User Guides

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

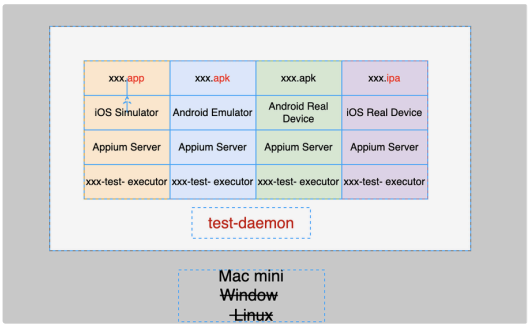
Test Report
QA: https://test-report.foodtruck-qa.com
UAT: https://test-report.foodtruck-uat.com

Test Automation Platform

Step	location
Prepare test-executor	• 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	• app regression test: http://192.168.11.38 • mobile web test: Chrome Or Safari
Deploy test-daemon	• 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

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	URL: Installation `#check install` `brew -v`		required **install**
xcode	`xcode-select --install` `xcrun --version`		Required install to use iPhone devices

libimobiledevice	 ``` # via homebrew brew install libimobiledevice # or via port [port install:https://www.macports.or sudo port install libimobiledevice # check install ideviceinfo -v ``` 		Required install to use iPhone real devices
android studio	download from Download Android Studio & App Tools - Android Developers	 ``` # modify ~/.bash_profile, and add the fol echo 'export ANDROID_HOME=~/Library/Androi echo 'export PATH=$PATH:$ANDROID_HOME/plat source ~/.bash_profile #check install: adb adb version #check install: emulator emulator -version ``` 	Required install to use android devices
android virtual device: google	Create and manage virtual devices | Android Studio | Android Developers	Modify the key-value pair hw.device.manufacturer & hw.device.model in the `~/.android/avd/<avd_id>.avd/config.ini` file ``` # Modify the key-value pair hw.device.mar # demo: # avd_id=Samsung_Galaxy_S21 # hw.device.manufacturer=Samsung # hw.device.model=Galaxy S21 cd $(cat ~/.android/avd/<avd_id>.ini | gr sed -i '' '/hw.device.manufacturer/d' con echo "hw.device.manufacturer=<manufacture sed -i '' '/hw.device.model/d' config.ini echo "hw.device.model=<model>" >> config. # how to get avd_id emulator -list-avds ``` 	Required install to use android virtual devices
