

Test Site User Guide

Test Site
QA: https://test-site.foodtruck-qa.com
UAT: https://test-site.foodtruck-uat.com
Test Report
QA: https://test-report.foodtruck-qa.com
UAT: https://test-report.foodtruck-uat.com

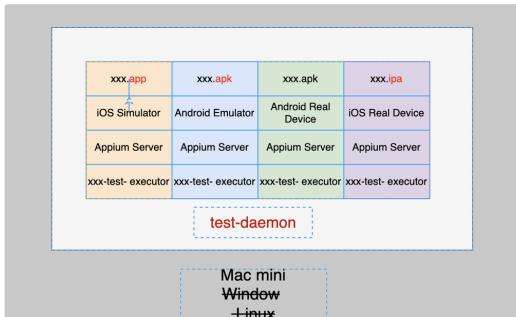
Test Automation Platform

Step	location
Prepare test-executor	<ul style="list-style-type: none"> https://github.com/food-truck/consumer-app-test-executor Connect your Github account https://github.com/food-truck/courier-app-test-executor Connect your Github account https://github.com/food-truck/consumer-web-test-executor Connect your Github account
Prepare application package	<ul style="list-style-type: none"> app regression test: http://192.168.11.38 mobile web test: Chrome Or Safari
Deploy test-daemon	<ul style="list-style-type: none"> mac mini m2: 192.168.33.2 (qa2023) mac mini i7: 192.168.33.5 (qa2018)
Register project	QA: https://test-site.foodtruck-qa.com UAT: https://test-site.foodtruck-uat.com
Make plan	QA: https://test-site.foodtruck-qa.com UAT: https://test-site.foodtruck-uat.com
Monitor device	QA: https://test-site.foodtruck-qa.com UAT: https://test-site.foodtruck-uat.com
View Report	QA: https://test-report.foodtruck-qa.com UAT: https://test-report.foodtruck-uat.com

- **Test Automation Platform**
 - How to deploy test daemon on mac mini
 - Environmental Requirements
 - Build test daemon on development environment
 - Deploy test daemon on mac mini
 - Real device parallel testing
 - Virtual device parallel testing
 - How to Register Project on Test site?
 - Step 1. Creating executor repo on github
 - Step 2: Creating a personal access token from gitgub
 - Step 3: Register project on test site?
 - How to make plan?
 - How to generate UI Difference Report?
 - How to Generate Allure Report?

How to deploy test daemon on mac mini

How to deploy test daemon on mac mini



Environmental Requirements

Tool	how to install ?	Env variable configuration	remark
homebrew	<p>URL: Installation</p> <pre> 1 #check install 2 brew -v </pre>		required install
xcode	<pre> 1 xcode-select --install 2 xcrun --version </pre>		Required install to use iPhone devices
libimobiledevice	<pre> 1 # via homebrew 2 brew install libimobiledevice 3 4 # or via port [port install:https://www.macports.org/install.php </pre>		Required install to use iPhone real devices

