**Super Espresso Test Recorder for Android.**

We have enhanced the features provided by Android Studio’s Espresso Test Recorder. All these additional features are recordable through Super Espresso Test Recorder. Features which are implemented are

• Add N number of methods within a Test Class.

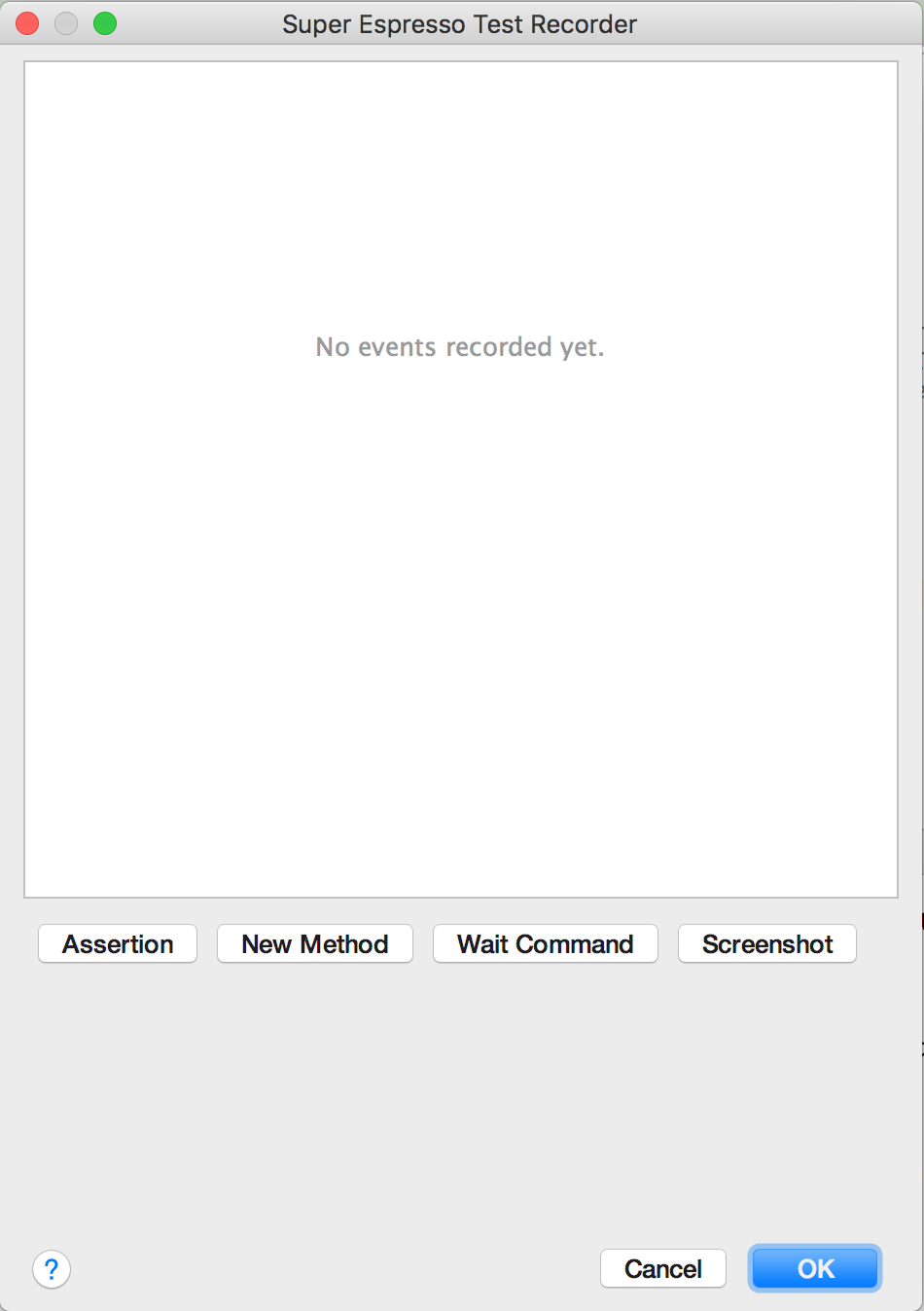
• Add wait statement on any of the View.

