

6.AddTo

在之前，我们接触了 AddTo 这个 API，但是还没有进行一个正式介绍。那么我们在这一节就先了解一下这个 AddTo API。

其实字面意思上理解为很简单，就是 添加到。

添加到哪里呢？

其实就是 Unity 的 GameObject 或者 MonoBehaviour。

为什么要添加到 GameObject 或者 MonoBehaviour 呢？

是因为，GameObject 和 MonoBehaviour 可以获取到 OnDestroy 事件。也就是 GameObject 或者 MonoBehaviour 的销毁事件。

那么用这个销毁事件干嘛呢？

答案是用来 进行与 UniRx 进行销毁事件的绑定，也就是当 GameObject 或者 MonoBehaviour 被销毁时，同样去销毁正在进行的 UniRx 任务。

这就是 AddTo API 的作用。

其实用起来很简单，代码如下：

```
Observable.Timer(TimeSpan.FromSeconds(1.0f))
    .Subscribe()
    .AddTo(this); // Or gameObject
```

这样，当 this 所在的 GameObject 销毁时，这个 Timer 就会被销毁。

为什么会这样？

本质上，AddTo 是一个 静态扩展关键字，他对 IDisposable 进行了扩展。

只要任何实现了 IDisposable 的接口，都可以使用 AddTo API，不管是不是 UniRx 的 API。

当 GameObject 销毁时，就会调用 IDisposable 的 OnDispose 这个方法。

很容易理解。

AddTo 能做什么？

有了 AddTo，在开启 Observable.EveryUpdate 时调用 当前脚本的方法，则不会造成引用异常等错误，它使得 UniRx 的使用更加安全。

今天的内容就这些。