manager	+ Add Exec	ution											
& Al Scripting ~	Define	input data for the B	ueorint usir	n the Data	set Manager, C	lick Innuts Ec	und link to sh	ow all found i	nages with in	outs to be de	fined				
Create Blueprint	Map de	esired inputs with co	rresponding	g variables	defined in Datas	sets and then	click the Exe	cution's Play I	outton to cont	inue Bluepri	nt.				
Compare Blueprints		Name	Contro	ol Panel	Dataset	Devices	Pages Found	Actions	Remaining	Duration (hb:mm:ss)	Status	Failed	Incon	sistent	
Blueprint Template Editor										(mining)		(1)			
Generate Scripts		test		Found	1 Datasets	0	10	216 JS 🗸 📥	0	00:07:04	Not Running	C Retry			
Coverage Map															
In Service Suite															
F Scenario Editor															
▶ Play															
Lat Analyze															
IDPL Options <															
al Traffic															

Android Mobile App Example

The following sections provide a complete end-to-end example of using the visual accessor functionality to test a mobile application.

The Android version of the Universe mobile app by Alpha Software is used for the example.

Selecting the Mobile Device and App in the Device Manager

In this section you will select a mobile device to emulate and install the example mobile app. This example uses a Pixel 2a as the emulated mobile device.

- 1. Open Android Studio.
- 2. Click on **Device Manager > Create device> Virtual**.
- 3. Select Category > Phone.

In the Name column select Pixel 2a with Play Store 5.6" Resolution: 108 2210, 440dp.

Note: Make sure the device has a Play Store icon (blue square) associated with it.

📥 Eile Eo	tit <u>V</u> iew <u>N</u> avigate <u>C</u> ode <u>B</u> efa	ctor <u>B</u> uild Ryn <u>I</u> oc	ols VC <u>S W</u> indow <u>H</u> elp	My Applica	tion (C:\Users\	Diego\AndroidSt	udioProjects\My	Application2]\Diego\D	Desktop\TestBluePrint\M	lobileTesting\/mage	sDocumenta	tion\/ibby5.png	-	٥	×
C:) Users)															۰ 🗉
별 🛎 And		ee				hhidang 🗶 💊									
2 •	th we we DS 20 4	∎ 61 50 ÷				• — .olor) 199	16 B Virtual								, see
2 >				`											0
Anna State												Size on Disi	Actions		2
ara 🕹															: Mani
		🛛 🚺 🛛 Sel	lect Hardware												. 9
														1	
Ň															
		Choose a devi	ice definition												
>								Pixel 2							
>															
×		Tablat		8				- 1080pr -							
									Ratio: long						
2								5.0° 1920px							
Structa															
-															
4 •															
-73 → AR ² Gr															
Terminal:															
Sec. 1															
B PS C:\	- Vi														
2															
₽ versis															
E Failed to															. н

- 4. Click Next.
- 5. In the **Release Name** option select Tiramisu and click **Next**.

× `	intual Device Configuration System	Image e	~		×
	Recommended x86 Ima	iges Other Imagi			Tiramisu
	Release Name				
	TiramisuPrivacySa 🛓	TiramisuPrivacySan	x86_64	Android API TiramisuPrivacy	API Level
	Tiramisu			Android Tiramisu (Google F	33
					Android
					Tiramisu
					Google Inc.
					x86_64
					We recommend these Google Play images because this device is compatible with Google Play.
?)				Previous Next Cancel Finish

Note: The Android version of the emulator displays in the Device Manager.

- 6. Accept the default settings. Be sure that Portrait is selected.
- 7. Click Finish.

-1

8. Start the emulator by clicking the **Play** button.





Note: Device Manager, Play button, and Emulator are highlighted in red.

9. Click at the center of the phone emulator and drag it up to show the available apps. Choose the Play Store.



Note: The emulated phone menu, Play Store highlighted in enclosed in red square.

10. Search Play Store for the Universe app by Alpha Software and Install it.



You are now ready for the next step, "Creating a Mobile Configuration File" on the next page.

Creating a Mobile Configuration File

A Mobile Configuration file is required to describe which app will be launched during the Play/Recording/Blueprint.

