

Display SnackBar on top of AlertDialog widget

Asked 6 months ago Active 1 month ago Viewed 1k times



7



I have an `AlertDialog` widget that will cause a `SnackBar` to display when you tap on its `Text`. The `SnackBar` currently displays behind the `AlertDialog` barrier, in the background. I want the `SnackBar` to display on top of the transparent `AlertDialog` barrier instead. Is the behavior that I'm seeking possible to achieve in Flutter? I have created a brand new Flutter app and included only the relevant code to illustrate the use-case below, as well as a screenshot.



[Main.dart](#) Gist

1



```
@override
Widget build(BuildContext context) {
  WidgetsBinding.instance!.addPostFrameCallback((_) async {
    showDialog(
      context: context,
      builder: (BuildContext dialogContext) => AlertDialog(
        content: GestureDetector(
          onTap: () {
            ScaffoldMessenger.of(dialogContext).showSnackBar(SnackBar(
              content: const Text('snack'),
              duration: const Duration(seconds: 1),
              action: SnackBarAction(
                label: 'ACTION',
                onPressed: () {},
              ),
            ));
          },
          child: Center(
            child: Text('Show SnackBar!'),
          ),
        ),
      ),
    );
  });
  // This method is rerun every time setState is called, for instance as done
  // by the _incrementCounter method above.
  //
  // The Flutter framework has been optimized to make rerunning build methods
```

```
// fast, so that you can just rebuild anything that needs updating rather
// than having to individually change instances of widgets.
return Scaffold(
  appBar: AppBar(
    // Here we take the value from the MyHomePage object that was created by
    // the App.build method, and use it to set our appbar title.
    title: Text(widget.title),
  ),
  body: Center(
    // Center is a layout widget. It takes a single child and positions it
    // in the middle of the parent.
    child: Column(
      // Column is also a layout widget. It takes a list of children and
      // arranges them vertically. By default, it sizes itself to fit its
      // children horizontally, and tries to be as tall as its parent.
      //
      // Invoke "debug painting" (press "p" in the console, choose the
      // "Toggle Debug Paint" action from the Flutter Inspector in Android
      // Studio, or the "Toggle Debug Paint" command in Visual Studio Code)
      // to see the wireframe for each widget.
      //
      // Column has various properties to control how it sizes itself and
      // how it positions its children. Here we use mainAxisAlignment to
      // center the children vertically; the main axis here is the vertical
      // axis because Columns are vertical (the cross axis would be
      // horizontal).
      mainAxisAlignment: MainAxisAlignment.center,
      children: <Widget>[
        Text(
          'You have pushed the button this many times:',
        ),
        Text(
          '$_counter',
          style: Theme.of(context).textTheme.headline4,
        ),
      ],
    ),
  ),
  floatingActionButton: FloatingActionButton(
    onPressed: _incrementCounter,
    tooltip: 'Increment',
    child: Icon(Icons.add),
  ), // This trailing comma makes auto-formatting nicer for build methods.
);
}
```



[flutter](#) [dart](#) [flutter-layout](#)

[Share](#) [Follow](#)

edited Jul 19 '21 at 18:37



[Christopher Moore](#)

11.9k 9 28 44

asked Jul 19 '21 at 18:31



[Josh Kautz](#)

320 2 3 21

4 Answers

Active

Oldest

Votes

▲ Update

5



Thanks to Amy, I realized that tapping on the barrier did not dismiss the dialog. Also, the code was causing to show multiple SnackBars due to the use of nested Scaffolds.



