

推荐一个小工具:LINQ to JavaScript

如果你用过 c#的 linq 语句,你会发现这东西确实能让我们从烦琐的技术细节中解脱出来,而且能让我们更加关注项目的业务逻辑.暂且不管 linq 的缺点,linq 语句的出现确实给我们带来了实惠.而 javascript 中的数组操作是否也能够实现 linq 似的查询呢?这也是本文推荐的一个开源小项目"LINQ to JavaScript",简称"JSLINQ".

先看其介绍:"LINQ to JavaScript (JSLINQ) is an implementation of LINQ to Objects implemented in JavaScript. It is built using a set of extension methods built on top of the JavaScript Array object. If you are using an Array, you can use LINQ to JavaScript."

项目地址:<http://www.codeplex.com/JSLINQ>.

```
var myList = [  
    {FirstName:"Chris",LastName:"Pearson"},  
    {FirstName:"Kate",LastName:"Johnson"},  
    {FirstName:"Josh",LastName:"Sutherland"},  
    {FirstName:"John",LastName:"Ronald"},  
    {FirstName:"Steve",LastName:"Pinkerton"}  
];
```

```
var exampleArray = From(myList).  
    Where("item.FirstName == 'Chris'").  
    OrderBy("item.FirstName").  
    Select("item.FirstName");
```

看到 From,Where 这些关键字,我们已经感觉到这是在使用 Linq 了.

看一下作者的源码,我发现作者其实并没有使用很复杂的技巧,而是把平时一些常用操作使用 prototype 给 Array 增加了一些扩展方法(可参见["C#中的 this 扩展方法与 javascript 中的 prototype 方法"](#)).这样我们在应用本扩展方法之后,在我们项目中任何需要进行 Array 查询的时候,就可以用这种直观的 linq 语句来查询就可以了.这样做的好处不仅仅是我们在查询数组时候更加方便,而且为后期的维护也降低了难度,因为这样简单的查询语句代替了原来繁琐的表达.

JSLINQ 对 Array 的扩展还真的不错,把 Dotnet 中的 linq 基本功能都已经实现了,大家先看看作者为 Array 增加的 19 个扩展方法吧.

```
Array.prototype.Where = function(clause){
};
Array.prototype.Select = function(clause){
};
Array.prototype.OrderBy = function(clause){
};
Array.prototype.OrderByDescending = function(clause){
};
Array.prototype.SelectMany = function(clause){
};
Array.prototype.Count = function(clause){
};
Array.prototype.Distinct = function(clause){
};
Array.prototype.Any = function(clause){
};
Array.prototype.All = function(clause){
};
Array.prototype.Reverse = function(){
};
Array.prototype.First = function(clause){
};
Array.prototype.Last = function(clause){
};
Array.prototype.ElementAt = function(index){
};
Array.prototype.Concat = function(array){
};
Array.prototype.Intersect = function(secondArray, clause){
};
Array.prototype.DefaultIfEmpty = function(defaultValue){
};
Array.prototype.ElementAtOrDefault = function(index, defaultValue){
};
Array.prototype.FirstOrDefault = function(defaultValue){
};
Array.prototype.LastOrDefault = function(defaultValue){
};
};
```

reactigod@gmail.com

严重推荐大家去看看作者的源码,从中你会学到很多有用的东西.