android virtual device : samsung	how to create android virtual device for samsung?	 ``` # Modify the key-value pair hw.device.mar # demo: # avd_id=Samsung_Galaxy_S21 # hw.device.manufacturer=Samsung # hw.device.model=Galaxy S21 cd $(cat ~/.android/avd/<avd_id>.ini | gr sed -i '' '/hw.device.manufacturer/d' con echo "hw.device.manufacturer=<manufacture sed -i '' '/hw.device.model/d' config.ini echo "hw.device.model=<model>" >> config. # how to get avd_id emulator -list-avds ``` 	Required install to use samsung virtual devices
appium	• appium1 ``` brew install node #check install node --version npm install -g appium #check install appium --version ``` • appium2 ``` brew install node node --version npm install -g appium@next appium --version // Installing drivers appium driver install uiautomator2 appium driver install xcuitest // npm install --global appium --drivers=xcuitest,u ``` 		Required install
iOS screen recording plugin	 ``` # install or sudo port install ffmpeg brew install ffmpeg # check install ffmpeg -version ``` 		Required install to use iPhone devices
JDK17	Java Downloads | Oracle 中国 ``` #check install java -version ``` 	 ``` export JAVA_HOME=/Library/Java/JavaVirtual export CLASSPATH=$JAVA_HOME/lib/tools.jar: export PATH=$PATH:$JAVA_HOME/bin ``` 	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}`

```
#dev
./gradlew installDist -p test-daemon -Penv=dev
#uat
./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 &**

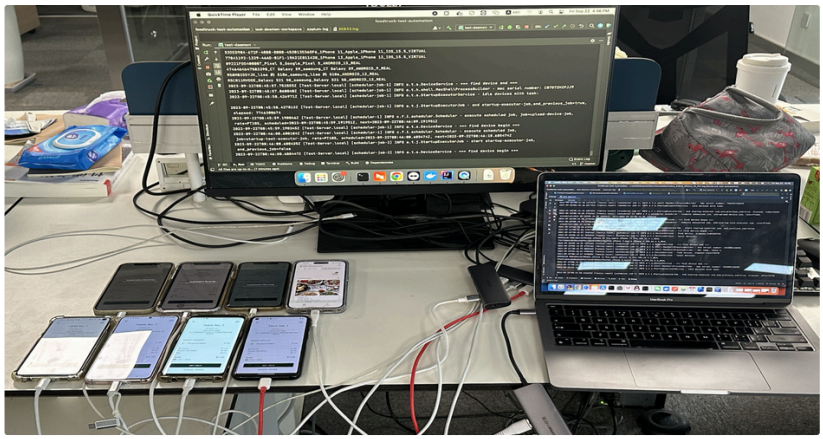
```
#dev
nohup sh ~/Documents/dev/test-daemon/bin/test-daemonmax_number_of_devices=3 app_name=WONDER > ~/Documents/dev/log/2024.04.01.log 2>&1 &

tail -f ~/Documents/dev/log/2024.04.01.log
```

- kill test-daemon

```
#get pid
jps
#kill process
kill {pid}
```

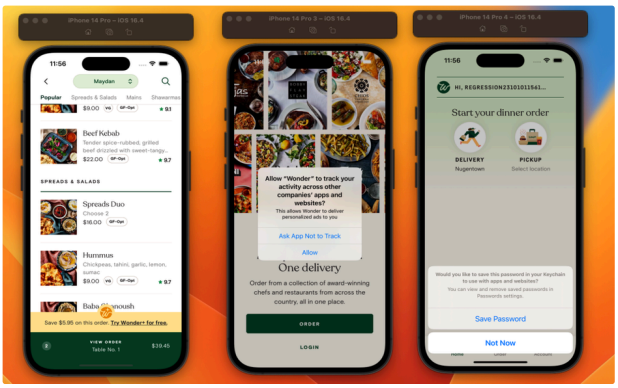
Real device parallel testing

- iPhone real device
- Android real device



Virtual device parallel testing

- iPhone simulator

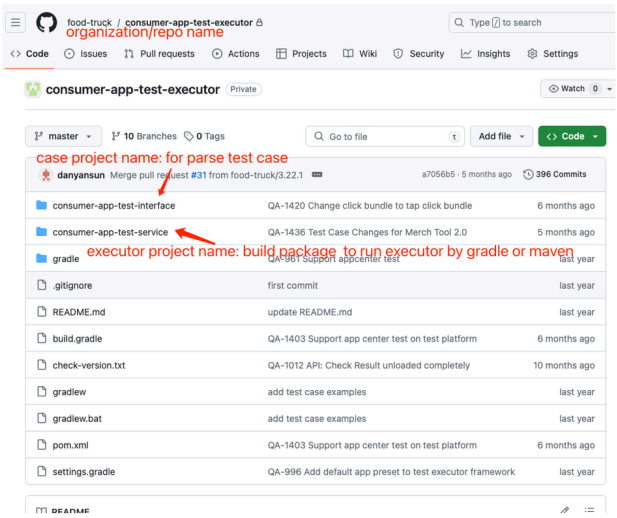


How to Register Project on Test site?