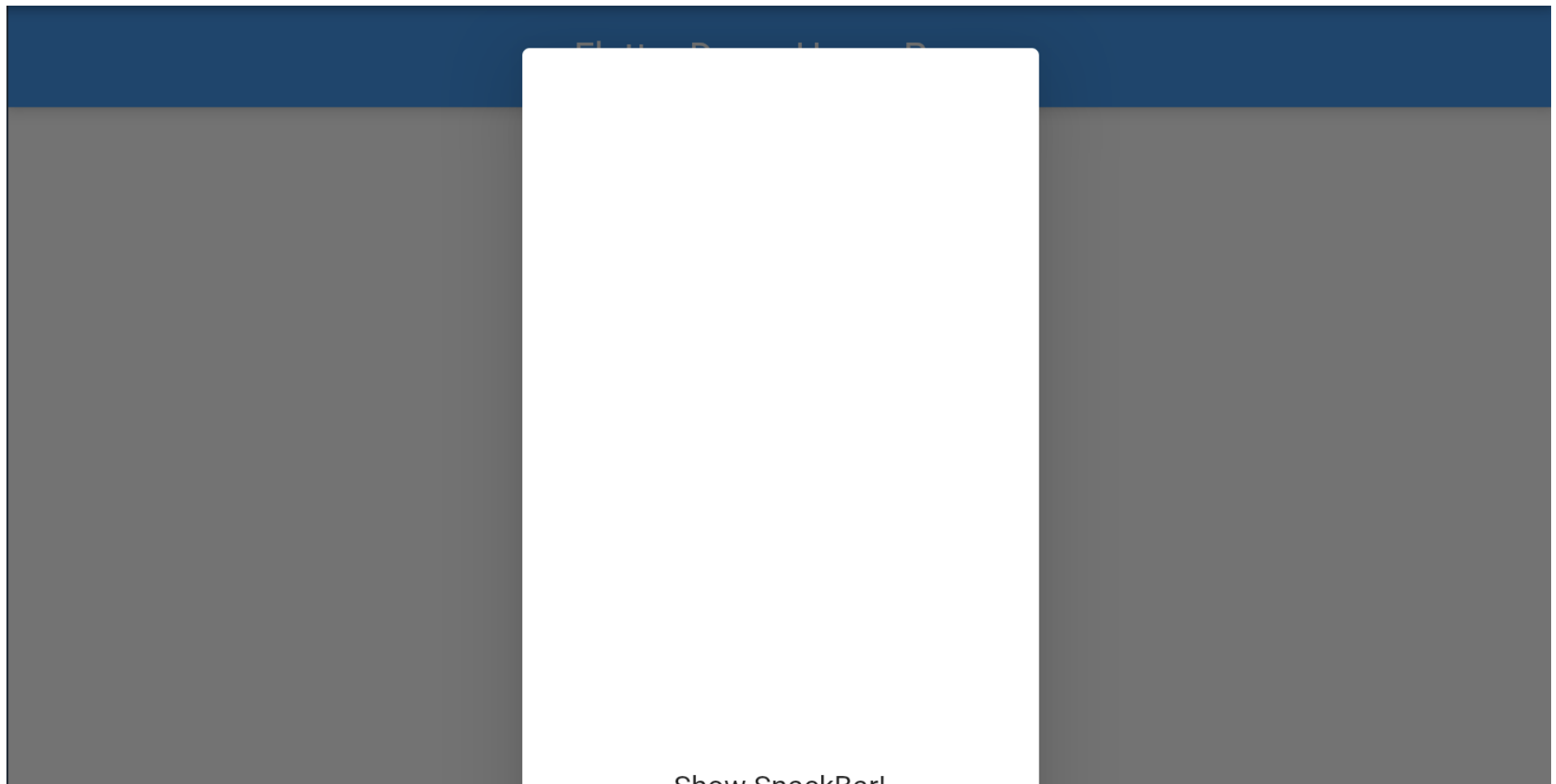
Check out the following model that fixes all issues:

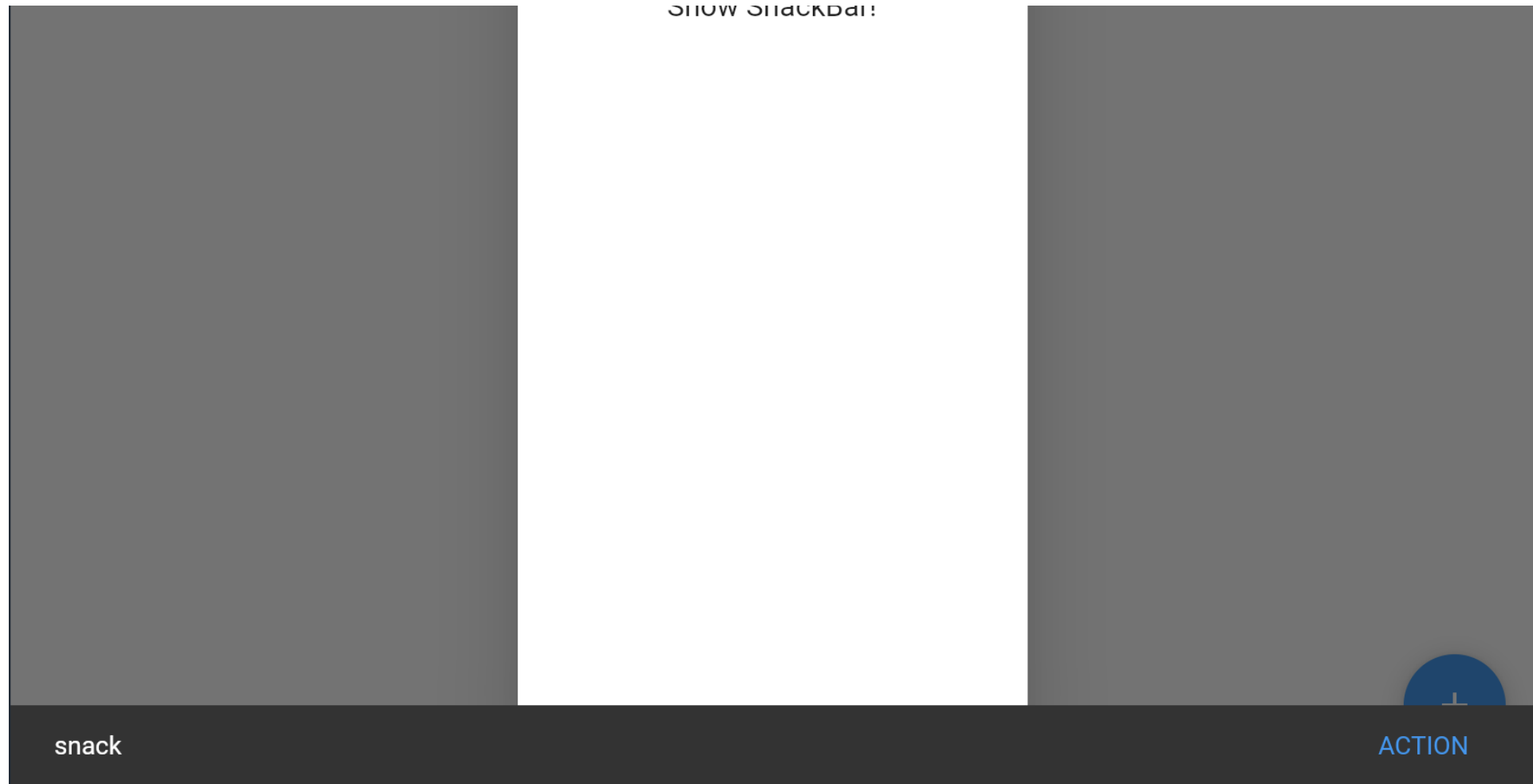
```
showDialog
|
|
ScaffoldMessenger => "Set a scope to show SnackBars only in the inner Scaffold"
|
--- Builder => "Add a Builder widget to access the Scaffold Messenger"
|
--- Scaffold => "The inner Scaffold that is needed to show SnackBars"
|
--- GestureDetector => "Dismiss the dialog when tapped outside"
|
--- GestureDetector => "Don't dismiss it when tapped inside"
|
--- AlertDialog => "Your dialog"
```

Here is the implementation:

```
showDialog(
  context: context,
  builder: (context) => ScaffoldMessenger(
    child: Builder(
      builder: (context) => Scaffold(
        backgroundColor: Colors.transparent,
        body: GestureDetector(
          behavior: HitTestBehavior.opaque,
          onTap: () => Navigator.of(context).pop(),
          child: GestureDetector(
            onTap: () {},
            child: AlertDialog(
              content: GestureDetector(
                onTap: () {
                  ScaffoldMessenger.of(context).showSnackBar(
                    SnackBar(
                      content: const Text('snack'),
                      duration: const Duration(seconds: 1),
                      action: SnackBarAction(
                        label: 'ACTION',
                        onPressed: () {},
                      ),
                    ),
                  ),
                ),
              ),
            ),
          ),
        ),
      ),
    ),
  ),
)
```

```
    );  
  },  
  child: Center(  
    child: Text('Show SnackBar!'),  
  ),  
),  
),  
),  
),  
),  
),  
),  
),  
);
```





Old answer

~~ScaffoldMessenger shows SnackBar in the nearest descendant Scaffold. If you add another Scaffold before AlertDialog, it will use it instead of the root one which is left behind the dialog.~~

```
showDialog(  
  context: context,  
  builder: (BuildContext dialogContext) => Scaffold(  
    backgroundColor: Colors.transparent, // Make Scaffold's background transparent  
    body: AlertDialog(  

```

```

content: GestureDetector(
  onTap: () {
    ScaffoldMessenger.of(dialogContext).showSnackBar(SnackBar(
      content: const Text('snack'),
      duration: const Duration(seconds: 1),
      action: SnackBarAction(
        label: 'ACTION',
        onPressed: () {},
      ),
    ));
  },
  child: Center(
    child: Text('Show SnackBar!'),
  ),
),
),
),
);

