

11.WhenAll

LINQ All 简介

确定序列中的所有元素是否都满足条件。

LINQ All 示例代码

```

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

using System.Linq;
using UnityEngine;

namespace UniRxLesson
{
    public class LINQAllExample : MonoBehaviour
    {
        private class Pet
        {
            public string Name { get; set; }
            public int Age { get; set; }
        }

        private void Start()
        {
            Pet[] pets =
            {
                new Pet {Name = "Barley", Age = 10},
                new Pet {Name = "Boots", Age = 4},
                new Pet {Name = "Whiskers", Age = 6}
            };

            var allStartWithB = pets.All(pet =>
                pet.Name.StartsWith("B"));

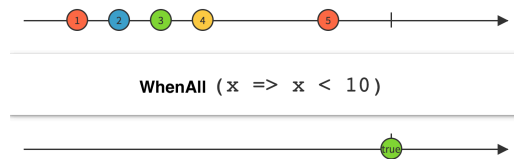
            Debug.LogFormat(
                "{0} pet names start with 'B'.",
                allStartWithB ? "All" : "Not all");
        }
    }
}
```

输出结果为:

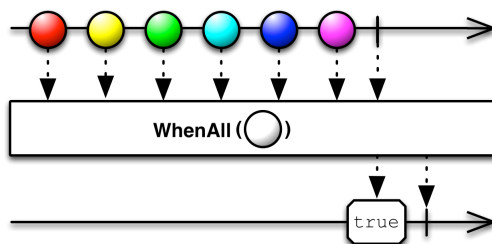
Not all pet names start with 'B'.

UniRx WhenAll 示意图

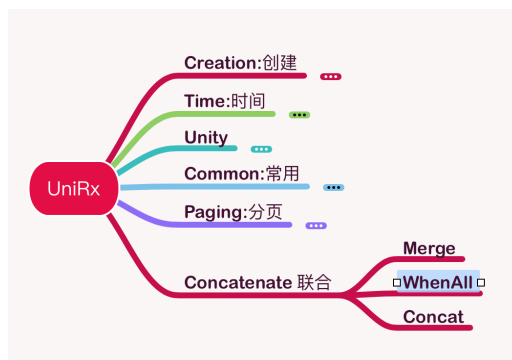
判定是否Observable发射的所有数据都满足某个条件



传递一个谓词函数给 **WhenAll** 操作符，这个函数接受原始 Observable 发射的数据，根据计算返回一个布尔值。**WhenAll** 返回一个只发射一个单个布尔值的 Observable，如果原始 Observable 正常终止并且每一项数据都满足条件，就返回 `true`；如果原始 Observable 的任意一项数据不满足条件就返回 `False`。



UniRx WhenAll 在知识地图中的位置



UniRx WhenAll 示例代码

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

using System.Collections;
using UniRx;
using UnityEngine;

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

        IEnumerator B()
        {
            yield return new WaitForSeconds(1.0f);
            Debug.Log("B");
        }

        IEnumerator C()
        {
            yield return new WaitForSeconds(1.0f);
            Debug.Log("C");
        }

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

            Observable.WhenAll(streamA, streamB, streamC)
                .Subscribe(_ => { Debug.Log("Completed"); });
        }
    }
}
```

输出结果为(A、B、C 输出顺序不一定):

```
B
C
A
Completed
```

今天的内容就这些