Step 1. Creating executor repo on github

- Naming format

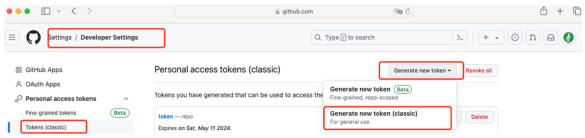
	format	example	Remark
Github Organization	food-truck	`food-truck`	-
Github Repo Name	xxx-test-executor	`consumer-app-test-executor` 、 `consumer-web-test-executor`	-
Executor Project Name	xxx-test-service	`consumer-app-test-service`	build package to run executor by gradle or maven
Case Project Name	xxx-test-interface	`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



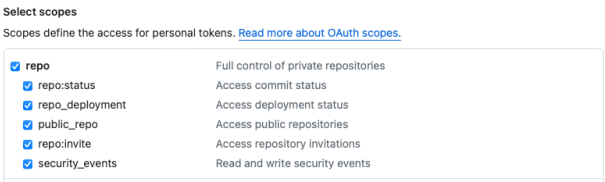
Step 2: Creating a personal access token from gitgub

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

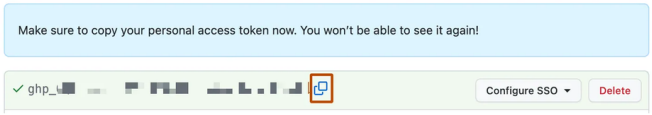
1. In the left sidebar, under **Personal access tokens**, click **Tokens (classic)**.
2. Select **Generate new token**, then click **Generate new token (classic)**.
3. In the "Note" field, give your token a descriptive name.
4. To give your token an expiration, select **Expiration**, then choose a default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



5. 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."

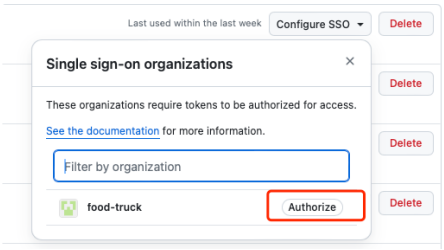


6. Click **Generate token**.
7. Optionally, to copy the new token to your clipboard, click



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: Let's build from here . For more information, see "About authentication with SAML single sign-on."

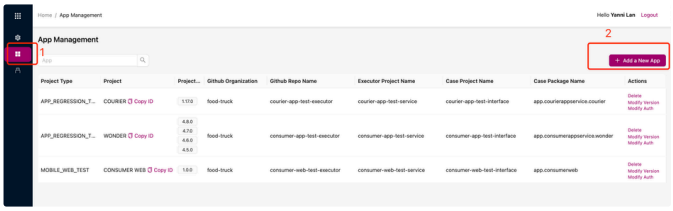
9. In the dropdown menu, to the right of the organization you'd like to authorize the token for, click **Authorize**.

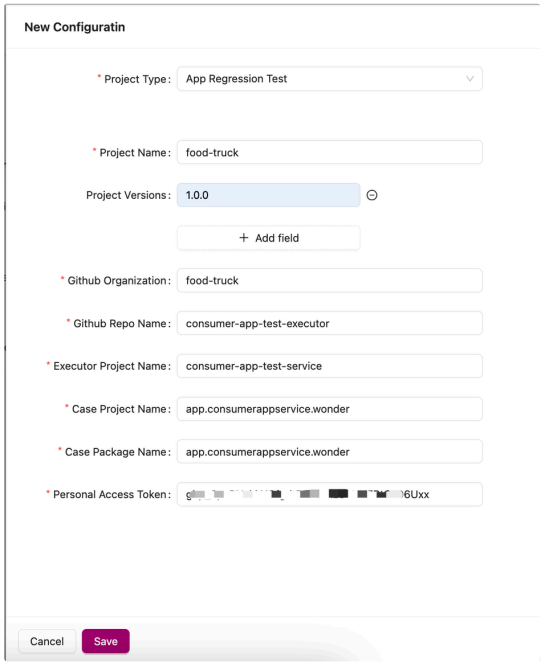


Step 3: Register project on test site?

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.

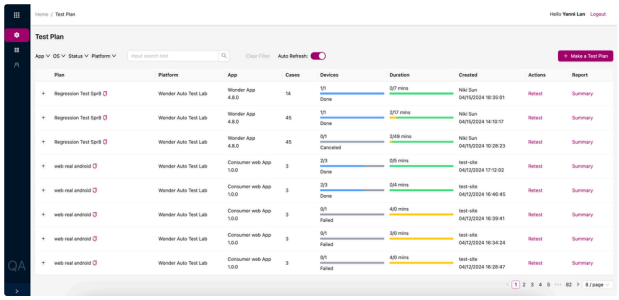




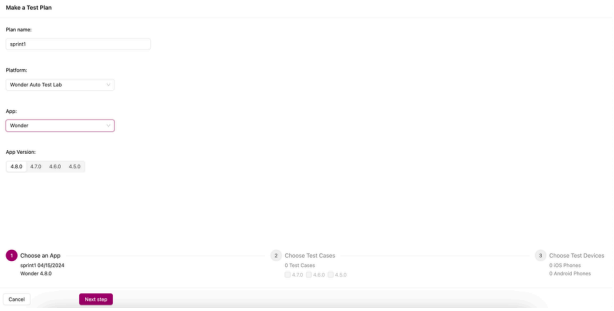
How to make plan?