```

Share Follow

edited Oct 15 '21 at 9:37

answered Jul 19 '21 at 20:07



Stewie Griffin

2,563 13 29

1 Then you can't click off the dialog to dismiss it – Amy Oct 14 '21 at 22:52

1 Thanks for letting me know. I edited my answer. – Stewie Griffin Oct 15 '21 at 9:32



3



The issue here is that `showDialog` uses the root navigator provided by `MaterialApp`. So when you show your dialog it is pushed completely over your scaffold. To solve this you need the navigator that is used to be a *child* of the scaffold that's showing the snackbars. So the following code adds this navigator, sets `useRootNavigator` to `false` to use this navigator, and importantly uses a `BuildContext` *under* the newly created navigator:

```

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text(widget.title),

```

```

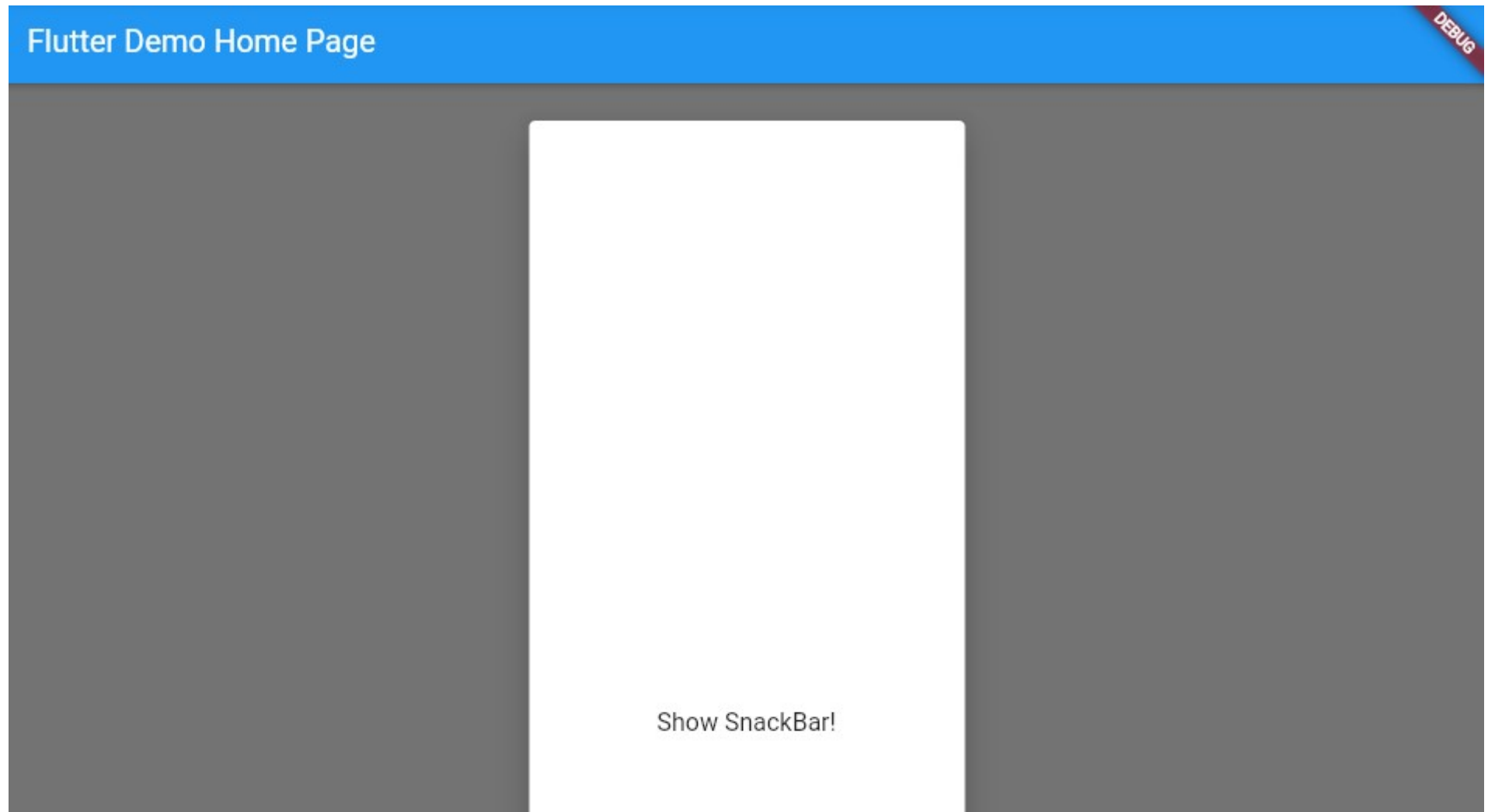
),
body: Navigator(      //New navigator added here
  initialRoute: '/',
  onGenerateRoute: (setting) {
    return MaterialPageRoute(
      builder: (context) => Center(
        child: Builder(builder: (context) {
          WidgetsBinding.instance!
            .addPostFrameCallback((_) async {
              showDialog(
                context: context,
                useRootNavigator: false, //Dialog must not use root navigator
                builder: (BuildContext dialogContext) =>
                  AlertDialog(
                    content: GestureDetector(
                      onTap: () {
                        ScaffoldMessenger.of(dialogContext)
                          .showSnackBar(SnackBar(
                            content: const Text('snack'),
                            duration: const Duration(seconds: 1),
                            action: SnackBarAction(
                              label: 'ACTION',
                              onPressed: () {},
                            ),
                          ),
                      ));
                    child: Center(
                      child: Text('Show SnackBar!'),
                    ),
                  ),
                ),
              );
            });
    return Column(
      mainAxisAlignment: MainAxisAlignment.center,
      children: <Widget>[
        Text(
          'You have pushed the button this many times:',
        ),
        Text(
          '$_counter',
          style: Theme.of(context).textTheme.headline4,
        ),
      ]);
    },
  ),
),