1. In AIQ navigate to **Mobile Designer > Mobile Configuration**.

🛥 Appvance IQ : Mobile IDE 💦 🤤	< +			v – C
← → C (O localhost 8080	/UI/MobileDesigner/config.html#			er 윤 ☆ 🖈 🖬
≡ appvance.ai				🛔 appvano
& Dashboard	*			
Test Designer	Configurations	Name		ම Config ⊡ Open 🛱 Save 🛱 Save As 🖒 Clea
· · · · · · · · · · · · · · · · · · ·	Select configuration	Configuration name		
Mobile Designer	Service	Platform	Source	Physical Device
IDE	On-Premises	Android (Native)	✓ Package	~ (
JS Edit & Play				—
Smort Togs Markhopsh	Package	Activity	Appium Server URL	Proxy URL
Smart rags workbenon	Package	com.example.MainActivity	http://localhost:4723/wd/hu	ub http://localhost:9090
Validations Workbench	Valid Base URLs		Capabilities	
Mobile Config	https://www.example1.com.https;	/www.example2.com	Select capability	~ +
Al Scripting				
	Capabilities (Required)		Capabilities (Extra)	
Service Suite	Name	Value	Nerve	Velue
Scenario Editor	pletformVersion	(Android Version)		
Play	udid	(identifier)		
- Flay	systemPort	(8203, or any value between 8200	- 8299 unique t	
il Analyze	deviceName	(model, can be any name)		
EDPL Options <				
al Traffic				

2. The first step is to get the basic parameters from the Device. Open a Console/Terminal in the adb directory.

> See "Setting up the Environment" on page 13 for information on how to get the directory if a variable path is not assigned to it.

- 3. Open the Universe app in the Device Emulator and perform the following steps:
 - a. UDID: adb devices

List of devices attached emulator-5554 device

So: UDID=emulator-5554

b. appPackage: adb shell pm list packages

package:com.android.traceur package:com.android.contacts package:com.google.android.apps.messaging **package:com.ed.universe** package:com.android.internal.emulation.pixel_3a_xl package:com.android.location.fused package:com.android.vpndialogs

It returns a list with active packages, and searches for the app name, in this case, Universe (appPackage=**com.ed.universe**).

 c. Activity/appActivity: adb shell dumpsys package com.ed.universe

Activity Resolver Table: Non-Data Actions: android.intent.action.MAIN: 237c878 com.ed.universe/.MainActivity filter 5a51b51 Action: "android.intent.action.MAIN" Category: "android.intent.category.LAUNCHER" .DayInDetailActivity: 7457fb6 com.ed.universe/com.edu.sundroid.DayInDetailActivity filter 24504b7 Action: ".DayInDetailActivity" Category: "android.intent.category.DEFAULT" versionCode=37 minSdk=21 targetSdk=32 minExtensionVersions=[] versionName=1.7.4 usesNonSdkApi=false splits=[base, config.arm64_v8a, config.en, config.xxhdpi] apkSigningVersion=3

appActivity=.MainActivity

Also, AndroidVersion appears as targetSDK = 32 (it can be an upper version if OS is supporting it)

- 4. Now fill the Mobile Configuration Panel with the parameters
 - a. Name the configuration.
 - b. In Source select Package.
 - c. In Capabilities add "noReset": "true" to prevent the app from asking repeatedly to log in.
 - d. Accept the default parameters for the remaining fields.
 - e. Save the configuration as "UniverseConfiguration".

Appvance IQ : Mobile ID	E X	() Index of REPOSITORY/DemoRep	× +				× -	0
\leftrightarrow \rightarrow \mathcal{O} (0 local	host:8080/UI	/MobileDesigner/config.html					아 순 ☆ 🌟	
≡ appvance.o	ai						≜ a	ppvance
8 Dashboard	A							
Test Designer	۰.	D UniverseConfig.json						
Mobile Designer	<	Configurations UniverseConfig	Name			O Con	lig 🗅 Open 🛍 Save 🛱 Save As	🖒 Gear
AI Scripting	<	Auto	No. Const.		Anne			
Service Suite	<	On-Premises	Android (Native)	~	Package	~	Physical Device	
Scenario Editor		Bickane	Artivity		Annium Server I IRI		Prove LB1	
Play		com.ed.universe	MainActivity		http://localhosts4723/wd/hub		http://iocalhost:9090	
L Analyze		Valid Base URLs			Capabilities			
DPL Options	<	https://www.example1.com.http:	://www.example2.com		Select capability	~	+	
Traffic	- 1	Capabilities (Required)			Capabilities (Extra)			
Global Options	~	Name	Value		Name	Velu	e	
Code al la ser		platformVersion	32	<u>^</u>	noReset	true		e
Output Logs		udid	emulator-5554					
AIQ Status		systemPort	8201					
Preferences	- 1	deviceName	Device					
Cloud Config								
Monitor Confia	-			~				