URL: Wonder QA Automation Admin

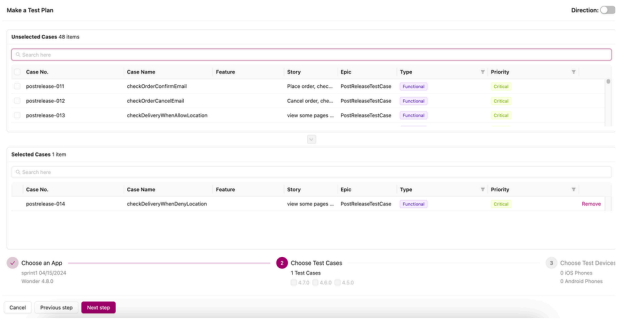
1. Navigate to "Test Plan Management" page
2. click "Make a Test Plan"



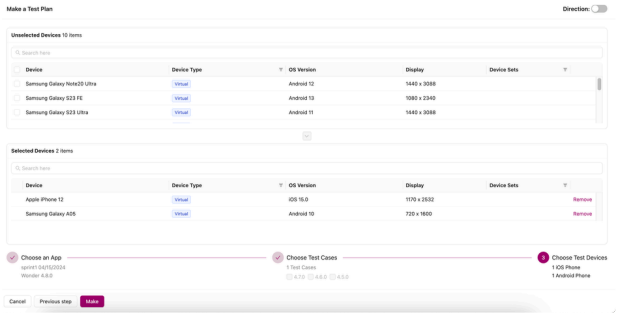
3. on "Make a test Plan" :
   - input plan name;
   - choose app & app version



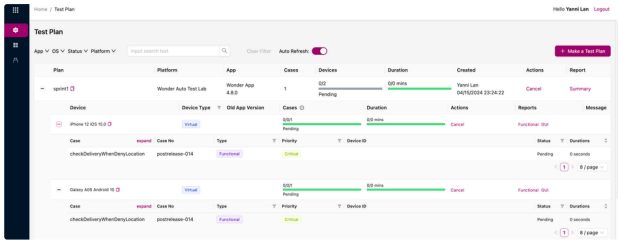
3. on “Make a test Plan” modal :
   - select cases



3. on “Make a test Plan” modal :
   - select devices



3. on test plan page :
   - test plan : device task (1:n)

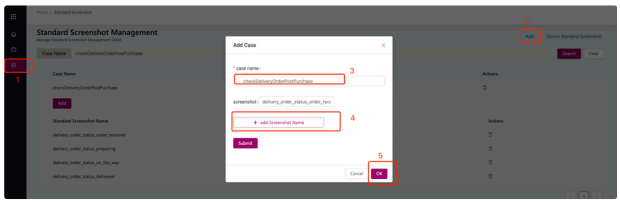


How to generate UI Difference Report?

