

State management in Flutter

ทำความรู้จักกับวิธีการจัดการ state แบบต่าง



About this section

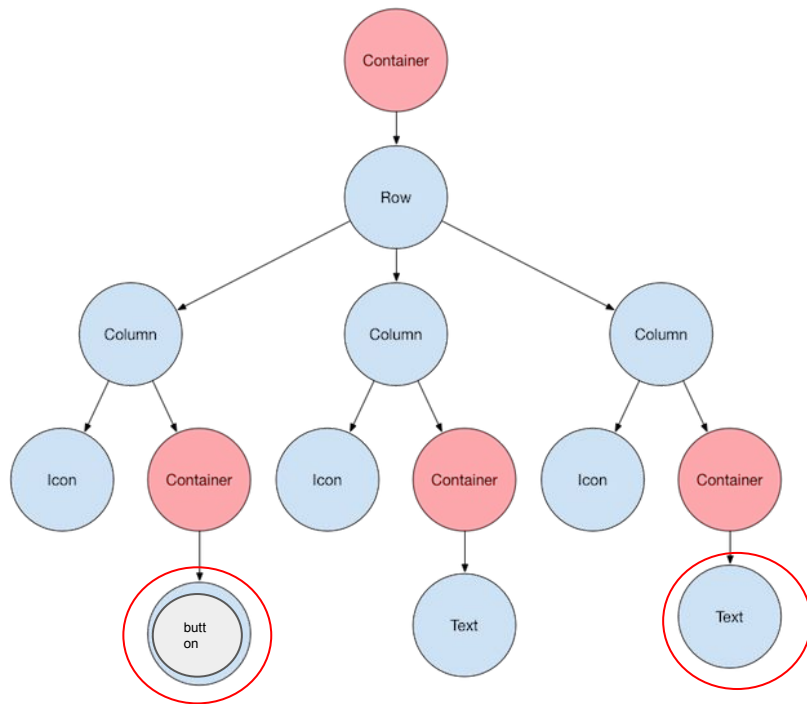
- Why do we need state management
- BloC pattern
- Provider pattern
- MobX pattern
- flutter_redux
- Flutter Modular



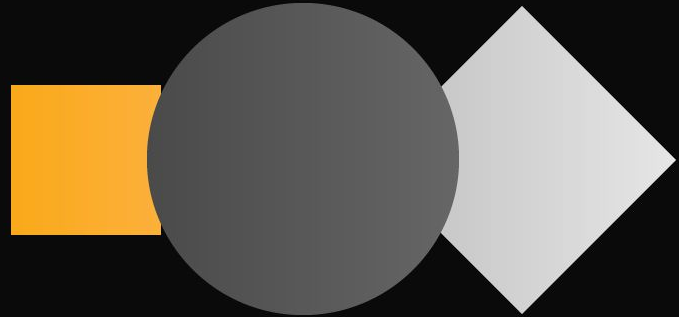
Why do we need state management?



State in Flutter

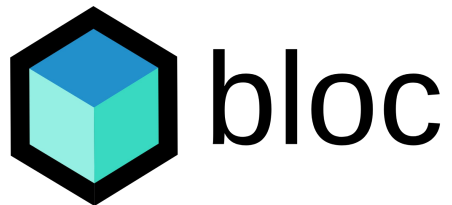


BLoC





BLoC



Simple

เข้าใจง่าย โครงสร้างไม่ซับซ้อน

Powerful

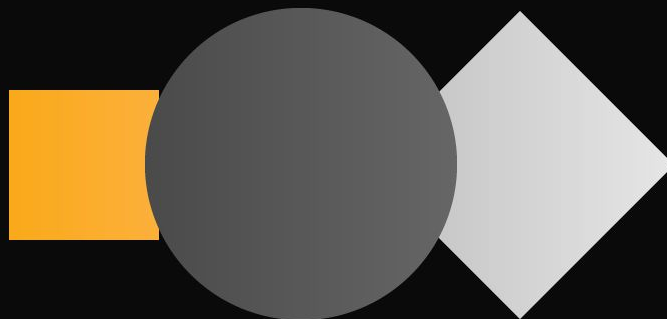
สามารถสร้างแอปที่ซับซ้อนได้ ด้วยการแบ่ง
เป็น component เล็กๆ

Testable

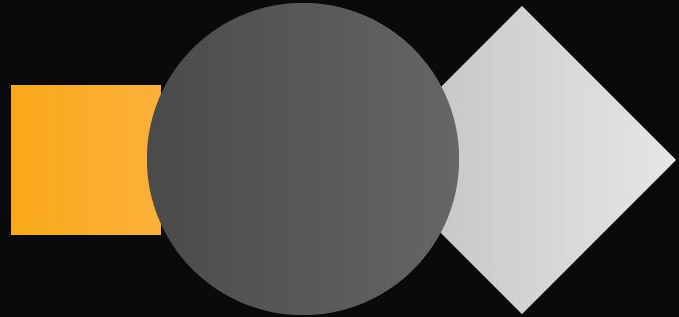
สามารถเขียนเทสได้ในทุกจุด presentation
และ business ไม่ปนกัน