• Add a screenshot test while recording the test.

**Getting Started**

This repository contains the Super Espresso Test Recorder plugin in [recorderlibs](https://github.com/Saddala/SuperEspressoTestRecorder/tree/master/recorderlibs), replace the existing test-recorder plugin in Android Studio (AndroidStudio/plugins/test-recorder/lib/test-recorder.jar) with this plugin. Restart the studio. Will share the Super Espresso Test Recoder code base soon in separate GitHub location.

Launch the test recorder.

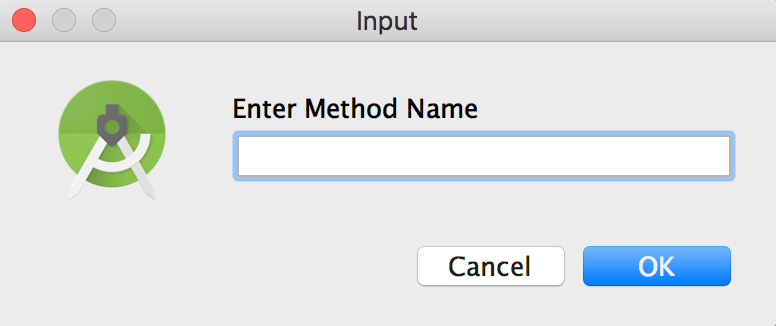


In the above image, you can see multiple new buttons are added. We are going to see about their functionalities.

**New Method:**

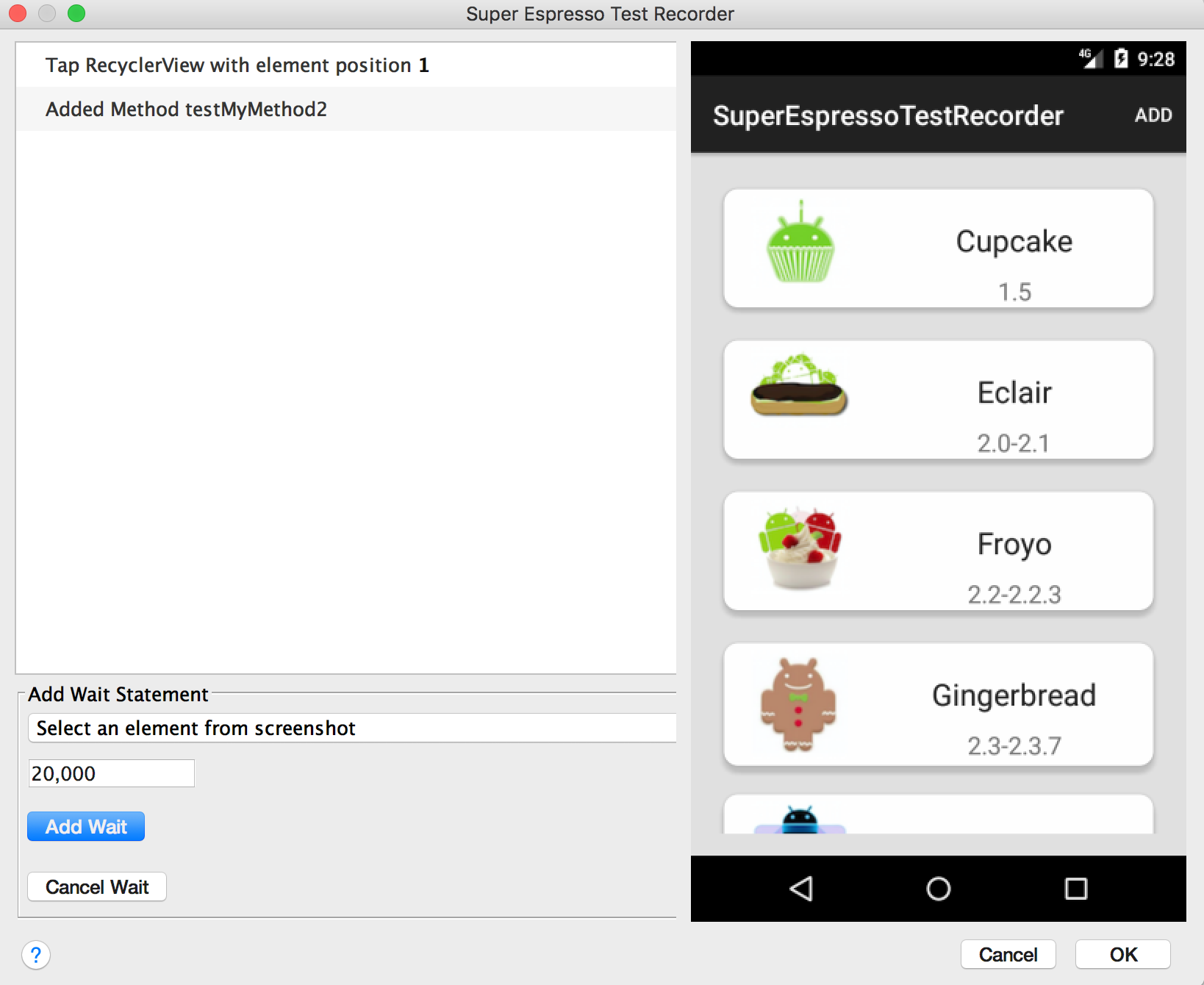
When we click on New Method, a new dialog will be shown. All we need to give a method name and click OK.