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



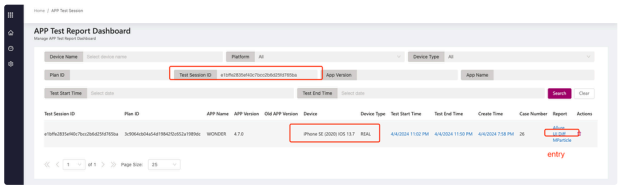
3. navigate to “UI Diff Case“ Page

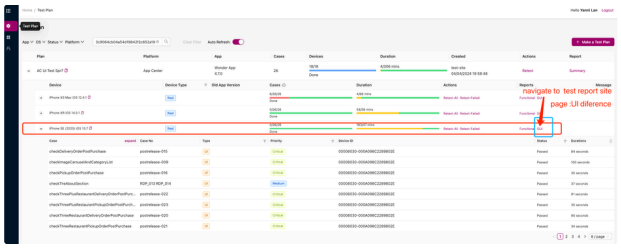
entry1: navigate to “APP Test Report Dashboard“ page after test completed (test report site)

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

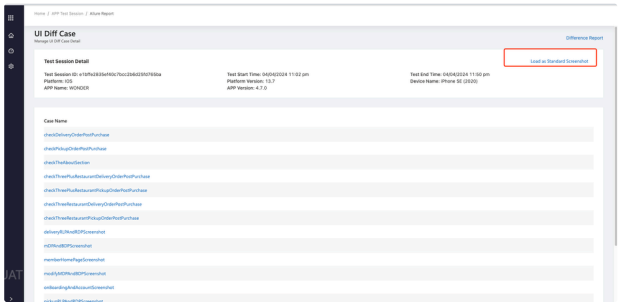
demo:

- device_task_id = test_session_id = `e1bffe2835ef40c7bcc2b6d25fd765ba`
- plan_id = `3c9064cb04a54d19842f2c652a1989dc`

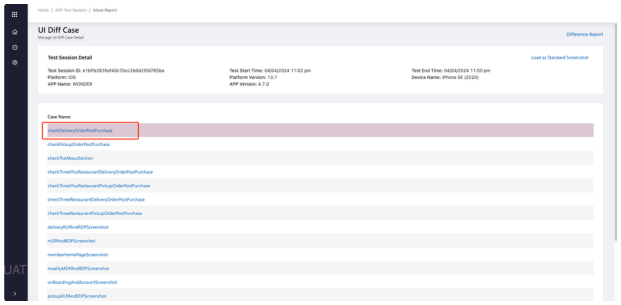




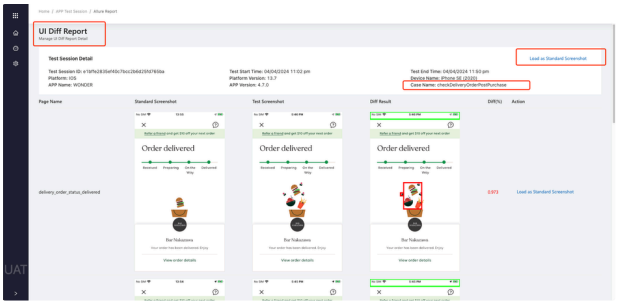
4. click “load as standard screenshot” to Updated standard screenshots for all test cases.



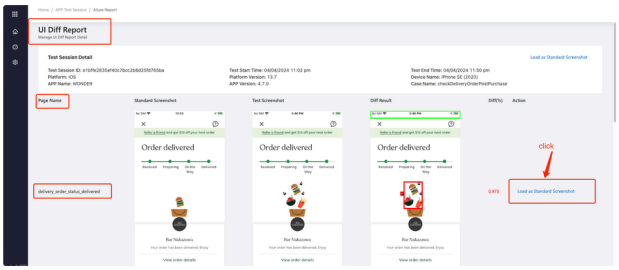
5. navigate to “UI Diff Report“ Page



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

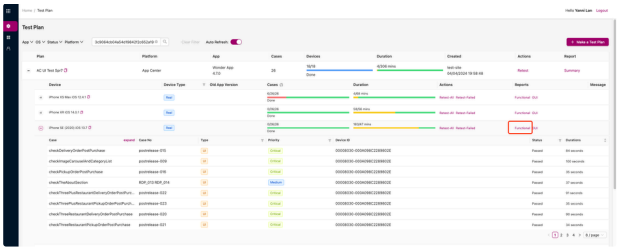


8. navigate to “Device Screenshot Management“ page”, click “Device Standard Screenshot “
9. on ““page, check device standard screenshot , demo
    a. test_session_id = e1bffe2835ef40c7bcc2b6d25fd765ba
    b. devcice= iPhone SE (2020)

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“.