```



```
    ));  
  },  
  floatingActionButton: FloatingActionButton(  
    onPressed: _incrementCounter,  
    tooltip: 'Increment',  
    child: Icon(Icons.add),  
  ), // This trailing comma makes auto-formatting nicer for build methods.  
);  
}
```

Result:





Note that this solution does constrain the dialog size a bit and the app bar and floating action button is above the content, which may be undesirable. This can be solved just by adding another scaffold below the newly created navigator and moving those appBar/FAB properties down as desired. Example with `AppBar` below the modal:

```
@override
Widget build(BuildContext context) {
  return Scaffold(
    body: Navigator(
      initialRoute: '/',
      onGenerateRoute: (setting) {
        return MaterialPageRoute(
          builder: (context) => Scaffold(
            appBar: AppBar(
              title: Text(widget.title),
            ),
            body: Center(
              child: Builder(builder: (context) {
                WidgetsBinding.instance!
```

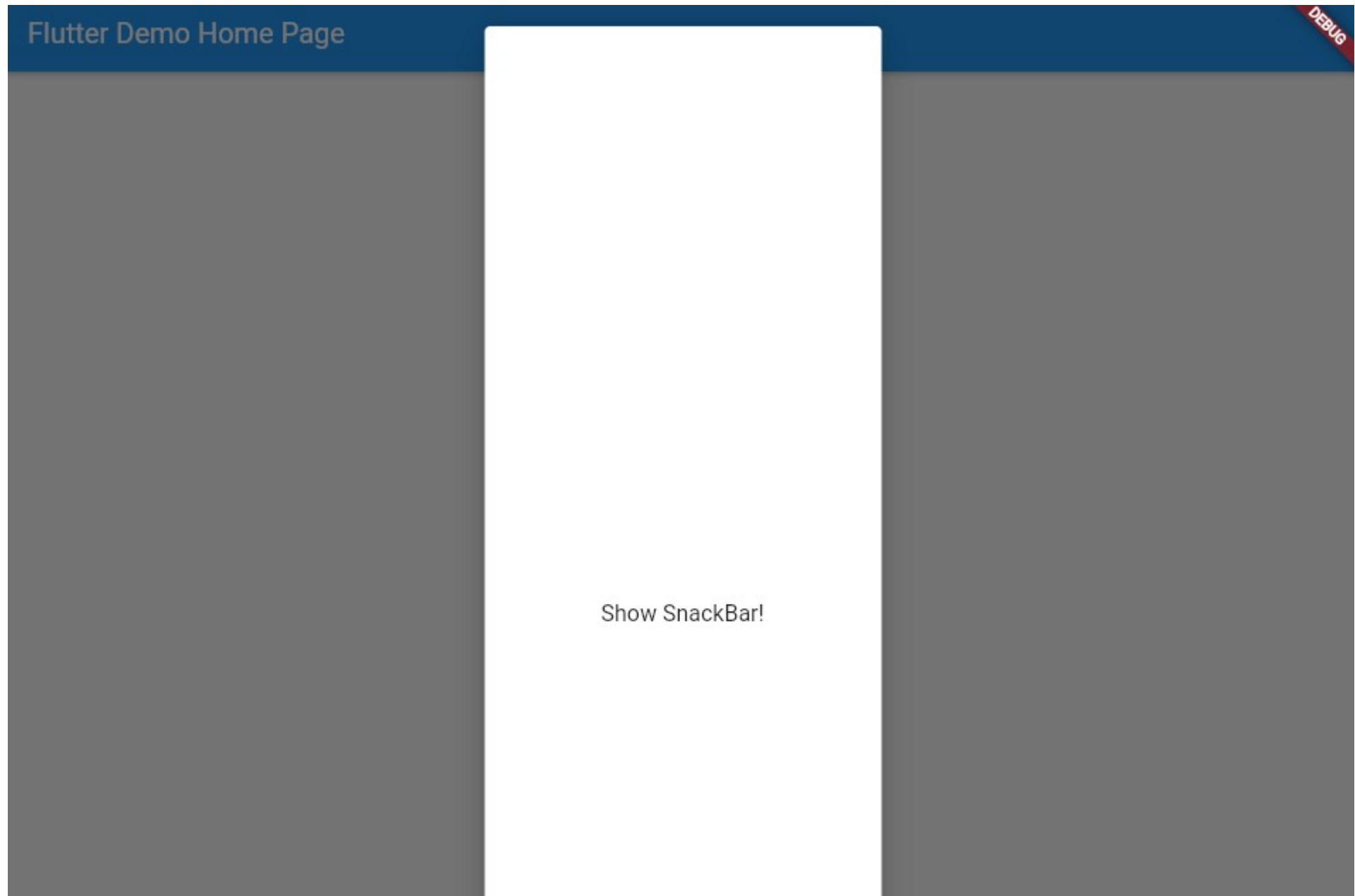
```

        .addPostFrameCallback((_) async {
showDialog(
  context: context,
  useRootNavigator: false,
  builder: (BuildContext dialogContext) =>
    AlertDialog(
      content: GestureDetector(
        onTap: () {
          ScaffoldMessenger.of(dialogContext)
            .showSnackBar(SnackBar(
              content: const Text('snack'),
              duration: const Duration(seconds: 1),
              action: SnackBarAction(
                label: 'ACTION',
                onPressed: () {},
              ),
            ));
        },
        child: Center(
          child: Text('Show SnackBar!'),
        ),
      ),
    ),
  );
});
return Column(
  mainAxisAlignment: MainAxisAlignment.center,
  children: <Widget>[
    Text(
      'You have pushed the button this many times:',
    ),
    Text(
      '$_counter',
      style: Theme.of(context).textTheme.headline4,
    ),
  ]);
    }),
  ));
  ),
  floatingActionButton: FloatingActionButton(
    onPressed: _incrementCounter,
    tooltip: 'Increment',
    child: Icon(Icons.add),
  ), // This trailing comma makes auto-formatting nicer for build methods.