Verifying the Mobile Configuration File

The file configuration can be opened in text format:

```
[
    {
        "configName": "UniverseConfig",
        "app": "",
        "appPackage": "com.ed.universe",
        "proxyURL": "",
        "bundleId": "",
        "source": "package",
        "appActivity": ".MainActivity",
        "deviceMode": true,
        "service": "On-Premises",
        "serverURL": "http://localhost:4723/wd/hub",
        "platformName": "Android",
        "validDomains": "",
        "capabilities": {
            "platformVersion": "33",
            "udid": "emulator-5554",
            "systemPort": "8201",
            "deviceName": "Device",
            "noReset": "true"
                                       }
    }
]
```

Testing the Configuration

For testing, if the configuration is correct,

- 1. Start an Appium Server and an active emulator.
- 2. In Appvance IQ navigate to **Mobile Designer > IDE**.
- 3. In the File dropdown select New.

- 4. Click Record.
- 5. Add the configuration you created. Wait until the app is launched.
 - If there is a problem with the configuration, it will display in the log or the Appium Inspector. Use the log file to fix any configuration issues. Typically errors are related to incomplete names or copied parameters. Use the example file above to help.
 - If there are no configuration errors, you will see the emulator with the Universe app running in AIQ.



You are now ready to proceed to the next step "Getting Access to a Repository" on the facing page.

Getting Access to a Repository

In order to save images Appvance IQ can connect to a repository.

1. In AIQ navigate to Global Options > Preferences > Repository.

Approace RL: Preferences	×	 moex or kePOSITORI/Dem 	ionep A T									0
\leftrightarrow \rightarrow C \odot localh	ost:8080/UI/	preferences.html							07 Bjj	ピ ☆	*	п.
appvance.c	ai										å app	vance
B Dashboard	î.	Repository Repo	orts APM Integration	Git Credentials	EC2 Credentia	s AIQ Credential	8					
K Test Designer	<											
Mobile Designer	<	DemoRepo	✓ Refresh La	ist updated (yyyy-N	(M-dd): 2022/11/0	9 14:35:29		Pull selected repository	Checkout Branch	Clor	e Repo	sitory
& Al Scripting	<	Туре		Name (Case inse	insitive)	Branch: M	AIN		Explore	Delete	Cle	ar
Service Suite	<	GIT	~	DemoRepo								
Scenario Editor												
Play		URL					Visit	ility				
M Analyze		https://gitlab.o	com/AIQBlueprints/mobilebl	ueprint.git			Pu	blic v				
DPL Options	<	Username		Password / SSH	(If your account u	ses 2FA provide the a	ccess tol	en instead of the password)				
al Traffic		dcastro		Password / Tol	ken 🗸							
Clobal Options	~											
Output Logs	- 1	Proxy Options										
AIQ Status		acteur an opur										
Preferences		Default Comm	it Message			Status:						
Cloud Config		DS3				WORKIN	G					
Monitor Config		Stacktrace:										

- 2. Suggested parameters:
 - Name: DemoRepo
 - URL: <u>https://gitlab.com/AIQBlueprints/mobileblueprint.git</u>
 - Username: user name with access to the repository
 - Password: user password, if not working please contact Mobile Designer Team for a Token
 - Visibility: Public
 - Type: GIT

Those parameters can be changed if the user has access to another repository.

- 3. Click **Clone Repository** (Repository Successfully saved if everything is correct, if not check your parameters)
- 4. Anytime a file or document is saved or loaded it can be accessed from the repository in ROOT > REPOSITORY > name_repo.

Root / Repository					
Quick Access HOME	REPOSITORY Name	Modifie	Q Search	Size	
Image: Continue of the second sec	 DemoRepo HOME 	11/09/2 08/22/2	2022 14:42		
File Name			✓ Oper	X Canc	el

You are now ready to perform the next step, "Taking Images as Possible Smart Tags" on the facing page.

Taking Images as Possible Smart Tags

Visual accessor functionality allows you to use graphical representations as possible Smart Tags. Images from the application can be obtained from Mobile Designer IDE in the Recording.

Follow these steps to get some images from the Universe app.

- 1. Start an Appium Server and an Emulator.
- 2. In AIQ navigate to **Mobile Designer > IDE**.
- Click Record and load the configuration file that you created in "Creating a Mobile Configuration File" on page 36.



4. Wait until it loads and then draw a square around the HOME button.

- Click Save and search for ROOT > REPOSITORY > DemoRepo > Targets and save it as "UniverseHome".
- 6. Click on Home.

7. Draw a square around the Rocket image and save it as "UniverseRocket".

- 8. Click on **Content from Astroverse** located above Rocket Image.
- 9. Draw a square around the Solar System image and save it as "UniverseSolarSystem".