	<pre> 5 sudo port install libimobiledevice 6 7 # check install 8 ideviceinfo -v </pre>	
android studio	download from Download Android Studio & App Tools - Android Developers	<pre> 1 # modify ~/.bash_profile, and add the follow content 2 echo 'export ANDROID_HOME=~/Library/Android/sdk' >> ~/.bash_profile 3 echo 'export PATH=\$PATH:\$ANDROID_HOME/platform- tools:\$ANDROID_HOME/emulator' >> ~/.bash_profile 4 source ~/.bash_profile 5 #check install: adb 6 adb version 7 #check install: emulator 8 emulator -version </pre>
android virtual device: google	Create and manage virtual devices Android Studio Android Developers	<p>Required install to use android devices</p> <p>Modify the key-value pair hw.device.manufacturer & hw.device.model in the <code>~/.android/avd/<avd_id>.avd/config.ini</code> file</p> <pre> 1 # Modify the key-value pair hw.device.manufacturer & hw.device.model in the ~/.android/avd/<avd_id>.avd/config.ini file 2 # demo: 3 # avd_id=Samsung_Galaxy_S21 4 # hw.device.manufacturer=Samsung 5 # hw.device.model=Galaxy S21 6 cd \$(cat ~/.android/avd/<avd_id>.ini grep path= awk -F'=' '{print \$2}') 7 sed -i '' '/hw.device.manufacturer/d' config.ini 8 echo "hw.device.manufacturer= <manufacturer>" >> config.ini 9 sed -i '' '/hw.device.model/d' config.ini 10 echo "hw.device.model=<model>" >> config.ini 11 12 # how to get avd_id 13 emulator -list-avds </pre>
android virtual device : samsung	how to create android virtual device for samsung?	<p>Required install to use samsung virtual devices</p> <pre> 1 # Modify the key-value pair hw.device.manufacturer & hw.device.model in the ~/.android/avd/<avd_id>.avd/config.ini file 2 # demo: 3 # avd_id=Samsung_Galaxy_S21 4 # hw.device.manufacturer=Samsung 5 # hw.device.model=Galaxy S21 6 cd \$(cat ~/.android/avd/<avd_id>.ini grep path= awk -F'=' '{print \$2}') 7 sed -i '' '/hw.device.manufacturer/d' config.ini 8 echo "hw.device.manufacturer= <manufacturer>" >> config.ini 9 sed -i '' '/hw.device.model/d' config.ini 10 echo "hw.device.model=<model>" >> config.ini 11 12 # how to get avd_id 13 emulator -list-avds </pre>
appium	<ul style="list-style-type: none"> • appium1 <pre> 1 brew install node 2 #check install 3 node --version 4 5 npm install -g appium 6 #check install 7 appium --version </pre> <ul style="list-style-type: none"> • appium2 <pre> 1 brew install node 2 node --version 3 npm install -g appium@next </pre>	<p>Required install</p>

	<pre> 4 appium --version 5 // Installing drivers 6 appium driver install uiautomator2 7 appium driver install xcuitest 8 // npm install --global appium -- drivers=xcuitest,uiautomator2 </pre>		
iOS screen recording plugin	<pre> 1 # install or sudo port install ffmpeg 2 brew install ffmpeg 3 # check install 4 ffmpeg -version </pre>		Required install to use iPhone devices
JDK17	<p>Java Downloads Oracle 中国</p> <pre> 1 #check install 2 java -version </pre>	<pre> 1 export JAVA_HOME=/Library/Java/JavaVirtualMachines/jdk-17.jdk/Contents/Home 2 export CLASSPATH=\$JAVA_HOME/lib/tools.jar:\$JAVA_HOME/lib/dt.jar 3 export PATH=\$PATH:\$JAVA_HOME/bin </pre>	Required install to run test-daemon & test-executor

Build test daemon on development environment

- download test-daemon from Github Repo: <https://github.com/food-truck/foodtruck-test-automation> Connect your Github account
- on rootProject , run `./gradlew installDist -p test-daemon -Penv=(env)`

```

1 #dev
2 ./gradlew installDist -p test-daemon -Penv=dev
3 #uat
4 ./gradlew installDist -p test-daemon -Penv=uat

```

- Navigate to a directory: {rootProject}/build/test-daemon/install, compressed into file `test-daemon.zip`

Deploy test daemon on mac mini

- decompress `test-daemon.zip`
- run command: `nohup sh {} max_number_of_devices={} app_name={} 2>&1 &`

```

1 #dev
2 nohup sh ~/Documents/dev/test-daemon/bin/test-daemonmax_number_of_devices=3 app_name=WONDER > ~/Documents/dev/log/2024.04.01.log 2>&1 &
3
4 tail -f ~/Documents/dev/log/2024.04.01.log

```

- kill test-daemon

```

1 #get pid
2 jps
3 #kill process
4 kill {pid}

```

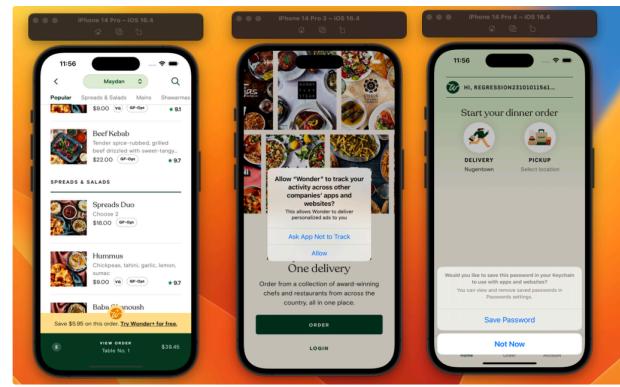
Real device parallel testing

- iPhone real device
- Android real device



Virtual device parallel testing

- iPhone simulator



How to Register Project on Test site?

