

# Flutter Bottom Navigation

## Basic Usage

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

class MyHomePage extends StatefulWidget {
  @override
  _MyHomePageState createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {
  int _selectedIndex = 0;

  void _onItemTapped(int index) {
    setState(() {
      _selectedIndex = index;
    });
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text("Bottom Navigation Example")),
      body: Center(child: Text("Selected Index: $_selectedIndex")),
      bottomNavigationBar: BottomNavigationBar(
        items: [
          BottomNavigationBarItem(icon: Icon(Icons.home), label: 'Home'),
          BottomNavigationBarItem(icon: Icon(Icons.search), label: 'Search'),
          BottomNavigationBarItem(icon: Icon(Icons.notifications), label:
'Notifications'),
        ],
        currentIndex: _selectedIndex,
        onTap: _onItemTapped,
```

```
        ),  
    );  
}  
}
```

## Common Attributes

### - items

List of BottomNavigationBarItem widgets.

Example: See example above.

### - currentIndex

Index of the currently selected item.

Example: int \_selectedIndex

### - onTap

Callback when an item is tapped.

Example: \_onItemTapped(int index)

### - selectedItemColor

Color of the selected item.

Example: Colors.blue

### - unselectedItemColor

Color of unselected items.

Example: Colors.grey

### - showSelectedLabels

Show labels for selected items.

Example: true/false

### - showUnselectedLabels

Show labels for unselected items.

Example: true/false

## Example: Custom Bottom Navigation

```
BottomNavigationBar(  
  items: [
```

```
items: [
  BottomNavigationBarItem(icon: Icon(Icons.home), label: 'Home'),
  BottomNavigationBarItem(icon: Icon(Icons.search), label: 'Search'),
  BottomNavigationBarItem(icon: Icon(Icons.notifications), label: 'Notifications'),
],
currentIndex: _selectedIndex,
onTap: _onItemTapped,
selectedItemColor: Colors.blue,
unselectedItemColor: Colors.grey,
showSelectedLabels: true,
showUnselectedLabels: true,
)
```

## Best Practices

1. Limit the number of items to 3-5 for clarity.
2. Use intuitive icons and labels for each item.
3. Ensure accessibility by providing meaningful labels.
4. Maintain a consistent design with the app theme.