Appvance IQ : Mobile IDE X	Index of R	EPOSITORY/DemoRep: × +						· - 0
	vwooneoreagi	wyidentinine-#					01 Q	
ashboard	File -	Options - Edit	▼ New*			0	Burley M.O.	
est Designer			Record	► Play II Pause	Stop C Step Over			• 4 •
nile Designer	No	Action	Accessor	Value	Description		< Astrove	rse ø
DE	3	Start	byld("com.ed.universe:id/home"					
Edit & Play	4	Click	byAccessibilityId("Astroverse")				串	<u></u>
nart Tags Workbench	5	Click	byld("com.ed.universe.id/home"				Current location of	
Idations Workbench	7	Click	by/accessibility/d("Astroverse") by/d("com.ed.universe:id/home"		_		the ISS	Day in detail
bile Config	8	Click	byAccessibilityId("Astroverse")					
cripting <	9	Click	byld("com.ed.universe.id/home"					
ice Suite	10	Data	byxPath("//android.widget.TextV	iewi@sext=Content f		Significant Person	Answers to big questions	
mario Editor	Adding	Action: setConfiguration						
	Step St Step Co	arted: setClientType And mpleted	roid					
1/20	Step St Step Co	arted: setServerURL http: mpleted	//localhost:4723/wd/hub				Save	
Onlines (Step St Step Co	arted: setCapabilities (' mpleted	appActivity":".MainActivity",";	erse","moReset":"true","pl:		Core	tour printos	
	Step St Startin	g Android Client.						202
C	4				*		2.5X	_^

You are now ready for the next step, "Creating the Smart Tag File" on the facing

Creating the Smart Tag File

The following process demonstrates creating a Smart Tag File for the Universe app.

Note: You must have a connection to the repository to perform this task. See "Getting Access to a Repository" on page 41.

- 1. In AIQ navigate to **Mobile Designer > Smart Tags Workbench**.
- 2. Define the following parameters:
 - Smart Tags Name : Home
 - Category : ImageRef
 - Smart Tag Type : Navigation

Agreened & Made Smart Rig X index # MPG051000-conduct: X +	~	- 0					
\leftrightarrow \rightarrow C (0 localhost:8080/UI/N	AobileDesigner/SmartTags.html?path=			0	• 🖩 🖻 🖈	* 🗆	1
≡ appvance.ai						🛎 appvan	ce -
Image of Agreement (2) Media Same Tigs Image of Agreement (2) Media Same Tigs Image (4) Image (4) Image (4) Image (4)	Su	mmary Smart Tags	Actions Extractions	Validations	Settings		
Test Designer <	Smart Taos 🕇	Optiont/conversion: Image: Ref Image: Ref Image: Ref <					
C Mobile Designer		• Image Ref • O	lisable	Add	Dupicale		
IDE	Smart Tag Name	Smart rag type					
JS Edit & Play	Home	Navigation	~				
Smart Tags Workbench	Note: Smart Tag name should begin with a letter at	nd not co ntain spaces.					
Validations Workbench	Smart Tag Description						
Mobile Config	Add a description for your Smart Tag						
& Al Scripting <							
I Service Suite <	Scan Mode Select an option below (recommended). In the	event no options are selected, a new Smart T	ag action will be added	l every time a Smart Tag is	identified.		
🖌 Scenario Editor	One Per Page	Once Per App	Singleton				
Play	Scan only first matching element for each page. Example: A Home button present	Scan all matching elements from only one page. Example: A navigation bar which	Scan an elemen in the entire app	nt which only exists once elication. Example: Login			
Lall. Analyze	across many pages.	exists across many pages.	elements.				
EI DPL Options <							
al Traffic	Images +						
of Global Options	Image URL	Brow	rse Show Image	Remove			

3. Click on the plus sign next to **Images**.

- Click Browse and search in ROOT > REPOSITORY> DemoRepo > targets > UniversalHome.
- 5. Click **Show Image** to check if the correct image is loaded.

