



100 Flutter Interview Questions

part - 2

#flutterdaily



Sayed Abdulaziz



@sayedabdulaziz

11) Can you nest a Scaffold? Why or why not?

Yes, you can nest a Scaffold. That's the beauty of Flutter. You control the entire UI.

Scaffold is just a widget, so you can put it anywhere a widget might go. By nesting a, you can layer drawers, snack bars, and bottom sheets.



12) Describe Some of the major features of Flutter.

- Fast Development — With the use of a rich set of fully customizable widgets, you can build native interfaces in minutes with Flutter.
- Expressive and Flexible UI — The layered architecture present with the Flutter enables you to fully customize your UI. This results in fast rendering and expressive designs.
- Native Performance — The widgets present in the Flutter incorporate all the critical platform differences such as scrolling, navigation, icons, and more. It gives a full native performance on all platforms



13) How do you reduce widget rebuild?

You rebuild widgets when the state changes. This is normal and desirable because it allows the user to see the state changes reflected in the UI. However, rebuilding parts of the UI that don't need to change is wasteful.

There are several things you can do to reduce unnecessary widget rebuilding.

- The first is to refactor a large widget tree into smaller individual widgets, each with its build method.
- Whenever possible, use the const constructor, because this will tell Flutter that it doesn't need to rebuild the widget.
- Keep the subtree of a stateful widget as small as possible. If a stateful widget needs to have a widget subtree under it, create a custom widget for the stateful widget and give it a child parameter.



14) What is BuildContext and how is it useful?

BuildContext is the widget's element in the Element tree — so every widget has its own BuildContext.

You usually use BuildContext to get a reference to the theme or another widget. For example, if you want to show a material dialog, you need a reference to the scaffold. You can get it with Scaffold.of(context), where context is the build context. of() searches up the tree until it finds the nearest scaffold.



15) How do you talk to native code from within a Flutter app?

Normally you don't need to talk to native code because the Flutter framework or third-party plugins handle it. However, if you do find yourself needing to get special access to the underlying platform, you can use platform channels.

- One type of platform channel is a method channel. Data is serialized on the Dart side and then sent to the native side. You can write native code to interact with the platform before sending a serialized message back. That message might be written in Java or Kotlin on Android or Objective-C or Swift on iOS.

You don't use platform channels on the web, however, because they're an unnecessary step.

- The second type of platform channel is the event channel, which you use to send a stream of data from the native platform back to Flutter. This is useful for monitoring sensor data.



16) What types of tests can you perform?

There are three main kinds of tests: unit tests, widget tests, and integration tests.

- Unit tests are all about checking the validity of your business logic.
- Widget tests are for making sure UI widgets have the components that you expect them.
- Integration tests check that your app is working as a whole.



17) What are the pros and cons of different state management solutions?

While there are countless varieties, some of the more popular state management solutions include BLoC, ChangeNotifier with Provider, Redux, MobX, and RxDart. These are all appropriate for medium- to large-scale apps, if you're only making a quick demo app, then a stateful widget is often enough.

Instead of listing the pros and cons of each state management option, it's more useful to look at the situations where a certain class of solutions is a better fit. For example, for someone who's overwhelmed with the sheer number of options, it's important to choose a solution that's easy to grasp, mentally. ChangeNotifier with Provider or MobX would be a good choice because it makes sense to directly call methods on the state class in response to events.

If you're heavily reliant on streams, such as with a Firebase API, then it's natural to choose a stream-based solution like BLoC or RxDart.



18) What is Stream in Flutter?

A stream is a sequence of asynchronous events. It provides an asynchronous sequence of data. It is the same as a pipe where we put some value on one end, and if we have a listener on the other end, it will receive that value. We can keep multiple listeners in a stream, and all of those will receive the same value when put in the pipeline.



19) How to create private variables in Dart?

In dart '_' is used before the variable name to declare it as private. Unlike other programming languages, here private doesn't mean it is available only to the class it is in, private means it is accessible in the file it is in and not accessible to other files.



20) What is the event loop, and what is its relationship to isolates?

In Flutter, the event loop is a central concept that helps manage the flow of control within an app. It is responsible for processing events and updating the app's state in response to those events.

An isolate is a separate thread of execution that is isolated from the main thread of the app. Isolates are used in Flutter to allow concurrent execution of code, which can be useful for tasks that might take a long time to complete, such as network requests or computationally intensive operations.

The event loop is related to isolates in that it is responsible for coordinating the execution of code across multiple isolates, if they are used in the app. It does this by sending messages between isolates and scheduling the execution of code on the appropriate isolate.

Overall, the event loop plays a crucial role in the operation of a Flutter app, helping to manage the flow of control and coordinate the execution of code across multiple isolates.



Like & follow



Share post



Thank you



@sayedabdulaziz