Step 1. Creating executor repo on github

- Naming format

	format	example	Remark
Github Organization	food-truck	food-truck	-
Github Repo Name	xxx-test-executor	consumer-app-test-executor . consumer-web-test-executor	-
Executor Project Name	xxx-test-service	consumer-app-test-service	build package to run executor by gradle or maven
Case Project Name	xxx-test-interface	consumer-app-test-interface	parse test cases
Case Package Name	xxx.xxx.xxx	app.consumerappservice.wonder	The package directory where the test case is located
branch	number.number.number	1.0.0 2.0.0	branch:project version=1:1

Step 2: Creating a personal access token from gitgub

URL: <https://github.com/settings/tokens> Connect your Github account

- In the left sidebar, under **Personal access tokens**, click **Tokens (classic)**.
- Select **Generate new token**, then click **Generate new token (classic)**.
- In the "Note" field, give your token a descriptive name.
- To give your token an expiration, select **Expiration**, then choose a default option or click **Custom** to enter a date.

- Select the scopes you'd like to grant this token. To use your token to access repositories from the command line, select **repo**. A token with no assigned scopes can only access public information. For more information, see "[Scopes for OAuth apps](#)".

Select scopes	
Scopes define the access for personal tokens. Read more about OAuth scopes .	
<input checked="" type="checkbox"/> repo	Full control of private repositories
<input checked="" type="checkbox"/> repo:status	Access commit status
<input checked="" type="checkbox"/> repo_deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories
<input checked="" type="checkbox"/> repo:invite	Access repository invitations
<input checked="" type="checkbox"/> security_events	Read and write security events

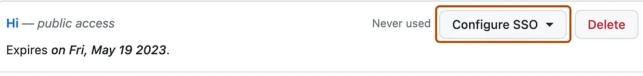
- Click **Generate token**.
- Optional, to copy the new token to your clipboard, click

8. Next to the token you'd like to authorize, click **Configure SSO**. If you don't see **Configure SSO**, ensure that you have authenticated at least once through your SAML IdP to access resources on [GitHub - Build and ship software on a single, collaborative platform](#). For more information, see "[About authentication with SAML single sign-on](#)."

Make sure to copy your personal access token now. You won't be able to see it again!



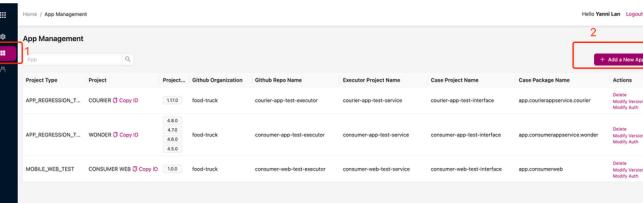
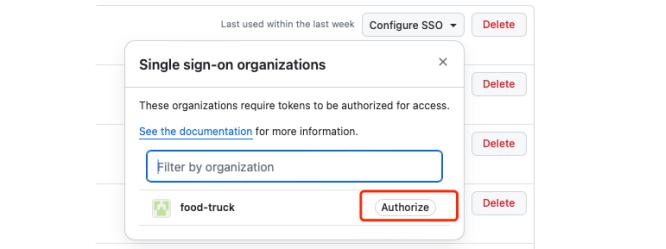
9. In the dropdown menu, to the right of the organization you'd like to authorize the token for, click **Authorize**.



Step 3: Register project on test site?

URL: [Wonder QA Automation Admin](#)

1. Navigate to "APP Management" page
2. click "Add a New APP"
3. on "New Configuration" : Binding project and executor.



How to make plan?