🚥 Appvance IQ : Mobile Smart Tag: 🗙	Index of REPOSITORY/DemoRep: x +	× - 0
← → C (O localhost:8080/UI/	MobileDesigner/SmartTags.html?path=	아 책 순 ☆ 🛊 🖬 😩
≡ appvance.ai		👗 appvance
& Dashboard	Note: Smart Tag name should begin with a letter and not contain spaces.	
K Test Designer	Smart Tag Description	
Mobile Designer	Add a description for your Smart Tag	
IDE		
JS Edit & Play	Scan Mode Select an option below (recommended). In the event no options are selected, a new Smart Tag action will be added every tim	e a Smart Tag is identified.
Smart Tags Workbench	One Per Page Once Per Ann Singleton	
Validations Workbench	Scan only first matching element for each Scan all matching elements from only one Scan an element which on	ly exists once
Mobile Config	page. Example: A Home button present page. Example: A navigation bar which in the entire application. Example: A navigation bar which in the entire application. Example: A navigation bar which exists across many pages. elements.	:ample: Login
Al Scripting		
Service Sulte	Images +	
Scenario Editor	Image URL	
▶ Play	http://localhost:8080/AppvanceServer/rest/public/appvance/REPOSITORY/DemoReport Browse Hide Image Remove	
al Analyze		
E DPL Options (A HOME	
al Traffic		

- 6. Click Add.
- 7. Click on the plus sign next to **Smart Tags**.

oa Appvance IQ : Mobile Smart	Tag: X	Index of REPOSITORY/DemoRep: × +							~	- 0	o >
$\leftarrow \rightarrow $ C O localhos	:8080/UI/N	AobileDesigner/SmartTags.html?path=						ov <u>2</u> ∦ Q	🖻 ✿	* 0	
≡ appvance.ai										🔺 appv	vance -
@ Dashboard	^	File -			Summary	Smart Tags Mobile (1)	Actions	Extractions	Validations	Settin	ngs
Test Designer	<	Smart Taga 🔸		Category				-	Durkente		
D Mobile Designer	~		•	Image Ref	 Disable 			~~~	CULACON		
IDE		Smart Tag Name		Smart Tag Type		v					
JS Edit & Play		Note: Smart Tag name should begin with a le	fler and not contain spe	ices.							
Smart Tags Workbench		Smart Tag Description									
Validations Workbench		Add a description for your Smart Tag									
Mobile Config		Scan Mode	ment on ordinar are co	lastad a pour Smart Tag asti	no uill he added avec time o	Passet Top is identified					
& Al Scripting	<	One Per Page	Once Per Ann	iecieo, a new smart rag act	Singleton	Smart rag is identified.					
III Service Suite	۰.	Scan only first matching element for each page. Example: A Home button present	Scan all matching e page. Example: A r	elements from only one avigation bar which	Scan an element which o in the entire application. E	nly exists once comple: Login					
✤ Scenario Editor		across many pages.	exists across many	pages.	elements.						
► Play		Images +									
Let Analyze		Image URL									
IDPL Options	< -	http://localhost:8080/AppvanceServer/rest/pi	ublic/appvance/REPOS	ITORY/DemoRepo/targets/	Jnh Browse Hide Ir	nage Remove					
al Traffic											
	٣	•									

8. Create a new Smart Tag named "Rocket" in the **Smart Tags Name** field.

- 9. Click the plus sign next to **Images** and search for "UniverseRocket" in the repository.
- 10. Click Add.
- 11. Create a new SmartTag named "SolarSystem" and browse for the "UniverseSolarSystem" image.

Appvance IQ : Mobile Smart	Teg: X	Index of REPOSITORY/DemoRep: x +					\sim	-	0	×
$\leftarrow \rightarrow C$ (O localhos	t:8080/UI/Mol	bileDesigner/SmartTags.html?path=	0	- Ge	Q	ß	$\dot{\mathbf{T}}$	*	0.0	8 E
≡ appvance.ai								4	appvant	xe -
& Dashboard	*	SolarSystem Navigation ~								
Test Designer	κ.	Note: Smart Tag name should begin with a letter and not contain spaces. Smart Tag Description								
D Mobile Designer	~	Add a description for your Smart Tag								
IDE		Scan Moria								
JS Edit & Play		Select an option below (recommended). In the event no options are selected, a new Smart Tag action will be added every time a Smart Tag is identified.								
Smart Tags Workbench		One Per Page Once Per App Singleton Scan only first matching element for each Scan all matching elements from only one Scan an element which only exists once								
Validations Workbench		page. Example: A Home button present page. Example: A navigation bar which in the entire application. Example: Login across many pages. exists across many pages. elements.								
Mobile Config										
& Al Scripting	<	Images +								
I Service Suite	<	Image URL http://localhost.8080/AppvanceServer/rest/public/appvance/REPOSITORY/DemoRepoltargets/Unix Biowse Hide Image Remove								
🖌 Scenario Editor										
▶ Play										
Lat Analyze										
DPL Options	<									
al Traffic										

