

# Flutter: Stream、StreamController、StreamBuilder 简单使用教程

ImWiki

Stream 并不是 Flutter 专属库，而是 Dart 的库，和 Future 一样都是非常重要的异步编程方式，[Rx Dart](#)、[Bloc](#)、[flutter\\_bloc](#) 都是基于 Stream 开发。Stream 的思想是基于管道（pipe）和 生产者消费者模式。

## 案例一：通过 Stream 实现每秒钟局部更新数据

通常我们在开发 Flutter 页面，数据发生更新，都是直接通过 setState 方式对整个页面进行更新。如果页面某些数据需要每秒都更新，对整个页面都刷新是很影响页面性能，有必要局部刷新。

以下案例，实现每秒钟显示当前的时间，甚至连 StatefulWidget 都没有使用就可以实现数据更新。

```
void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(
        primarySwatch: Colors.blue,
```

```

        visualDensity:
VisualDensity.adaptivePlatformDensity,
    ),
    home: StreamDemo(),
  );
}
}

class StreamDemo extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text('StreamDemo')),
      body: Center(
        child: StreamBuilder<String>(
          initialData: "",
          stream: Stream.periodic(Duration(seconds:
1), (value) {
            return DateTime.now().toIso8601String();
          }),
          builder: (context, AsyncSnapshot<String>
snapshot) {
            return Text(
              '${snapshot.data}',
              style: TextStyle(fontSize: 24.0),
            );
          },
        ),
      );
    }
}

```

```
}
```

## 案例二：通过按钮点击实现局部更新

```
class StreamDemo2 extends StatefulWidget {  
  @override  
  _StreamDemo2State createState() =>  
  _StreamDemo2State();  
}  
  
class _StreamDemo2State extends State<StreamDemo2> {  
  StreamController<int> _streamController;  
  var _counter = 0;  
  
  @override  
  void initState() {  
    _streamController = StreamController<int>();  
    super.initState();  
  }  
  
  @override  
  void dispose() {  
    _streamController.close();  
    super.dispose();  
  }  
  
  Stream<int> counter() {  
    return _streamController.stream;  
  }  
  
  @override
```

```

Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(title: Text('StreamDemo2')),
    body: Center(
      child: StreamBuilder<int>(
        initialData: 0,
        stream: counter(),
        builder: (context, AsyncSnapshot<int>
snapshot) {
          return Text(
            '${snapshot.data}',
            style: TextStyle(fontSize: 24.0),
          );
        },
      ),
    floatingActionButton: FloatingActionButton(
      child: Icon(Icons.add),
      onPressed: () {
        _streamController.add(++_counter);
      },
    ),
  );
}
}

```

虽然 Stream 可以实现数据局部的刷新，但是 Stream 属于比较底层的类，如果要想实现非常复杂的页面开发并实现逻辑分离，还是建议使用 [Bloc](#)，封装比较完善，降低开发成本。