State and Jetpack Compose

The Mystery

By Moro

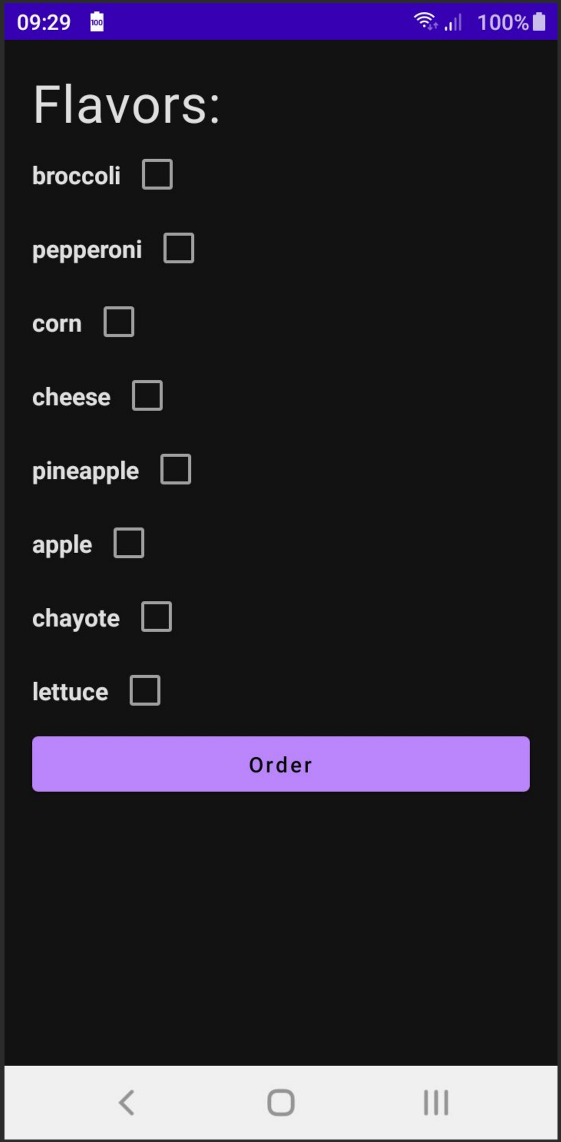
Let's learn about State in Compose

I want to eat pizza too.

Okay Scooby! Let's use pizza in the example…

Imagine a screen when you will order a pizza!

Velma, I'm sad this app is so poor. Few flavors : (



09:29 100%

Flavors:

broccoli □

pepperoni □

corn □

cheese □

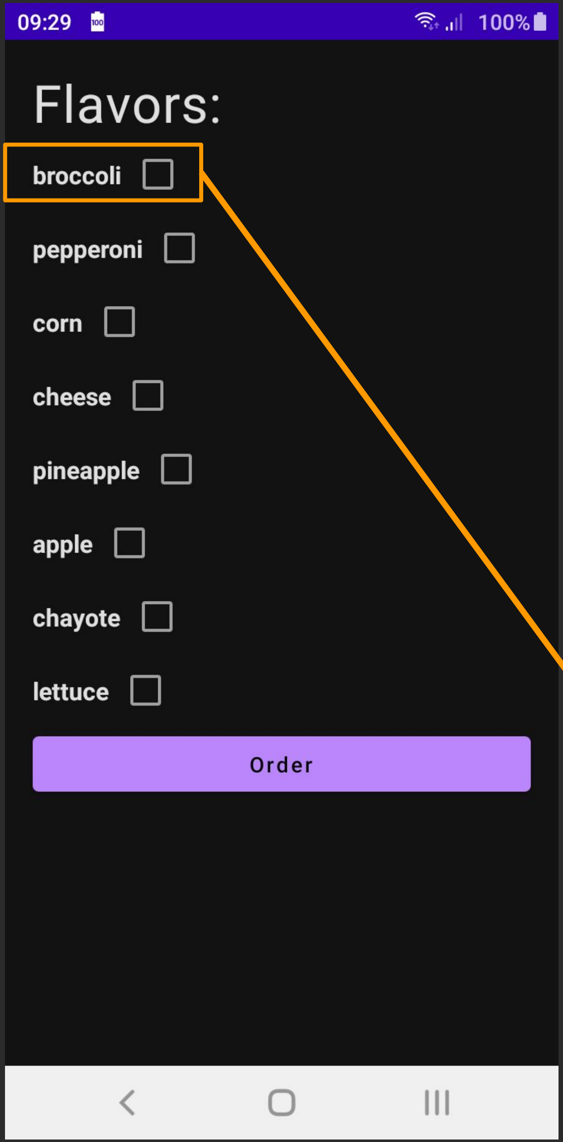
pineapple □

apple □

chayote □

lettuce □

Order



09:29 100%

Flavors:

broccoli □

pepperoni □

corn □

cheese □

pineapple □

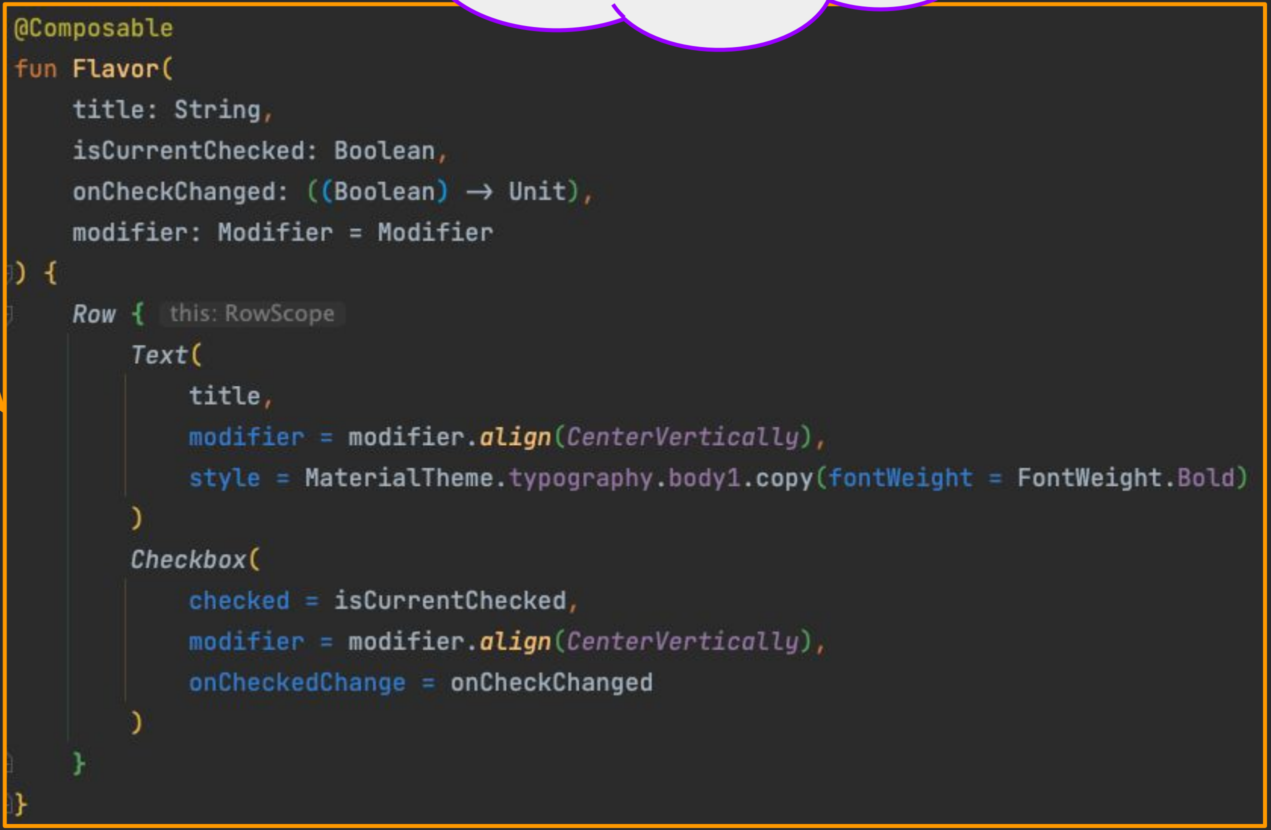
apple □

chayote □

lettuce □

Order

This component doesn't have state, so it is a ***Stateless Widget***

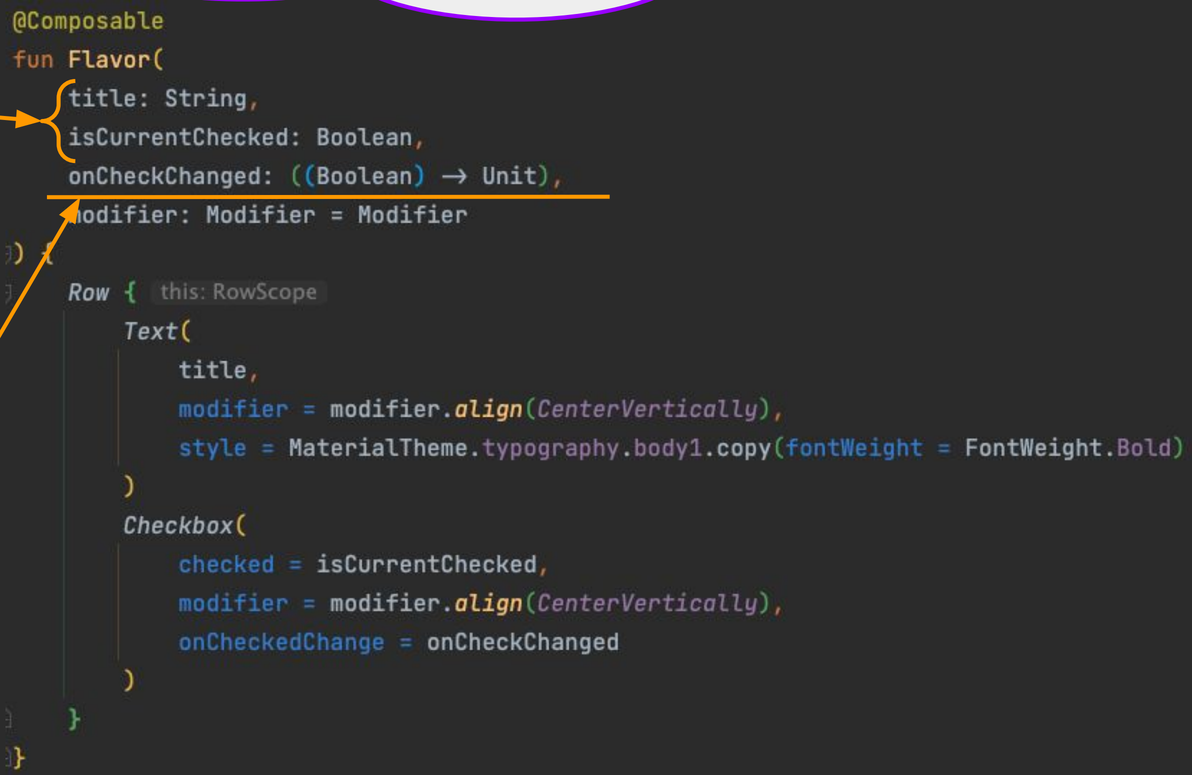


```
@Composable
fun Flavor(
    title: String,
    isCurrentChecked: Boolean,
    onCheckChanged: ((Boolean) -> Unit),
    modifier: Modifier = Modifier
) {
    Row {  this: RowScope
        Text(
            title,
            modifier = modifier.align(CenterVertically),
            style = MaterialTheme.typography.body1.copy(fontWeight = FontWeight.Bold)
        )
        Checkbox(
            checked = isCurrentChecked,
            modifier = modifier.align(CenterVertically),
            onCheckedChange = onCheckChanged
        )
    }
}
```