Limitation here is first method name in the generated Test File will be generated by the tool. We can provide test methods names from the second case onwards.



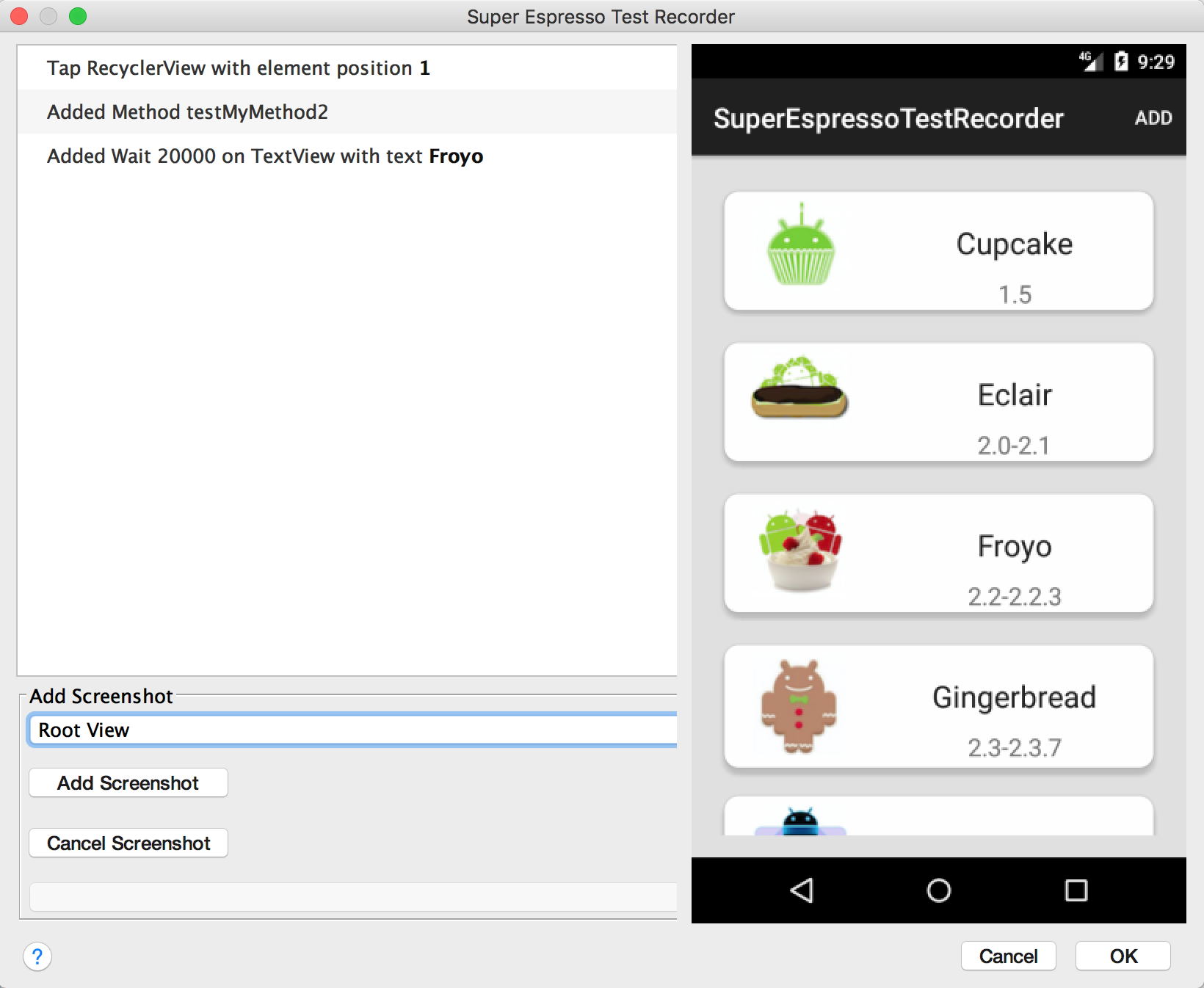
**Wait Command:**

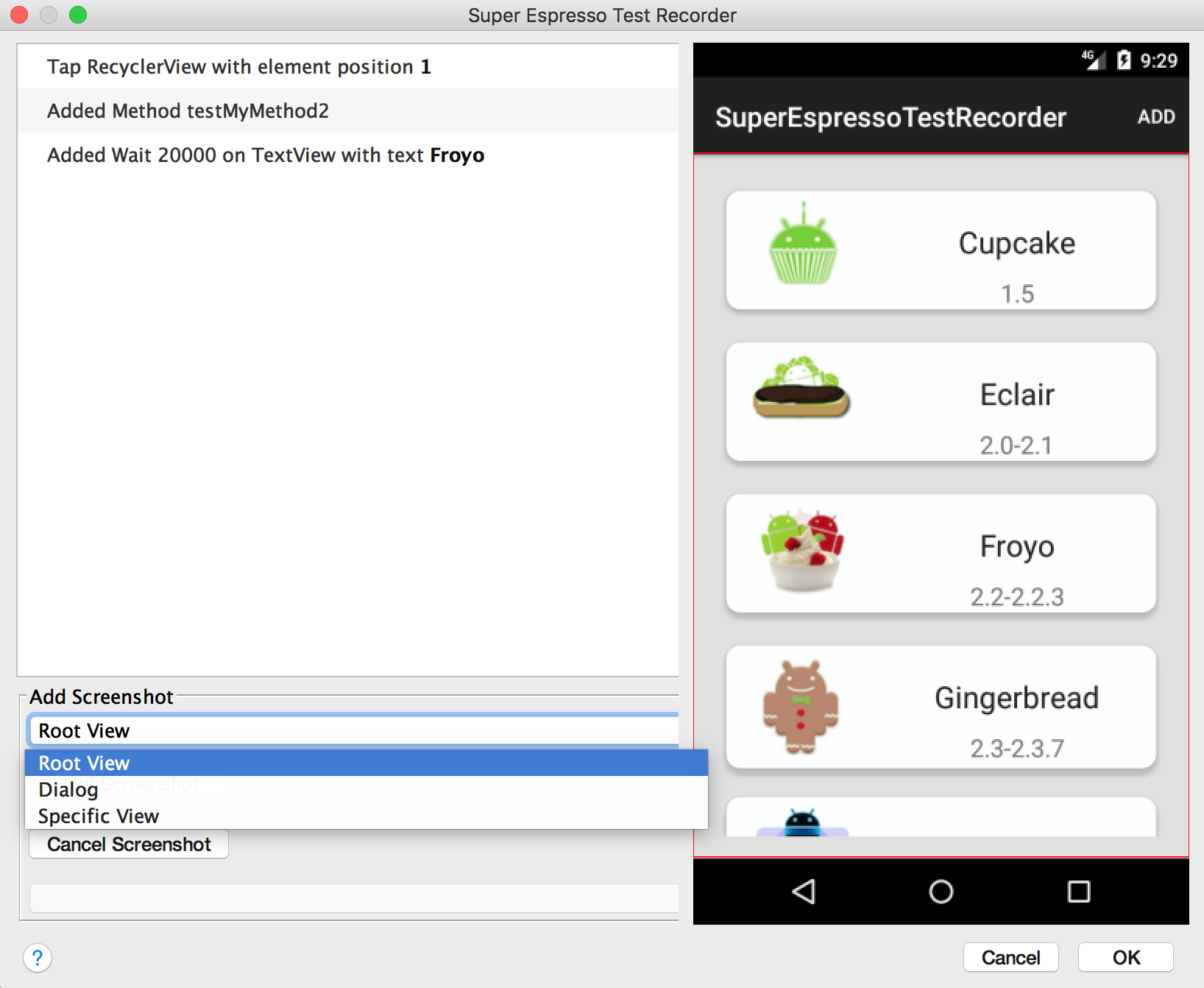
Test recorder shows following view when we click on Add wait. As we can see user can select a particular view from the rendered screenshot or from the drop down in the panel. Maximum timeout can also be provided. Default timeout given is 20000 ms. If no view is selected it will wait on root view. Internally it uses IdlingResource to wait.