```

```
);  
}
```

Result:





Share Follow

edited Jul 19 '21 at 19:44

answered Jul 19 '21 at 19:39



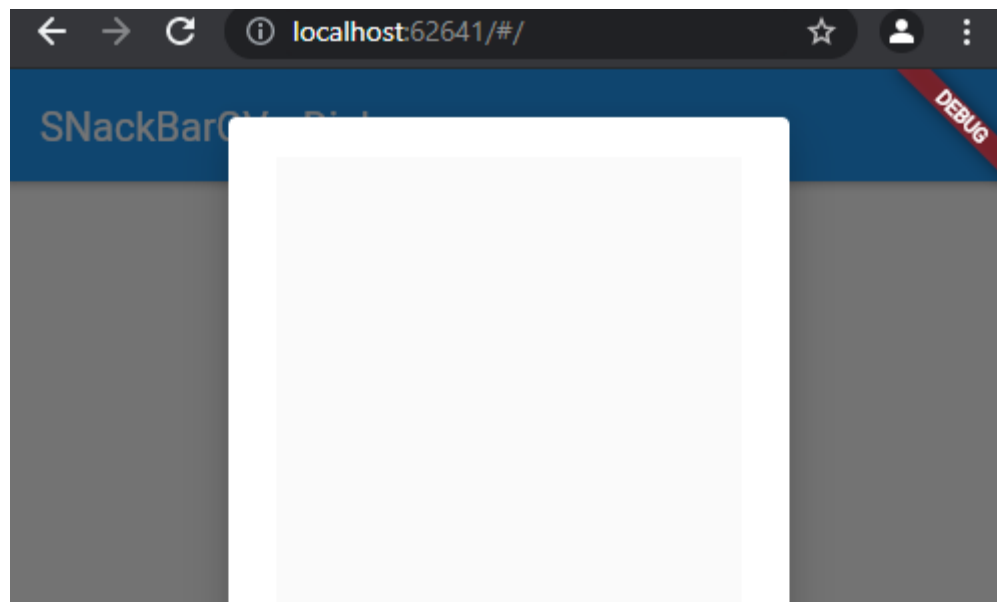
[Christopher Moore](#)

