Barista App How-To

Recording a Test Case

1. Install the Barista app on an Android device. If you would like to use activity-flow-based oracles, close the app and run from command line “adb shell pm grant barista.testing.client android.permission.READ_LOGS” while the device is connected to adb (you need to do this task only once after you install the app and you do not need to have your device connected through adb while recording test cases).

2. Set-up the app:
   a. Enable the Barista accessibility service through the phone accessibility settings (Figure 1). If you do not enable the accessibility service, the Barista app will direct you to the phone accessibility settings before starting the test case capture phase.
   b. Enable the Barista input method through the phone input and language settings (Figure 2). If you do not enable the input method service, the Barista app will direct you to the phone input and language settings before starting the test case capture phase.
   c. Select the Barista keyboard as the default input method. If you do not select the Barista keyboard as the default input method, the test case capture phase will not start.

3. Launch the Barista app by tapping the app icon (highlighted icon in Figure 3).

4. Start the test case capture phase:
   a. Enter an email address. Barista will send generated test cases to the address (item a in Figure 4).
   b. Select the package name of the app under test (item b in Figure 4 and Figure 5)
   c. Enter the test ID (item c in Figure 4)
   d. Optionally, set the flag (item d in Figure 4) that tells Barista to replicate the timing of the test case capture phase into the test case (e.g., if a user sets a 30-seconds timer in an alarm clock app and wants to check with an assertion the message displayed when the timer goes off, he or she would set the this flag to true).
   e. Optionally, set the flag (item e in Figure 4) that enables execution of test cases without having the source code of the app under test.
   f. Optionally, set the flag (item f in Figure 4) that enables the addition of activity-flow-based oracles.
g. Tap the start button (item g in Figure 4)

5. The app under test is launched with the Barista menu, which include navigation and test recording controls (shown in Figure 6).
   a. Figure 7 and 8 detail the navigation controls and the test recording controls.
b. Figure 9 and 10 show how to hide and bring back navigation controls.

Figure 9: Hide navigation controls.

Figure 10: Bring back navigation controls.

6. Interact with the app under test to record a test case. Tap the assert button to capture assertions (details in the following section). Tap the finish button to initiate the test case generation phase (on our web server) and receive the captured test case by email.
Adding Assertions

An assertion can be added through the following steps:

1. Tap the assert button (Figure 11). Barista adds a green pane that overlays the application under test for the remaining part of the assertion process.
2. Tap, hold, and drag until the target element of the assertion is highlighted (Figure 12).

![Figure 11: Assert button.](image1.png)  ![Figure 12: Selection of an element targeted by the assertion.](image2.png)

3. A contextual menu is shown, containing different types of assertion, when releasing the finger from the target element of the assertion.
4. Tap a property in the menu to add the specific assertion to the test case (Figure 13).
   a. For “negative” assertions (e.g., not enabled), tap and hold on the menu option to change the assertion to negative. Release and then tap to add the assertion to the test case. (Figure 14).

![Figure 13: Assert button.](image3.png)  ![Figure 14: Selection of an element targeted by the assertion.](image4.png)
Adding Intents

An intent can be sent to the app through the following steps:

1. Tap the intent button (Figure 15). Barista creates an interactive menu (Figure 16) that overlays the application under test.
2. Complete the information associated with the intent using the interactive menu.
3. Tap the send button (item a in Figure 16) to send the intent to the application under test.

![Figure 15: Intent button.](image)

![Figure 16: Interactive menu for sending an intent.](image)