Search API Docs

```
TabController(vsync: tickerProvider, length: tabCount)..addListener(() {
  if (!tabController.indexIsChanging) {
    setState(() {
      // Rebuild the enclosing scaffold with a new AppBar title
      appBarTitle = 'Tab ${tabController.index}';
    });
  }
})
```

Although there are some use cases, like a presentation app that shows embedded flutter content, where nested scaffolds are appropriate, it's best to avoid nesting scaffolds.

See also:

- AppBar, which is a horizontal bar typically shown at the top of an app using the appBar property.
- BottomAppBar, which is a horizontal bar typically shown at the bottom of an app using the bottomNavigationBar property.
- FloatingActionButton, which is a circular button typically shown in the bottom right corner of the app using the floatingActionButton property.
- Drawer, which is a vertical panel that is typically displayed to the left of the body (and often hidden on phones) using the drawer property.
- BottomNavigationBar, which is a horizontal array of buttons typically shown along the bottom of the app using the bottomNavigationBar property.
- BottomSheet, which is an overlay typically shown near the bottom of the app. A bottom sheet can either be persistent, in which case it is shown using the ScaffoldState.showBottomSheet method, or modal, in which case it is shown using the showModalBottomSheet function.
- ScaffoldState, which is the state associated with this widget.
- material.io/design/layout/responsive-layout-grid.html
- Cookbook: Add a Drawer to a screen

Inheritance

Object DiagnosticableTree Widget StatefulWidget Scaffold

Constructors