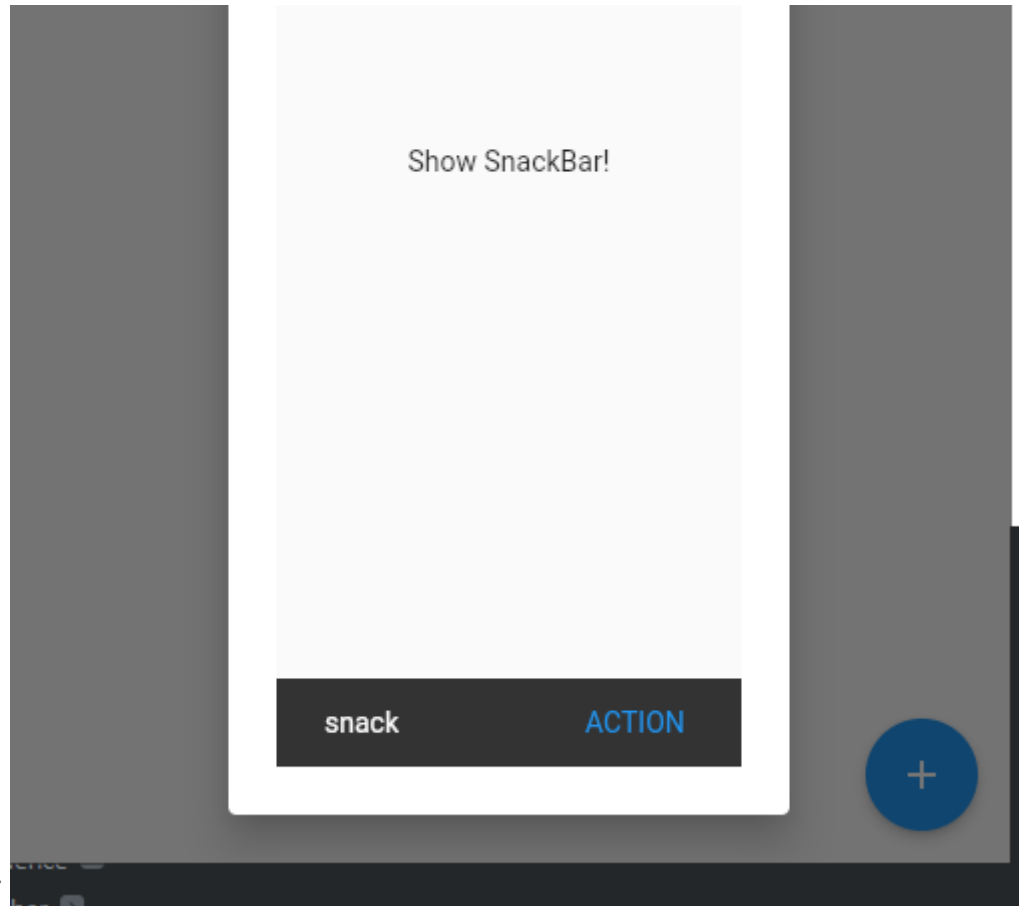
11.9k 9 28 44

This is fantastic. A great explanation. Thanks a ton Christopher! Greatly appreciated! – [Josh Kautz](#) Jul 20 '21 at 0:40



1





hope this is what you are looking for

```
import 'package:flutter/material.dart';

class SnackOverDialog extends StatefulWidget {
  SnackOverDialog({Key? key}) : super(key: key);

  @override
  _SnackOverDialogState createState() => _SnackOverDialogState();
}

class _SnackOverDialogState extends State<SnackOverDialog> {
  final GlobalKey<ScaffoldState> _scaffoldkey = new GlobalKey<ScaffoldState>();
  @override
  Widget build(BuildContext context) {
    ///* show snack
    _snackbar(BuildContext context) {
```

```

_scaffoldkey.currentState!.showSnackBar(SnackBar(
  content: const Text('snack'),
  duration: const Duration(seconds: 1),
  action: SnackBarAction(
    label: 'ACTION',
    onPressed: () {},
  ),
));
}

///  

_dialog(BuildContext context) {
  WidgetsBinding.instance!.addPostFrameCallback((_) async {
    showDialog(
      context: context,
      builder: (BuildContext dialogContext) => AlertDialog(
        content: Scaffold(
          key: _scaffoldkey,
          body: GestureDetector(
            onTap: () {
              _snackbar(dialogContext);
            },
            child: Center(
              child: Text('Show SnackBar!'),
            ),
          ),
        ),
      ),
    );
  });
}

return Scaffold(
  appBar: AppBar(
    title: Text("SnackBarOverDialog"),
  ),
  body: Center(
    child: Column(
      mainAxisAlignment: MainAxisAlignment.center,
      children: <Widget>[
        Text(
          'You have pushed the button this many times:',
        ),
      ],
    ),
  ),
);

```

```
    ),  
    floatingActionButton: FloatingActionButton(  
      onPressed: () => _dialog(context),  
      tooltip: 'Increment',  
      child: Icon(Icons.add),  
    ),  
  );  
}  
}
```

Share Follow

answered Jul 19 '21 at 19:19



Yeasin Sheikh

10.2k

2

12

28



Instead of the SnackBar Use [another flushbar](#), It will Appear Above AlertDialog.