URL: [Wonder QA Automation Admin](#)

1. Navigate to "Test Plan Management" page
2. click "Make a Test Plan"

- ### 3. on “Make a test Plan” :

- input plan name;

- choose app & app version

Make a Test Plan

Plan name:
sprint1

Portfolio:
Wunder Auto Test Lab

Area:
Wunder

App Version:
4.8.0 4.7.0 4.6.0 4.5.0

3. on “Make a test Plan” modal :

- select cases

1 Choose an App
sprint1 24/10/2024
Wonder 4.8.3

2 Choose Test Cases
0 Test Cases
0.470 0.460 0.450

3 Choose Test Devices
0 iOS Phones
0 Android Phones

Cancel | Reschedule | Start Now

Make a Test Plan

Direction: 

Unselected Cases 48 items

Search here 

Case No.	Case Name	Feature	Story	Epic	Type	Priority
postrelease-011	checkOrderConfirmEmail		Place order, che...	PostReleaseTestCase	Functional	Critical
postrelease-012	checkOrderCancelEmail		Cancel order, che...	PostReleaseTestCase	Functional	Critical
postrelease-013	checkDeliveryWhenAllowLocation		view some pages ...	PostReleaseTestCase	Functional	Critical

Selected Cases 1 item

Search here 

Case No.	Case Name	Feature	Story	Epic	Type	Priority
postrelease-014	checkDeliveryWhenDenyLocation		view some pages ...	PostReleaseTestCase	Functional	Critical

Choose an App
sprint1 24/10/2024
Wonder 4.8.3

Choose Test Cases
0 Test Cases
0.470 0.460 0.450

Choose Test Devices
0 iOS Phones
0 Android Phones

Cancel | Reschedule | Start Now

3. on “Make a test Plan” modal :

- select devices

Make a Test Plan					
Unselected Devices 10 items			Selected Devices 2 items		
Device	Device Type	OS Version	Display	Device Sets	Direction
Samsung Galaxy Note20 Ultra	Virtual	Android 12	1440 x 3080		↔
Samsung Galaxy S23 FE	Virtual	Android 13	1080 x 2340		↔
Samsung Galaxy S23 Ultra	Virtual	Android 11	1440 x 3088		↔
<hr/>					
Device	Device Type	OS Version	Display	Device Sets	Remove
Apple iPhone 12	Virtual	iOS 15.0	1170 x 2032		Remove
Samsung Galaxy A05	Virtual	Android 12	720 x 1600		Remove

3. on test plan page:

- test plan : device task (1:n)

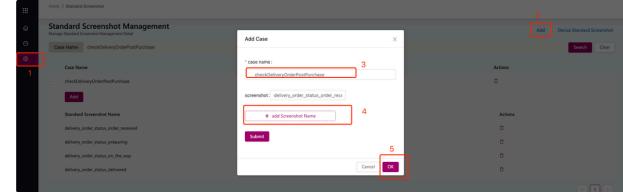
Test Plan							Actions		Report		Message	
Plan	Platform	App	Cases	Devices	Duration	Created	Action	Link	Cancel	Summary	Details	
- sprint 1	Wonder Auto Test Lab	Wonder App	1	02	00 mins	2024-01-10 10:24:22	Cancel					
Device	Device Type	Old App Version	Cases	Duration	Actions	Reports						
iPhone 12 iOS 16.0	Virtual	Virtual	Pending	00 mins	Cancel	Functional	Detail					
Case	expand	Case No.	Type	Priority	Device ID	Status	Details					
checkDeliveryWhenDempLocation	postrelease-C15	Functional	Medium	Low		Pending	Detail					
Galaxy A5 Android 10	Virtual	Virtual	Pending	00 mins	Cancel	Functional	Detail					
Case	expand	Case No.	Type	Priority	Device ID	Status	Details					
checkDeliveryWhenDempLocation	postrelease-C16	Functional	Medium	Low		Pending	Detail					

How to generate UI Difference Report?