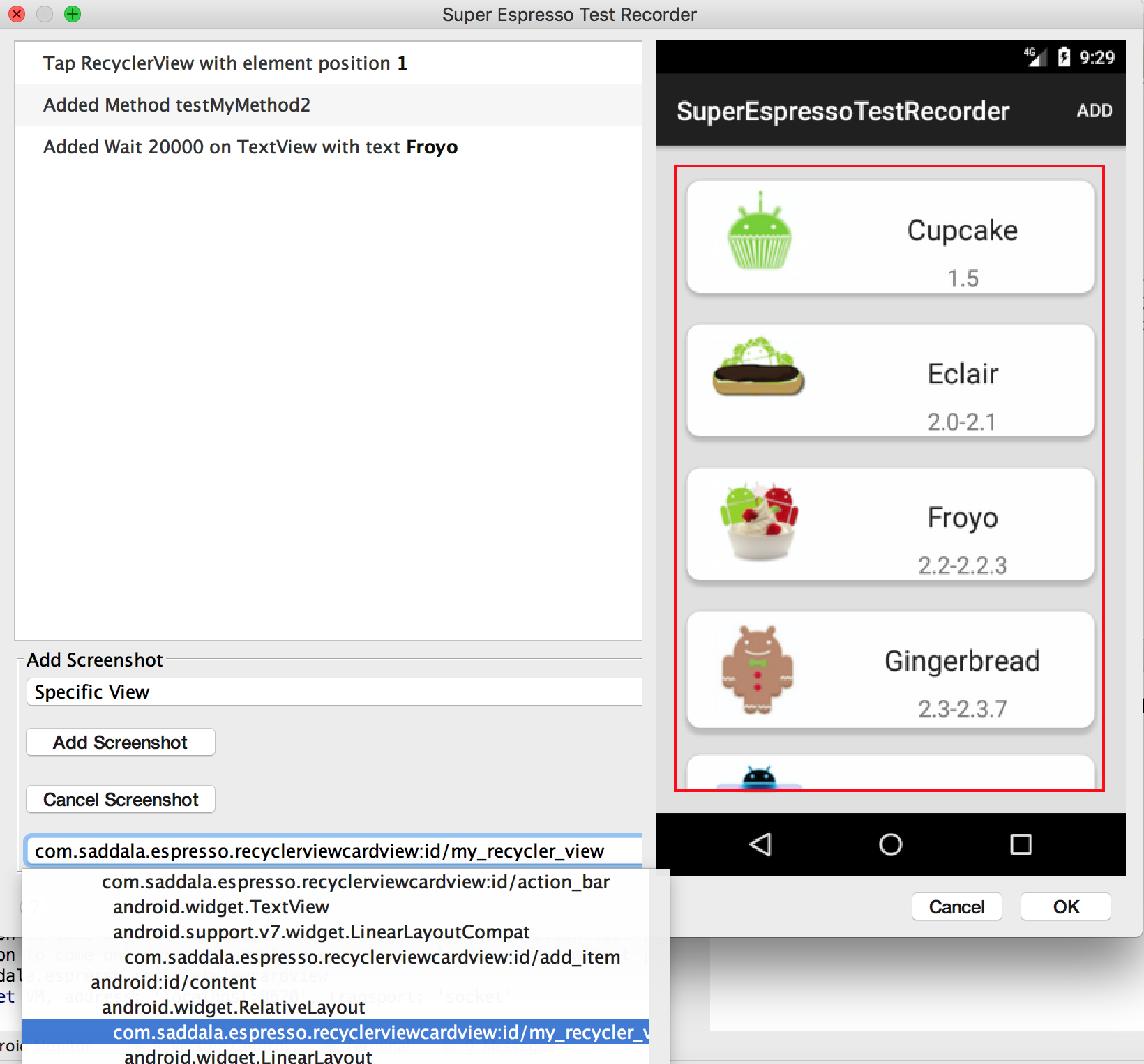
**Add Screenshot:**

Adding screenshot can be of three types. User can select a dialog view or any view in the rendered screen or root view which is default. Combo box provided to select among these three options.





When User selects Specific view, layout hierarchy will be provided in the combo box and rendered screen also will be shown. User can select a particular view or view group from the rendered screen or combo box.



Once we done with the test scripts recording, user can select main OK button to stop the recording, Test Class file will be created in the

com.saddala.espresso.<appname>.view.tests package

**Dependent Framework:**

Custom dependent project for this Super Espresso Test Recorder is added into this project. All the files are incuded into the test app’s androidTest folder.

Running the Recorded Tests:

We have used Facebook’s screenshot-tests-for-android plugin to generate screenshots and execute recorded tests. Running recorded tests will be depended on the Facebook’s plugin. You can check this page [here](http://facebook.github.io/screenshot-tests-for-android/) to execute your tests.

**Documentation**

There are some links which can be useful to work with Super Espresso Test Recorder:

* [Screenshot Testing Framework](http://facebook.github.io/screenshot-tests-for-android/)
* [Screenshot Tests For Android](http://facebook.github.io/screenshot-tests-for-android/)
* [Android Testing Support Library official documentation](https://google.github.io/android-testing-support-library)
* [Espresso Cheat Sheet](https://google.github.io/android-testing-support-library/docs/espresso/cheatsheet/index.html)
* [Espresso Idling Resources](http://dev.jimdo.com/2014/05/09/wait-for-it-a-deep-dive-into-espresso-s-idling-resources/)

Data provided by Saddala. © 2017 SADDALA

#License

Copyright 2017 Saddala

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.