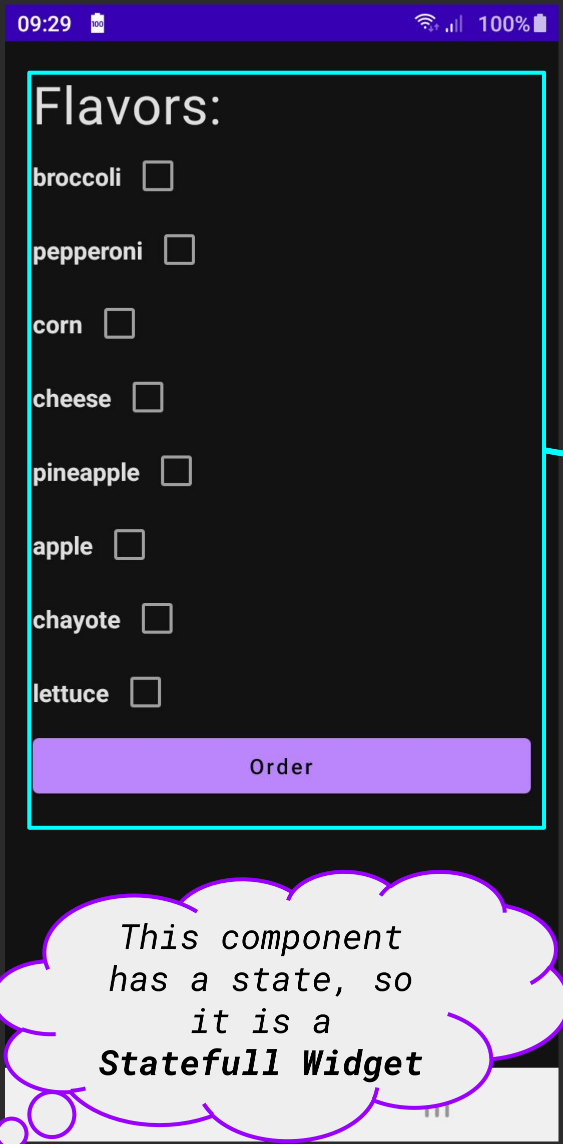
The Screen should pass the current value and a **callback** to change the state. So we have this component for each flavor

Current value to be drawn

Callback to signaling to the screen “the value was changed, we need a recomposition”



```
@Composable
fun Flavor(
    title: String,
    isCurrentChecked: Boolean,
    onCheckChanged: ((Boolean) -> Unit),
    modifier: Modifier = Modifier
) {
    Row { this: RowScope
        Text(
            title,
            modifier = modifier.align(CenterVertically),
            style = MaterialTheme.typography.body1.copy(fontWeight = FontWeight.Bold)
        )
        Checkbox(
            checked = isCurrentChecked,
            modifier = modifier.align(CenterVertically),
            onCheckedChange = onCheckChanged
        )
    }
}
```



09:29 100%

Flavors:

broccoli □

pepperoni □

corn □

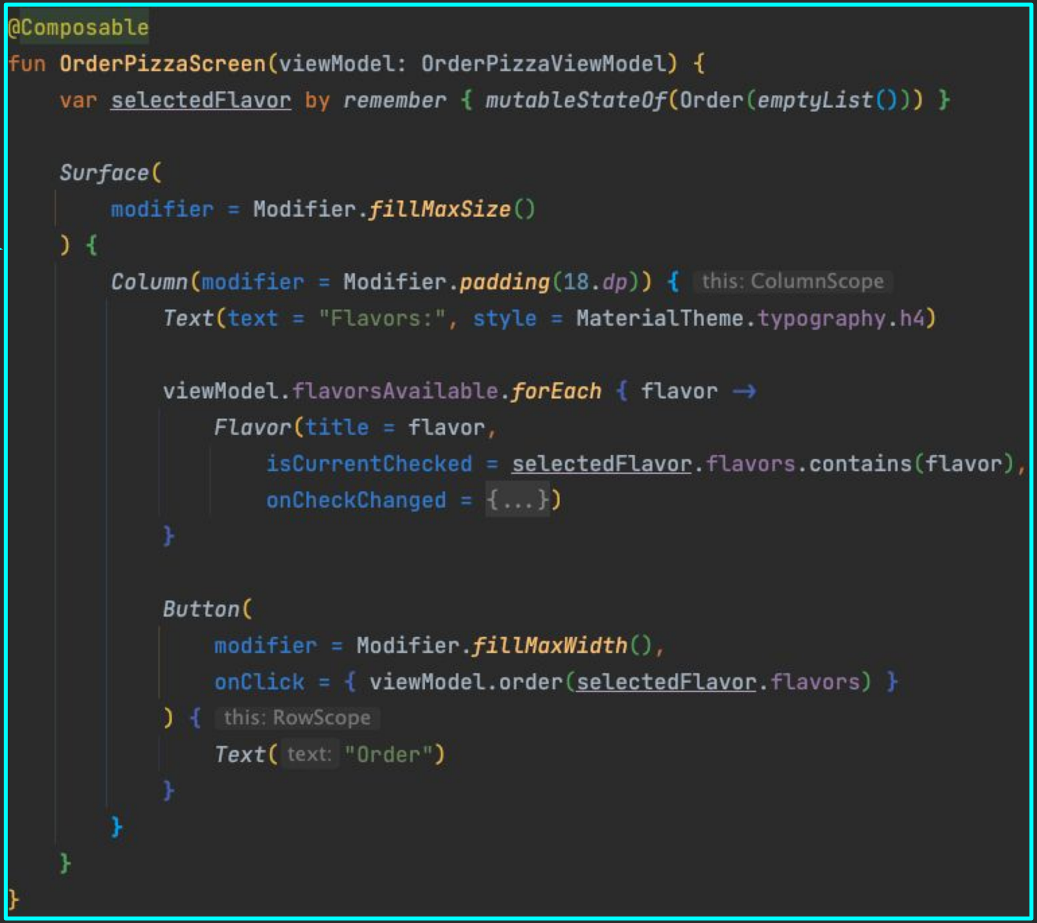
cheese □

pineapple □

apple □

chayote □

lettuce □

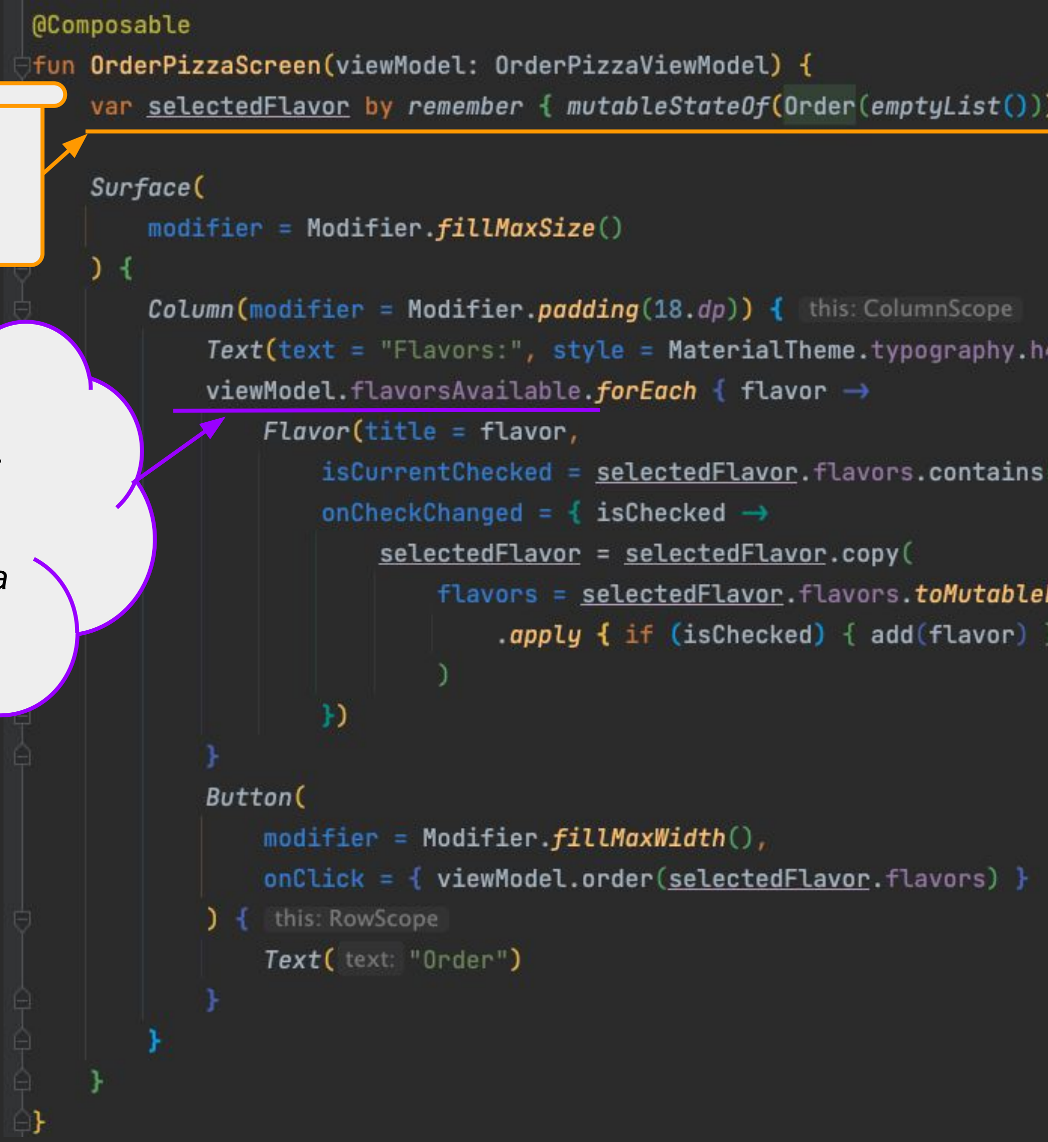


```
@Composable
fun OrderPizzaScreen(viewModel: OrderPizzaViewModel) {
    var selectedFlavor by remember { mutableStateOf(Order(emptyList())) }

    Surface(
        modifier = Modifier.fillMaxSize()
    ) {
        Column(modifier = Modifier.padding(18.dp)) {  this: ColumnScope
            Text(text = "Flavors:", style = MaterialTheme.typography.h4)

            viewModel.flavorsAvailable.forEach { flavor ->
                Flavor(title = flavor,
                    isCurrentChecked = selectedFlavor.flavors.contains(flavor),
                    onCheckChanged = {...})
            }

            Button(
                modifier = Modifier.fillMaxWidth(),
                onClick = { viewModel.order(selectedFlavor.flavors) }
            ) {  this: RowScope
                Text( text: "Order")
            }
        }
    }
}
```

This component has a state, so it is a ***Statefull Widget***

This represents the user selection

This is coming from the server. Shaggy, and Scooby, please ask the Pizzeria to add more flavors!



```kotlin
@Composable
fun OrderPizzaScreen(viewModel: OrderPizzaViewModel) {
    var selectedFlavor by remember { mutableStateOf(Order(emptyList())) }

    Surface(
        modifier = Modifier.fillMaxSize()
    ) {
        Column(modifier = Modifier.padding(18.dp)) {  this: ColumnScope
            Text(text = "Flavors:", style = MaterialTheme.typography.h4)
            viewModel.flavorsAvailable.forEach { flavor ->
                Flavor(title = flavor,
                    isCurrentChecked = selectedFlavor.flavors.contains(flavor),
                    onCheckChanged = { isChecked ->
                        selectedFlavor = selectedFlavor.copy(
                            flavors = selectedFlavor.flavors.toMutableList()
                                .apply { if (isChecked) { add(flavor) } else { remove(flavor) } }
                            )
                    })
            }
            Button(
                modifier = Modifier.fillMaxWidth(),
                onClick = { viewModel.order(selectedFlavor.flavors) }
            ) {  this: RowScope
                Text( text: "Order")
            }
        }
    }
}
```

By Moro