- 12. Click Add.
- 13. In the File dropdown, select **Save** and save it as "UniverseSmarttagsDemo".
- 14. The saved file has the following structure:

{

```
"categoryId": "ImageRef",
"threshold": "0.75",
"smartTags": [
{
```

```
"name": "Home",
                                        "category": "ImageRef",
                                        "type": "Navigation",
                                        "tags": [
                                                http://-
loc-
alhost:8080/Ap-
pvanceServer-
/rest/pub-
lic/ap-
pvance/REPOSITORY/DemoRepo/targets/universeHome.png"
                                ],
                                "rules": [],
                                "info": {},
                                "disabled": false,
                                "description": "",
                                "singleton": false,
                                "scanOncePerApp": false,
                                "scanOnePerPage": false
                       },
                        {
                                "name": "Rocket",
                                "category": "ImageRef",
                                "type": "Navigation",
                                "tags": [
                                "http://-
loc-
```

```
alhost:8080/Ap-
```

```
pvanceServer-
/rest/pub-
lic/ap-
pvance/REPOSITORY/DemoRepo/targets/UniverseRocket.png"
                               ],
                               "rules": [],
                               "info": {},
                               "disabled": false,
                               "description": "",
                               "singleton": false,
                               "scanOncePerApp": false,
                               "scanOnePerPage": false
                       },
                       {
                               "name": "SolarSystem",
                               "category": "ImageRef",
                               "type": "Navigation",
                               "tags": [
                               "http://-
loc-
alhost:8080/Ap-
pvanceServer-
/rest/pub-
lic/ap-
pvance/REPOSITORY/De-
moRepo/targets/UniverseSolarSystem.png"
                               ],
                               "rules": [],
                               "info": {},
```

```
"disabled": false,
    "description": "",
    "singleton": false,
    "scanOncePerApp": false,
    "scanOnePerPage": false
    `}
    ],
    "actions": [],
    "customTags": [],
    "validators": [],
    "extractors": []
```

You are now ready for the next step "Mobile Blueprint using Visual Accessors" on the facing page.

Mobile Blueprint using Visual Accessors

This section demonstrates creating a Mobile Blueprint using visual accessors for the Universe app.

- 1. Start an Appium Server and an Emulator with the Universe app installed.
- 2. Open the app in the Emulator to verify everything is working correctly.
- 3. In AIQ navigate to **AI Scripting > Create Blueprint**.

- 4. Click Mobile.
- Select the configuration file for your device. Click Mobile Configuration
 >Browse and load "UniverseConfiguration".

6. In Scan Type select Smart Tags.

7. Click Browse next to Smart Tags Library Path and load the "Uni-

verseSmartTagsDemo" file.

ď	"UniverseSmartTa "Creating the Sma	agsDemo" is art Tag File" c	the file you cr n page 45.	eated in	
😅 Appvance KQ.: Blueprint	x 🚥 Appvance IQ UberGuide - Mabin x +			~ - 0 X	
$\leftarrow \rightarrow $ C (localhost	8080/UI/Blueprint/Create.html		ං දී ලි	• 🖈 🗯 🖬 🕕 🗄	
≡ appvance.ai				🚢 appvance 🗸 🚔	
& Dashboard	File - Options - New Blueprint	0 Devices	0 0 0 Base Found Actions Found Remaining Actions		
Test Designer	Basic Execution Management (Mobile)	Centes	Crages round (Actions round) (Remaining Actions	Constant (mannessy)	
Mobile Designer	<			Next	
& Al Scripting	Service: Platform:	Source:	Scan Type:		
Create Blueprint	On-Premises ~ Android (Na	ve) v Package	✓ Smart Tags	~	
Compare Blueprints	_				
Blueprint Template Edito	r Visual Hints 🕥 Validations Library	Al Hints Files	Setup		
Generate Scripts	Smart Tags Library Path:			Browse	
Coverage Map	Package:				
Service Suite	< Package				
⊁ Scenario Editor	Activity:				
► Play	MainActivity				
Lat Analyze	Valid Base URLs: https://www.example1.com.https://www.example2.com				
DPL Options	<				
.al Traffic	Physical Devic	·		Rouse	
08 Global Options	<			browse	

- 8. Click Next.
- 9. Create a new execution with the name "UniverseDemo".
- 10. Click **Create**The Blueprint will run.

ecution Name:	time			
Default	× ^	Variable	Value	
		Not Set	Set a value	
		Not Set	Set a value	
		Not Set	Set a value	
tal Selected Datasets: 1		Not Set	Set a value	
Select All Deselect All		Not Set	Set a value	
pe		Not Set	Set a value	
nasir or c • reaus 0.3V, provid 🗸		Not Set	Set a value	