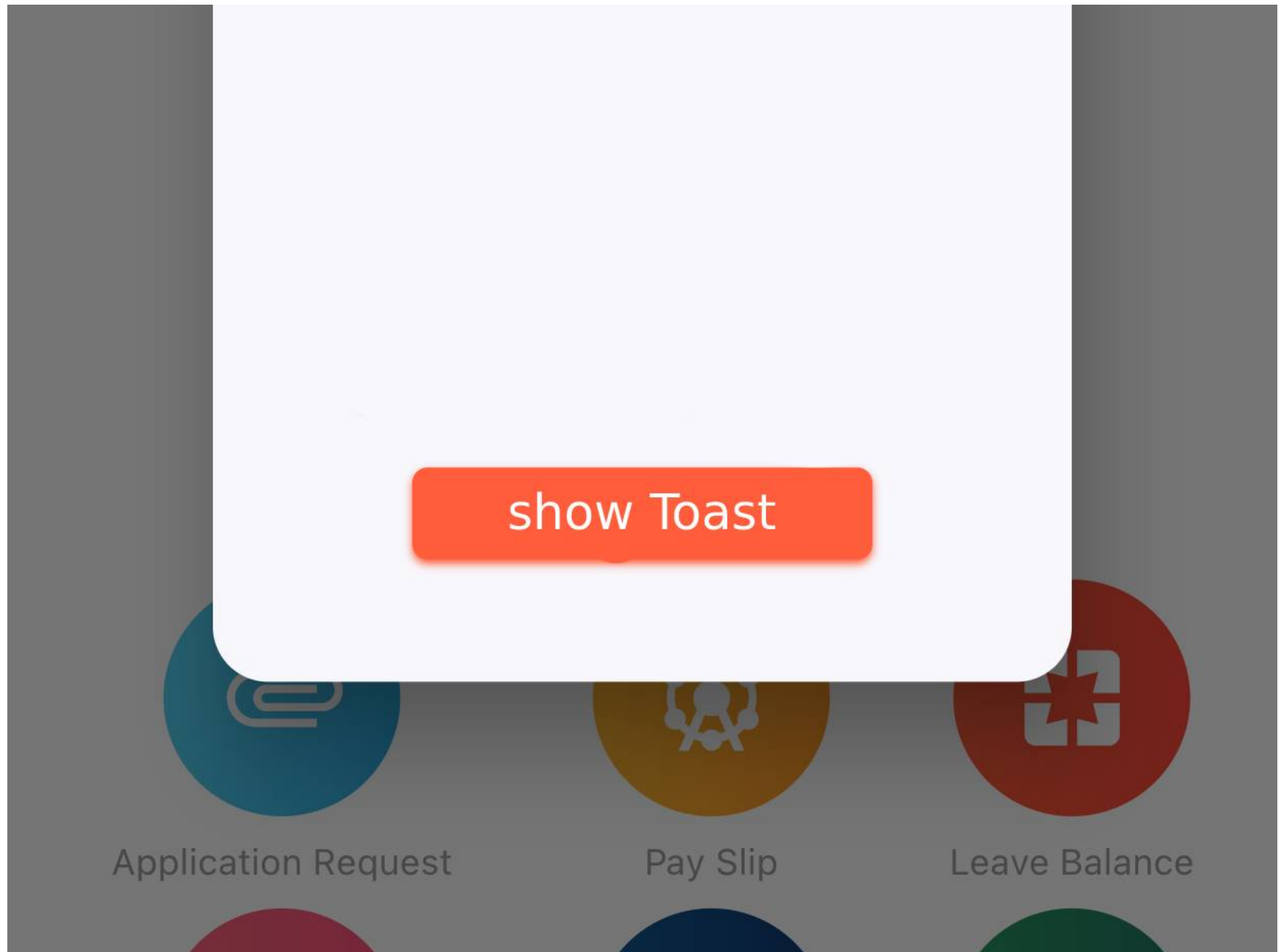
0



```
Flushbar(  
  backgroundColor: Colors.red,  
  message: S.of(context).choose_date,  
  duration: Duration(seconds: Constants.TOAST_DURATION),  
).show(context);
```

Result:







Share Follow

edited Dec 2 '21 at 8:08

answered Dec 2 '21 at 7:45



[MRazalmtiaz](#)

1,036

1

7

19