URL: <https://test-report.foodtruck-qa.com> or <https://test-report.foodtruck-uat.com>

Step

1. navigate to "Standard Screenshot Management" page
2. click "add" to new standard screenshot



3. navigate to "UI Diff Case" Page

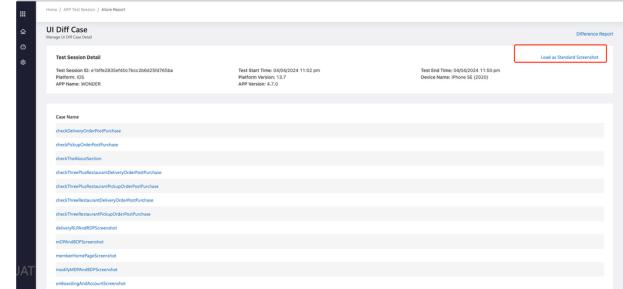
entry1: navigate to "APP Test Report Dashboard" page after test completed (test report site)

entry2: navigate to "Test Plan" page after test completed (test site)

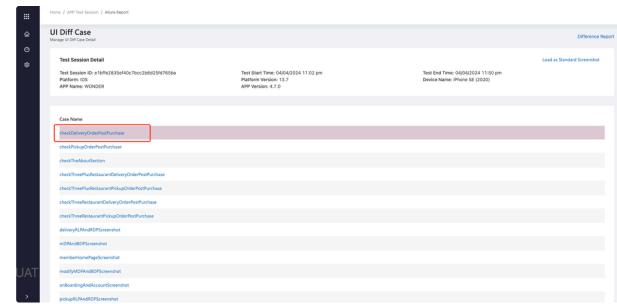
demo:

- device_task_id = test_session_id = e1bffe2835ef40c7bcc2b6d25fd765ba
- plan_id = 3c9064cb04a54d19842f2c652a1989dc

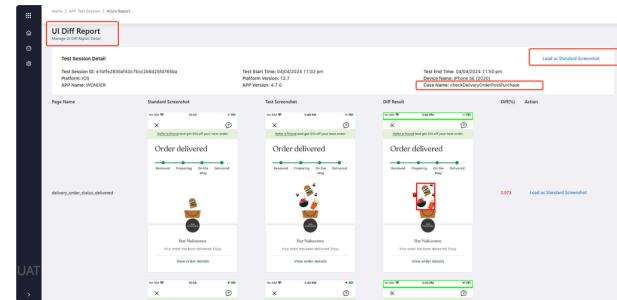
4. click "load as standard screenshot" to Updated standard screenshots for all test cases.



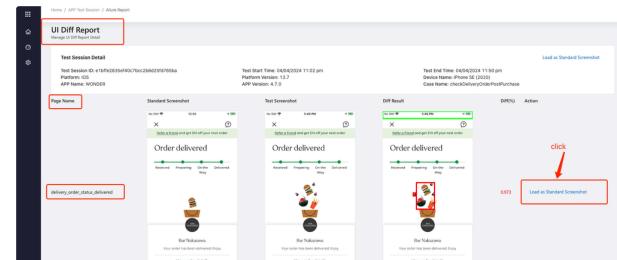
5. navigate to "UI Diff Report" Page



6. click "load as standard screenshot" to update all standard screenshots for a test case



7. click "load as standard screenshot" to update a standard screenshot of a test case

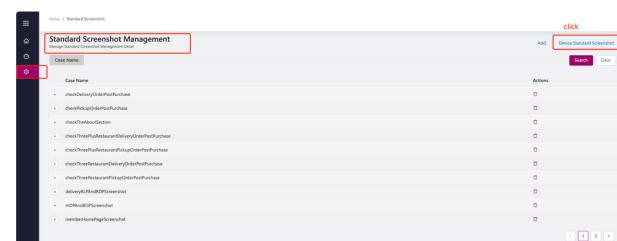


8. navigate to "Device Screenshot Management" page, click "Device Standard Screenshot"

9. on "page, check device standard screenshot , demo

a. test_session_id = e1bffe2835ef40c7bcc2bd25fd765ba

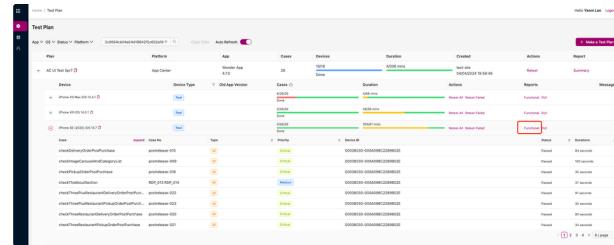
b. devcie= iPhone SE (2020)



How to Generate Allure Report?

step

1. entry: navigate to "test plan" page, on device task level: click "Functional".



2. on "Allure Report" Page, click "Generate Allure Report".