**Stateful vs Stateless Widget**

Before diving into the stateful vs stateless widget debate, let’s discuss Flutter’s development environment a little. Unlike React, Kotlin, or Swift, where you build components from scratch. Flutter has plenty of pre-defined widgets (modules) that you play around with and build an interactive and seamless cross-platform application.

Now onto the real deal:

**Stateful widget:**

When working on application screens that require continuous or action-based updates, developers implement the stateful widget. We take an example of CS 1.6/ Valorant or any of your favorite FPS game(s). You have a gun in your hand with a fixed bullet count per magazine. Every time you fire, the bullet counter goes to 0 in real-time. This is “updating” the state (which in this case is the bullet counter), and vice versa happens when you reload. The same principle can also be applied to the health bar and inventory, where your data is dynamic and not idle.

**Stateless widget:**

You might think why do we need the stateless widget at all when you can simply work on dynamic data using the stateful widget, but this is where classification helps optimize both your code and development resources. Think about WhatsApp for an instant here. Every time you message someone, does the message interface update? The answer is no; it doesn’t because here the overall state (which is your user interface) does not need changing, but only the messages you type that change with each response. This is why we separate the data types based on usage and work on the intricacies of app design to accommodate the user as best we can, making the application as streamlined as possible.