11. To view the search while the Blueprint is running refer to the Emulator.

12. Result : List of pages and actions found.

asi Appvance IQ : Blueprint	×	Index of F	REPOSITORY/DemoRepo	+									~	- 0
← → C ① localhos	t:8080/UI/	Blueprint/Cre	ate.html#									0v 🔤	ର୍ଜ ★	* 🗆 🛎
≡ appvance.ai														🔺 appvan
8 Dashboard	ŕ	File +	Options 👻 📑							0 Devices	3 Pages Found	4 Actions Found	0 Remaining Actions	00:01:31 Duration (hh:mm
Test Designer		Basic	Execution Managemen	nt (Mobile)	Requests	Reports Valia	dations Sma	rt Tags						Not Runn
Mobile Designer	<		Sataset Manager 🔰 🕇 Add	Execution										
& Al Scripting	~	Define Map d	input data for the Blueprin esired inputs with correspo	t using the De Inding variable	ataset Manager es defined in Da	Click Inputs Found tasets and then click	link to show all fo the Execution's I	und pages with Play button to co	inputs to be definent	id.				
Create Blueprint			Name	Con	trol Panel	Dataset	Devices	Pages Found	Actions Found	Remaining Actions	Duration (htcmmoso)	Status	Failed Actions	Inconsistent Actions
Compare Blueprints			UniverseDerno	E	Inputs Found	E 1 Detasets	D	4	4.15 🕶 📥	0	00:01:31	Not Running		
Blueprint Template Editor	_						Pages with Inputs	All Pages	~					
Generate Scripts		00 [1 0 00 [1 0	Dustom Actions) International Spac Dustom Actions) Astroverse	e Station										
Coverage Map	_	(1)	Custom Actions) Astroverse Custom Actions) Blank											
Service Suite	¢							1 to 10 👻 Page	1					
✗ Scenario Editor														
Play														
al Analyze														
DPL Options														
al Traffic														

13. Result : Home Smart Tag found.

osi Appvance IQ : Blueprint	×	Index of REPOSITOR	RY/DemoRepo	+									~	-	0	
← → C (O localhos	st:8080/UI/B	lueprint/Create.html	¢									04 B) Q 🖻 🗯	*		1
≡ appvance.ai														4	appvar	c
8 Dashboard	^	File • Option	ns 💌 🧧							0 Devices	3 Pages Found	4 Actions Found	0 Remaining Action	00 Duratio	0:01:31 in (hh.mm	
Test Designer	¢	Basic Executi	on Managemei	t (Mobile)	Requests	Reports Vali	dations Srr	art Tags						N	ot Runn	
Mobile Designer	<	🛢 Dataset Mar	tagar 🛛 🕇 Ada	Execution												
S Al Scripting	~	Define input data Map desired input	a for the Blueprin	t using the D oding variab	ataset Manager	Click Inputs Found	l link to show all k the Execution?	ound pages with Play button to c	inputs to be defi	ned.						
Oversity Discoversity	-	map depited ripe	an mar correspo	rung runus			e -	r nay bollon to c	ona noo bhooprinn	Remaining	Duration			Incons	sistert	
Create blueprint	_	UniverseDemo		Cor	ICOL Panel	Lagaset	Devices	Pages Hound	Actions Found	Actions	(htcmm:ss)	suns	Falled Actions	Acti	ions	
Compare Blueprints		Univer	seDemo	×	Inputs Found	1 Datasets	0	4	4.38 🗸 📥	0	00:01:31	Not Running				
Blueprint Template Editor							Pages with Inputs	All Pages	*							
Generate Scrinte	- 1	(1 Custom Action)	ons) international S	bace Station										Hit	de image	
Contrate Conpes		Actions 👻					Add Action to 5	un 1	0.37 🏚 🛍	A			•	11	^	
Coverage Map		Type	Name			Ver	able Crea	ed .								
	_	Navigation	1) Home()			(2) Do	Nothing 👻									
Service Suite	<	De Nothing 🐱	None			~ III	90	92 - C C.	•				A			
	_	ProcessSQLFile	Do Nathing					*	= <i>`</i>				A HON	16		
Scenario Editor		Internal State	Do Nathing					¥					-			
Play		Custom	JS Code													
L Analyze	- 1	4					Add Action to	ad 🗸		ntern	ationa	al Spa	ace			
	_										Static	n				
DPL Options	•							1.0				un Canth				
d Traffic																
	-															

