

Flutter Developer

Flutter App Life Cycle

1- resumed → The application is visible and responding to user input.

2- inactive → The application is in an inactive state and is not receiving user input.

On iOS, this state corresponds to an app or the Flutter host view running in the foreground inactive state.

Apps transition to this state when in a phone call, responding to a TouchID request, when entering the app switcher

or the control center, or when the UIViewController hosting the Flutter app is transitioning.

On Android, this corresponds to an app or the Flutter host view running in the foreground inactive state.

Apps transition to this state when another activity is focused, such as a split-screen app, a phone call, a picture-in-picture app, a system dialog, or another window. Apps in this state should assume that they may be paused at any time.

3- paused → The application is not currently visible to the user, not responding to user input, and running in the background. When the application is in this state, the engine will not call the `PlatformDispatcher.onBeginFrame` and `PlatformDispatcher`.

`onDrawFrame` callbacks.

4- detached → The application is still hosted on a flutter engine but is detached from any host views.

When the application is in this state, the engine is running without a view. It can either be in the progress of

attaching a view when engine was first initializes, or after the view being destroyed due to a Navigator pop.