

## aardio 范例: 在 aardio 中使用 .NET 事件 (event)

```
//在 aardio 中使用 .NET 事件 (event )
import console;
import dotNet;

var compiler = dotNet.createCompiler("C#");
compiler.Source = /*****
namespace CSharpLibrary
{
    public class Object
    {
        //定义一个委托类型
        public delegate int TestDelegateType(string str,int a);

        //定义一个事件
        public event TestDelegateType onTestEvent;

        public int Test( )
        {
            return onTestEvent("你好",123);
        }
    }
}
*****/

compiler.import("CSharpLibrary"); //编译 C# 代码并导入名字空间
var netObj = CSharpLibrary.Object();//创建 .Net 对象

/*
.NET 中的 event 实际上是在委托 (delegate) 上再做了层封装。
在 aardio 中对 event 赋值总是追加而不是覆盖 (对delegate 赋值则总是覆盖而不是追加)
*/
netObj.onTestEvent = function(str,a){
    //无论.NET 回调是否在同一线程,被 .NET 回调的 aardio 函数总是在原调用线程执行 (不必考虑多线程规则与同步)。
    console.log("1、aardio 函数被 C# 调用了,参数:",a,b)
    return 2; //如果不返回委托指定类型的返回值会导致报错,委托返回值类型为 void 时这里可以不返回值。
}

//也可以这样追加,追加的用法与委托是一样的
var delgate = dotNet.delegate.combine(
    netObj,"onTestEvent",function(a,b){
        console.log("2、aardio 函数被 C# 调用了,参数:",a,b)
        return 2;
    }
)

//上面的函数会返回追加的委托对象,可以传入下面的函数用于移除事件
//dotNet.delegate.remove(netObj,"onTestEvent",delgate);

//调用 .NET 函数触发事件
netObj.Test();

console.pause();
```