Predefined Steps

Please note that pre-defined steps is intended for people to get started quickly, and is not considered best practice. For best practices we recommend people to architect their test suite based on this article.

Overview

This document gives examples, not the full step definitions. It is a "human readable" description of the step definitions found in the file calabash*steps.rb (calabash-cucumber/features/step*definitions/calabash_steps.rb).

You can easily generalize the examples. For example, Then I touch the "login" button can have any string in quotes, not just "login".

Also note that most steps find views by their accessibility labels or ids. This means that for the tests to work, you must enable accessibility on the simulator or phone you are testing on.

If you have any questions, please use the google group:

Calabash-iOS

There are a number of features that can be tested with predefined steps such as:

- Assertions
- Taking Screenshots
- Touching
- Input
- Waiting
- Buttons
- Gestures (Includes swiping, pinching and scrolling)
- Playback of Touch Events
- Device Orientation

- Location Steps
- Internationalization

Assertions

These steps are used to check for the existence of some UI elements. Click on the platform icon to see the ruby source code for the step definition.

Step	Platform
Then I should see "text or label"	?
Then I should not see "text or label"	?
Then I see the "someview"	?
Then I should see a "login" button	?
Then I should not see a "login" button	?
Then I see the text "some text"	?
Then I don't see the text "text or label"	?
Then I don't see the "someview"	?
Then I should see a "login" button	?
Then I should not see a "login" button	?
Then I should see text starting with "prefix"	?
Then I should see text containing "sub text"	?
Then I should see text ending with "suffix"	?
Then I see 2 input fields	?
Then I should see a "Username" input field	?
Then I should see a "Username" input field	?
Then I should not see a "Username" input field	?
Then I should see the user location	?

Step	Platform
Then I should see a map	?
Then /^I don't see "([^\"]*)"$/	?

Screenshots

Step	Platform
Then take picture	?

Touching (tapping)

Step	Platform
Then I touch "accLabel"	?
Then I touch the "login" button	?
Then I touch button number 1	?
Then I touch the "placeholder" input field*	?
Then I touch list item number 1**	?
Then I toggle the switch	?
Then I toggle the "accLabel" switch	?
Then I toggle the "accLabel" switch	?
Then I touch done	?
Then I touch the user location	?
Then I touch on screen 100 from the left and 250 from the top	?

Note: *This looks for an UITextField with the placeholder property set to the quoted string.

**This can only be used to touch visible cells - it doesn't try to scroll down/up.

*Make sure that use the short name and not the fully quantified name. That means if your id is 'com.foo.R.id.bar*label' you would use 'I press view with id "bar*label"'.

**Will look for a view in the following order:1. Looks for a visible button with matching text. 1. Look for a visible view with a matching content description. 1. Look for a visible view with class name that matches the provided indentifier.

If a view is found we will try to click it.

Input

Note, all these steps work on text fields. Text is entered by setting the text directly on the object.

Calabash uses placeholders for identifying text fields, so ... the "login" input field means the input field with placeholder "login".

Step	**Platform**
Then I enter "text to write" into the "placeholder" input field	❓

I fill in text fields as follows:
 	
`| field      | text    |`
`| Last Name  | Krukow  |`
`| Email      | a@b.c   |`
`| Username   | krukow  |`
`| Password   | 123     |`
`| Confirm    | 123     |`
	❓

Then I enter "text" into input field number 1	❓
Then I clear "placeholder"	❓
Then I clear input field number 1	❓

Waiting

These are usually about waiting to see certain text or ui components. Usually these are identified by their accessibilityLabels, component type (like a navigation bar) or pure text like in a label or a web view.

Step	**Platform**
Then I wait to see "text or label"	?
Then I wait for "text or label" to appear	?
Then I wait until I don't see "text or label"	?
I wait to not see "text or label"	?
Then I wait for the "login" button to appear	?
Then I wait to see a navigation bar titled "title"	?
Then I wait for the "label" input field	?
Then I wait for 2 input fields	?
Then I wait	?

Buttons

Step	**Platform**
Then I go back	?

Gestures

Step	**Platform**
Then I swipe left	?
Then I swipe left on number 2	?
Then I swipe left on number 2 at x 20 and y 10	?
Then I swipe left on "accLabel"	?

Step	Platform
Then I swipe on cell number 2	?
Then I swipe on cell number 2	?
Then I pinch to zoom in	?
Then I pinch to zoom in on "accLabel"	?
Then I scroll down	?
Then I scroll down on "accLabel"/td>	?
Then I scroll down on "accLabel"	?

Playback of touch events

Step	Platform
Then I playback recording "mytouch"	?
Then I playback recording "mytouch on "accLabel"	?
Then I playback recording "mytouch on "accLabel" with offset 10,22	?

Device orientation

Step	Platform
Then I rotate device left	?

Manual Steps

These steps are useful for allowing mixed manual and automated tests and to be documented and kept together. These steps do nothing when the tests are run automatically but are still documented in Cucumber output formatters such as the HTML report. This allows a manual tester to perform the same test case but with extra manual steps such as manual image verification.

To manually request a manual tester to compare the screen with a reference image:

```
Then /^I compare the current screen with the reference image "
([^\"]*) manually"$/ do |name|
```

For example:

```
Then I compare the current screen with the reference image
"features/ref1.png" manually
```

To manually perform a custom step:

```
Then /^I manually (.*)$/ do |action|
```

For example:

```
Then I manually check that the list scrolls smoothly
```

Note that you use the short name and not the fully quantified name. That means if your id is 'com.foo.R.id.bar_label' you would use 'I press view with id "bar_label"'.

Next Steps

After getting familiar with the predefined steps for iOS, you should next take a look through the Best Practice guide