14. Result : Rocket Smart Tag found.

														-
= appvance.ai													4	appvan
b Dashboard	Â	File * Option	s * 🧧						0 Devices	3 Pages Found	4 Actions Found	0 Remaining Actions	00 Duration	:01:31 (hh:mm
N Test Designer	•	Basic Executio	en Management (M	obile) Reque	its Reports Vi	lidations Sr	hart Tags						No	t Runr
Mobile Designer	<	B Dataset Mana	ager 🛛 🕇 Add Exec	ution 🔲 Stop										
Al Scription		Define input data Map desired input	for the Blueprint usin	g the Dataset Ma	nager. Click Inputs Fou	nd link to show all	found pages with	inputs to be defi	ned.					
Create Disperint	- 1	Ne		Control Broad	Dataset	Devices	Prove Found	Arriver Erund	Renaining	Duration	\$100.00	Exited Artistee	income	istent
create tracprine	_						regestione		Actions	(htcnin:ss)			Activ	65
Compare Blueprints		Universe	eDemo	Inputs F	ound 👼 1 Datasets	0	4	4.38 🗸 👗	0	00:01:31	Not Running			
Blueprint Template Editor						Pages with Input	s: All Pages	~						
Generate Scripts		I Custom Actions	s) international Space Stati	on										
		co [1 Custom Action	na] Aatroverse					_					Hđ	i krage
Coverage Map		Actions 🗸				Add Action to	sur AS	A						
Consiste Cuite		Type Nome Variable Created												
Service Surve		Co Nation	None			a many •		1						
ervice Suite		ProcessSQLFile	Do Nothing						~ /		70			
Scenario Editor	_	Internal State	Do Nothing				· /							
Scenario Editor		Oustam	JS Code				- 1 A			- 7		_		
Scenario Editor Play						Page Artico In	End -							
Scenario Editor Play Analyze	-													_
Scenario Editor Play Analyze DPI Detions		4							Ľ		•			

15. Result : Solar System Smart Tag found.

aal Appvance IQ : Blueprint	×	Index of REPOSITOR	RY/DemoRep: ×	+										\sim	-	0
← → C	st:8080/UI/E	llueprint/Create.html	14									o- 0	QB	*	* 0	1 4
≡ appvance.ai															4 a;	ppvano
2 Dashboard	^	File - Option	ns 💌 📒							0 Devices	3 Pages Found	4 Actions Found	0 Remaining A	ctions	00:01 Duration (h	1:31
M Test Designer	<	Basic Executi	ion Management	(Mobile)	Requests	Reports V	alidations Sm	art Tags							Not I	Runnin
Mobile Designer	¢	B Dataset Ma	nager 🔰 🕇 Add E	Execution												
& Al Scripting	×	Define input data Map desired input	a for the Blueprint i uts with correspon	using the D ding veriab	lataset Manage les defined in D	r. Click Inputs Fou atasets and then c	ind link to show all lick the Execution's	ound pages with Play button to o	h inputs to be defi continue Blueprint	ned.						
Create Blueprint	reate Blueprint		larne	Control Panel		Dataset	Devices	Pages Found	Actions Found	Remaining Actions	Duration (htcmm:ss)	Status	Failed Acti	ions	Actions	ant s
Compare Blueprints		Univer	rseDemo	E	Inputs Found	E 1 Cetasets	0	4	4.35 🗸 🛦	0	00:01:31	Not Running				
Blueprint Template Editor							Pages with Inputs	All Pages	×							
Generate Scripts	- 1	of [1 Custom Action	ins) International Space	Station												
	- 1	(i) [1 Custon Actor (ii) [2 Custon Actor	ins Astroverse												LEAN IN	
Coverage Map	- 1	Artices					••••••••••••••••••••••••••••••••••••		_	_	_	Anov	ioro to I	bie		
I Service Suite		Type	Name			1	Arishie Const	det.								
		Navigation	1) Rocket()				Da Nothing 👻			int Perso						
✗ Scenario Editor		Navigation	2) SolarSystem()			2	Da Nothing 🐱									
		Navigation	3) Rocket()				Da Nothing 👻 👘									
 Play 		Navigation	4) SolarSystem()				Da Nothing 🐱 👘		_	_						
		Do Nathing 🐱	None			~	Nane	1 V		\sim						
<u>e</u> Analyze		ProcessBQLFile	Do Nothing					*								
	_	internal State	Do Nothing					¥								
BDPL Options	· · ·	Oustom	JS Code													
al Traffic							Add Action to	nd 🖕				- 4				

16. Save the results "Actions Found" and in **File** save the Blueprint.