



## 2. Интерфейс приложения 1/2

Кроссплатформенная разработка мобильных приложений на Flutter

Дмитрий Тараев, разработчик

# В общем

**Всё виджеты**

**Метод `build(BuildContext context)` возвращает `Widget`**

# Важные концепты

**Stateless Widget**

**Stateful Widget**

**Inherited Widget**

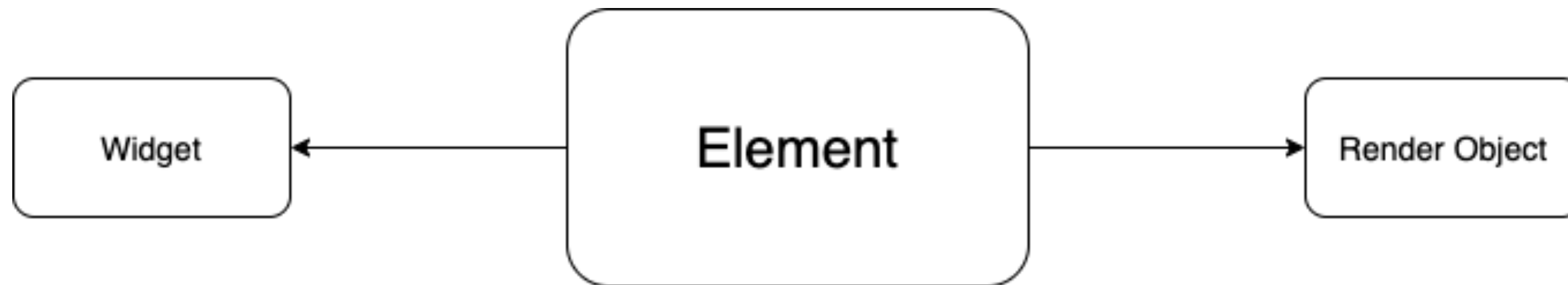
**Build Context**

# Как устроено “под капотом”

**Widget - immutable (эскиз, только описание)**

**Element - реализация виджета в конкретном месте дерева**

**Render Object - размер, расположение и отрисовка**



То есть:

конфигурация -> управление элементами -> отрисовка

# Принципы отрисовки UI

Максимально переиспользовать созданное при обновлении интерфейса

Виджеты - immutable (заменяются), но остальное переиспользуется (обновляются поля)

Если тот же runtimeType, то вызывается updateRenderObject, а не create

# Inherited Widget

- Для передачи данных от родителя к потомку**
- › Не наследование, а иерархия виджетов
- › Inherited - потому что его “наследуют”  
(то есть можно в “наследниках” обратиться к нему)
- › Реализация `MyInheritedData.of(context)`
- › `override updateShouldNotify`

# Build Context

**Чтобы widget знал, где он находится в дереве виджетов**

**Something.of(context) - возвратит ближайший Something из дерева**

# Пример (с ошибкой)

```
class MyHomePage extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      body: Center(  
        child: MaterialButton(  
          child: Text(  
            "SnackBar",  
            style: TextStyle(color: Colors.white),  
          ),  
          color: Colors.red,  
          onPressed: () => Scaffold.of(context).showSnackBar(  
            SnackBar(  
              content: Text("Oh! You clicked me!"),  
            ),  
          ),  
        ),  
      ),  
    );  
  }  
}
```

