

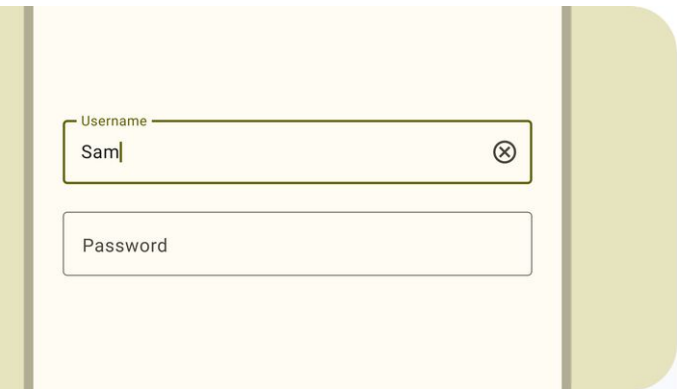
# **Сохранение состояния (state)**

# Материалы

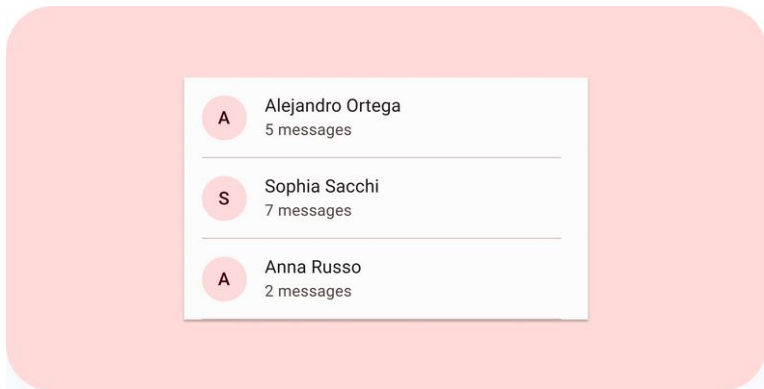


[github.com/adamxrvn/hse-lyceum-android-course](https://github.com/adamxrvn/hse-lyceum-android-course)

# Что такое состояние (state)?

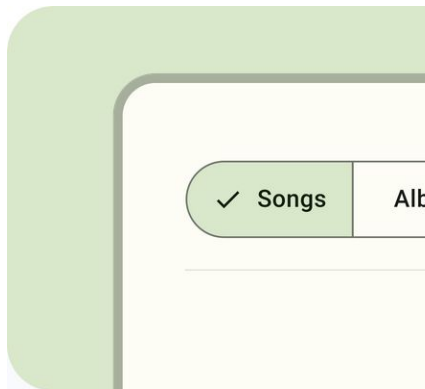


A login form with two input fields. The first field is labeled "Username" and contains the text "Sam". The second field is labeled "Password" and is empty. Both fields have a small 'x' icon in the top right corner, indicating a clear or delete function.

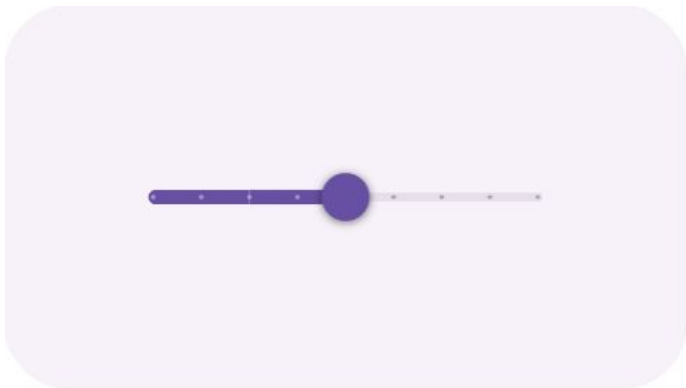


A list of messages displayed in a white box with a pink background. Each message entry consists of a circular avatar icon, a name, and a message count.

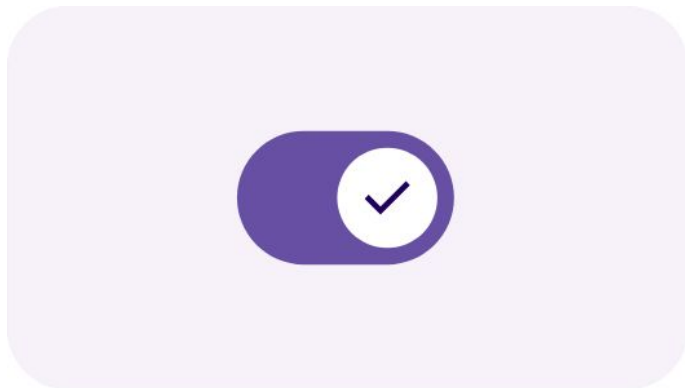
Avatar	Name	Message Count
A	Alejandro Ortega	5 messages
S	Sophia Sacchi	7 messages
A	Anna Russo	2 messages



A UI element for song selection. It features a green button with a checkmark and the text "Songs". To its right, a portion of another button with the text "Alb" is visible.



A horizontal slider control. It consists of a horizontal line with a central circular knob and small tick marks along the line.



A toggle switch control. It features a dark purple oval with a white circular knob in the center. The knob contains a white checkmark, indicating that the toggle is turned on.

# Как создать состояние?

Для одного объекта:

```
val myState = remember { mutableStateOf("Initial Value") }
```

—

`mutableStateOf()` – создаем изменяемое состояние; в скобках  
изначальное значение

`remember` – создаем объект только один раз и получаем этот  
экземпляр во всех последующих перезапусках

# Как создать состояние?

Для списка:

```
val myList = remember { mutableStateListOf<Int>() }
```

—

*mutableStateListOf()* – создаем изменяемое состояние; в скобках  
изначальное значение

*remember* – создаем объект только один раз и получаем этот  
экземпляр во всех последующих перезапусках

# Когда мы теряем состояние?

**Изменение  
конфигурации**

**Недостаток  
ресурсов**

**Закрытие  
приложения**

# Способы сохранить состояние?

**Saved State APIs**

rememberSaveable

**ViewModel**

Рассмотрим на занятии  
об архитектуре

**Persistent Storage**

Рассмотрим на занятии о  
БД (Room)

# Saved State APIs — rememberSaveable

```
var count by rememberSaveable {  
    mutableStateOf(0)  
}
```

—

*mutableStateOf* – создаем изменяемое состояние

*rememberSaveable* – remember & сохранять состояние



# Что делаем сегодня?

