

### mutable data

```
add to lib/main.dart: This class
stores mutable data which could
change over the lifetime
class _FavoriteWidgetState
extends State<FavoriteWidget> {
  // ...
  @override
  Widget build(BuildContext
context) {
    return Row(
      mainAxisAlignment: MainAxisS-
ize.min,
      children: [
        Container(
          padding: EdgeInsets.a-
ll(0),
          child: IconButton(
            icon: (_isFavorited
? Icon(Icons.star) : Icon(Icons-
s.star_border)),
            color: Colors.re-
d[500],
            onPressed: _toggl-
eFavorite,
          ),
        ),
        SizedBox(
          width: 18,
          child: Container(
            child: Text('$_favo-
riteCount'),
          ),
        ),
      ],
    );
  }
}
```

### tapbox

```
the tap box is managing its owen
state
class TapboxA extends StatefulW-
idget {
  TapboxA({Key key}) :
super(key: key);
  @override
  _TapboxAState createState() =>
_TapboxAState();
}
class _TapboxAState extends
State<TapboxA> {
  bool _active = false;
  void _handleTap() {
    setState(() {
      _active = !_active;
    });
  }
  Widget build(BuildContext
context) {
    return GestureDetector(
      onTap: _handleTap,
      child: Container(
        child: Center(
          child: Text(
            _active ? 'Active' :
'Inactive',
            style: TextStyle(fo-
ntSize: 32.0, color: Colors.wh-
ite),
          ),
        ),
        width: 200.0,
        height: 200.0,
        decoration: BoxDecoration(
          color: _active ?
Colors.lightGreen[700] :
Colors.grey[600],
        ),
      );
    }
  }
}
```

### parent widget

```
manages the activity state,
implements handle checkbox
class ParentWidget extends
StatefulWidget {
  @override
  _ParentWidgetState create-
State() => _ParentWidgetState();
}
class _ParentWidgetState extends
State<ParentWidget> {
  bool _active = false;
  void _handleTapboxChanged(bool
newValue) {
    setState(() {
      _active = newValue;
    });
  }
  @override
  Widget build(BuildContext
context) {
    return Container(
      child: TapboxC(
        active: _active,
        onChanged: _handleTapbo-
xChanged,
      ),
    );
  }
}
```

### creating API

When creating API, consider using the @required annotation for any parameters(which re-exports Dart's meta.dart library):	import 'package:flutter/foundation.dart';
---	---