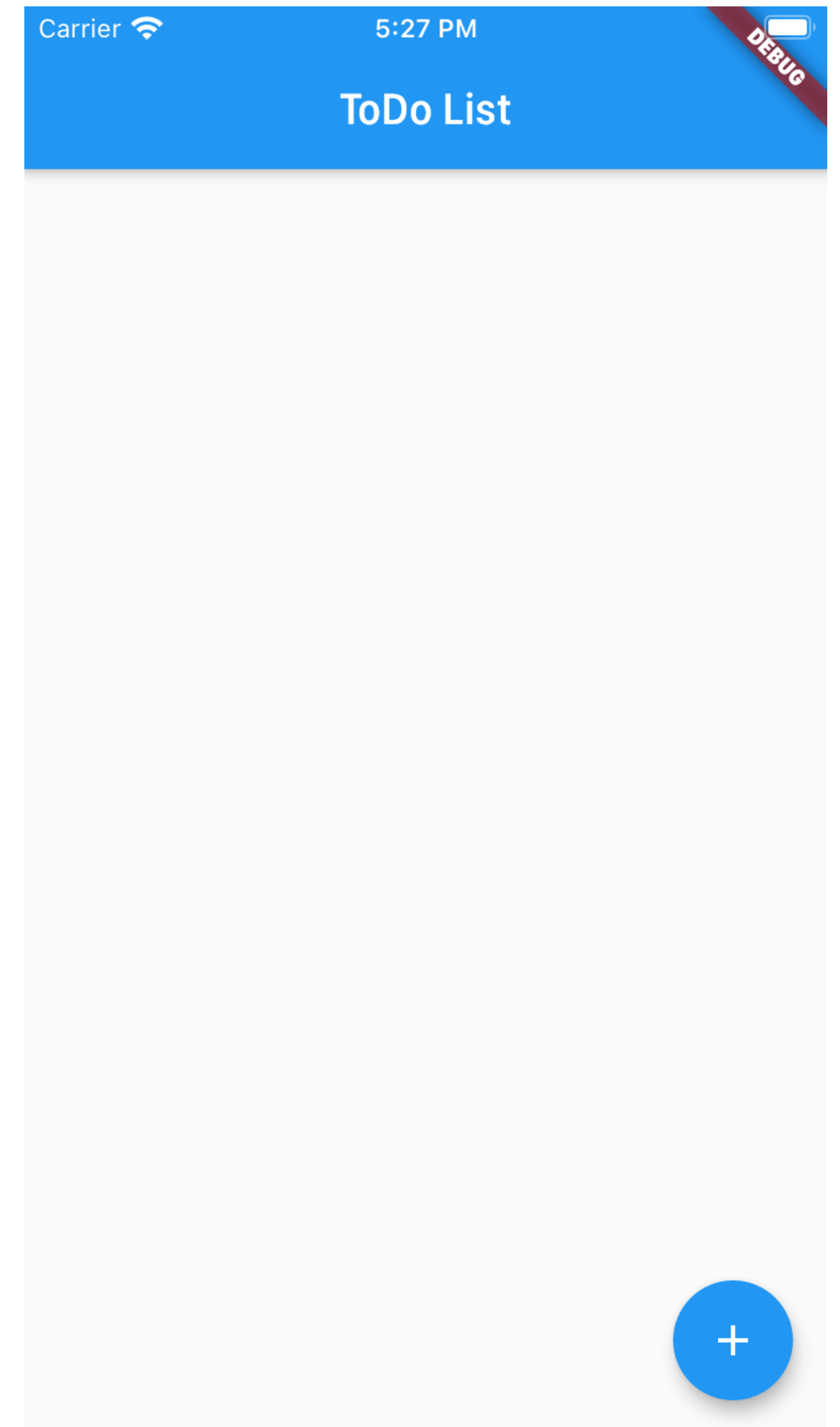


# Пример (правильный)

```
class MyHomePage extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      // The `Builder` widget will provide a `builder` method with its own `BuildContext` so that we can access any ancestor above it with this new  
      context  
      body: Builder(  
        builder: (secondContext) => Center(  
          child: MaterialButton(  
            child: Text(  
              "SnackBar",  
              style: TextStyle(color: Colors.white),  
            ),  
            color: Colors.red,  
            // Using the `Builder` `BuildContext`, named `secondContext` we can reach the `Scaffold` ancestor.  
            onPressed: () => Scaffold.of(secondContext).showSnackBar(  
              SnackBar(  
                content: Text("Oh! You clicked me!"),  
                ),),),),),),),),);}}
```

# Пример (Todo List)

- дополнение  
(с Inherited Widgets)



**Я**ндекс Такси

**Спасибо**

**Дмитрий Тараев**

Разработчик



dtaraev@yandex-team.ru



@dmitryta