Example



Provider





Provider

Upgraded InheritedWidgets

เป็น wrapper ของ inheritedwidgets

Less boilerplate code

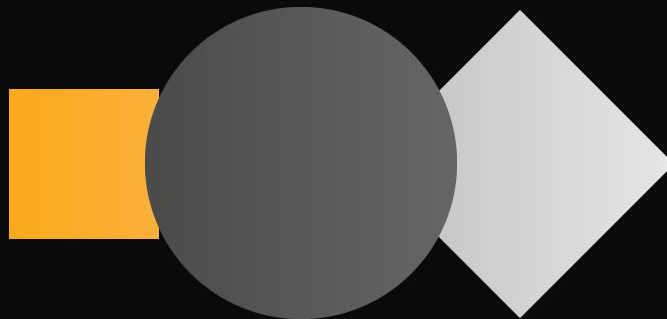
จำนวนโค้ดที่เป็น boilerplate น้อยลง เขียนง่ายขึ้น

More low-level

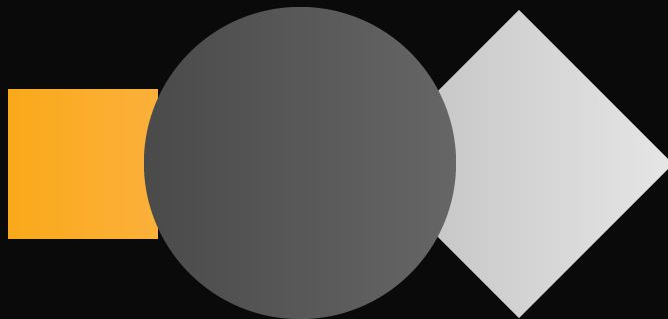
State management บางตัวก็นำ provider ไปใช้ด้านใน เช่น bloc



Example



MobX





MobX

Reactive programming

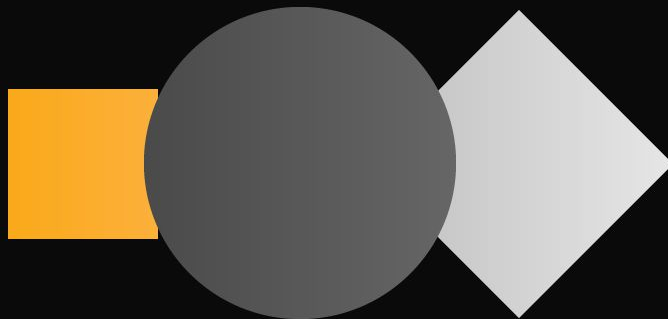
เป็นการใช้ state management แบบ reactive

Shared concept with other MobX

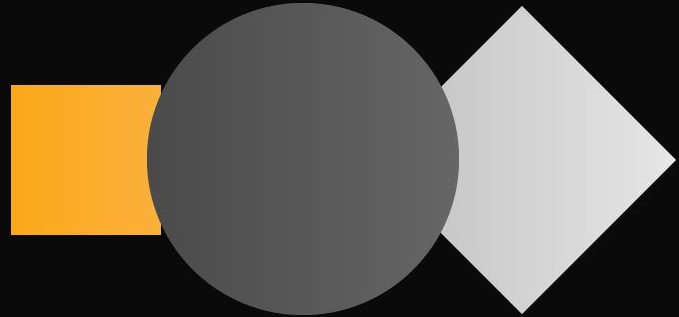
ใช้ concept เดียวกับการเขียน mobx ใน platform อื่นๆ เช่น web



Example



Flutter Redux





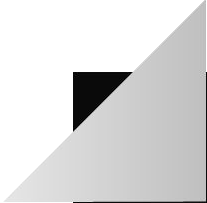
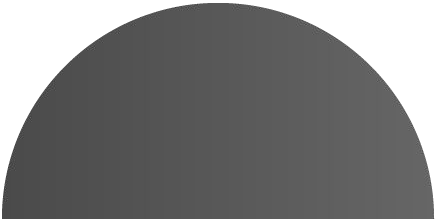
Flutter Redux

Suitable for complex projects

เหมาะสำหรับ app ที่มีสเกลใหญ่ มีโครงสร้างที่ซับซ้อน ไม่สามารถใช้งาน state ตรงๆได้

Same concept with Redux

ใช้ concept เดียวกับ redux ของ platform อื่น



Example

