

# 8.SelectMany

在之前我们学习了 Select 操作符，今天我们学习 SelectMany。

## LINQ SelectMany 简介

字如其意，就是 Select 许多次的意思。

微软官方的定义是: 将序列的每个元素投影到 IEnumerable<T> 并将结果序列合并为一个序列。

## LINQ SelectMany 示例代码

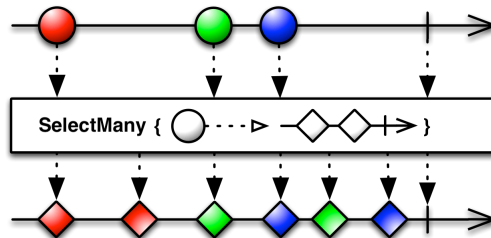
```
/* *****  
 * http://sikiedu.com liangxie  
 * ***** */  
  
using System.Collections.Generic;  
using System.Linq;  
using UnityEngine;  
  
namespace UniRxLesson  
{  
    public class LINQSelectManyExample : MonoBehaviour  
    {  
        class Student  
        {  
            public string Name;  
  
            public int Age;  
        }  
  
        private void Start()  
        {  
            var students = new List<Student>()  
            {  
                new Student() {Name = "张三", Age = 50},  
                new Student() {Name = "张三", Age = 45},  
                new Student() {Name = "李四", Age = 50}  
            };  
  
            var singleChars = students.SelectMany(student => student.Name + ":" +  
student.Age);  
  
            foreach (var singleChar in singleChars)  
            {  
                Debug.Log(singleChar);  
            }  
        }  
    }  
}
```

```

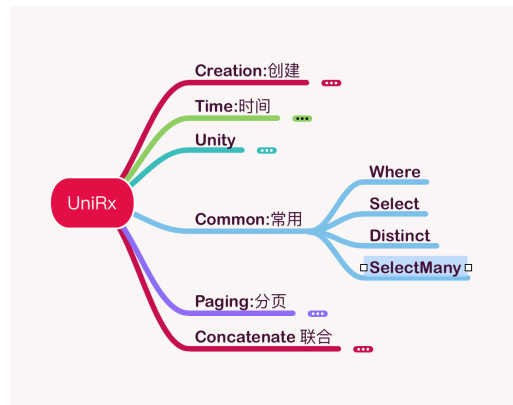
    }
}

```

## UniRx SelectMany 示意图



## UniRx SelectMany 在知识地图中的位置



SelectMany 是比较常用的。

## UniRx SelectMany 代码示例

SelectMany 理解起来会比较晦涩，一般在 UniRx 中，主要是完成 Coroutine 的顺序执行功能。

代码如下：

```

/*****
* http://sikiedu.com liangxie

```

```

*****/

using System.Collections;
using UniRx;
using UnityEngine;

namespace UniRxLesson
{
    public class UniRxSelectManyExample : MonoBehaviour
    {
        IEnumerator A()
        {
            yield return new WaitForSeconds(1.0f);
            Debug.Log("A");
        }

        IEnumerator B()
        {
            yield return new WaitForSeconds(2.0f);
            Debug.Log("B");
        }

        IEnumerator C()
        {
            yield return new WaitForSeconds(3.0f);
            Debug.Log("C");
        }

        private void Start()
        {
            var streamA = Observable.FromCoroutine(A);
            var streamB = Observable.FromCoroutine(B);
            var streamC = Observable.FromCoroutine(C);

            streamA.SelectMany(streamB.SelectMany(streamC))
                .Subscribe(_ => Debug.Log("Hello"));
        }
    }
}

```

输出结果为:

```

A
B
C
Hello

```

今天的内容就这些。