

基于 RESTful 标准的 Web Api - 文章 - 伯乐在线



微软的web api是在vs2012上的mvc4项目绑定发行的，它提出的web api是完全基于RESTful标准的，完全不同于之前的（同是SOAP协议的）wcf和WebService，它是简单，代码可读性强的，上手快的，如果要拿它和web服务相比，我会说，它的接口更标准，更清晰，没有混乱的方法名称，有的只有几种标准的请求，如get, post, put, delete等，它们分别对应的几个操作，下面讲一下：

GET：生到数据列表（默认），或者得到一条实体数据

POST：添加服务端添加一条记录，记录实体为Form对象

PUT：添加或修改服务端的一条记录，记录实体的Form对象，记录主键以GET方式进行传输

DELETE：删除 服务端的一条记录

注意上面公开的API接口都是在XMLHttpRequest情况下调用的，当然你可以使用jquery的ajax组件来完成这个请求调用，它的代码更加面向对象，下面举例说明一下

这是页面的HTML代码部分，实现最基本的CRUD操作

C#

```
<fieldset>
    <legend>测试Web Api
    </legend>
    <a href="javascript:add()">添加(post)</a>
    <a href="javascript:update(1)">更新(put)</a>
    <a href="javascript:deletes(1)">删除(delete)</a>
    <a href="/api/test">列表(Get)</a>
    <a href="/api/test/1">实体(Get)</a>
</fieldset>
<script>
    function add() {
        $.ajax({
            url      : "/api/Test/",
            type     : "POST",
            data     :
{"UserID":4,"UserName":"test","UserEmail":"Parry@cnblogs.com"},
            success: function (data) { alert(JSON.stringify(data)); }
        });
    };
</script>
```

```

    }
    //更新
    function update(id) {
        $.ajax({
            url      : "/api/Test?id="+id,
            type     : "Put",
            data     :
{"UserID":1,"UserName":"moditest","UserEmail":"Parry@cnblogs.com"},
            success: function (data) { alert(JSON.stringify(data)); }
        });
    }
    function deletes(id) {
        $.ajax({
            url      : "/api/Test/1",
            type     : "DELETE",
            success: function (data) { alert(data);}
        });
    }
}
</script>C#
///
/// Test模块API
/// URI: /Api/Test
///
public class TestController : ApiController
{
    ///
    /// User Data List
    ///
    private readonly List _userList = new List
    {
        new Users {UserID = 1, UserName = "zzl", UserEmail =
"bfyxzls@sina.com"},
        new Users {UserID = 2, UserName = "Spiderman", UserEmail =
"Spiderman@cnblogs.com"},
        new Users {UserID = 3, UserName = "Batman", UserEmail =
"Batman@cnblogs.com"}
    };
    ///
    /// 得到列表对象
    ///
    ///
    public IEnumerable Get()

```

```
{
    return _userList;
}
///  
/// 得到一个实体，根据主键  
///  
///  
///  
public Users Get(int id)
{
    return _userList.FirstOrDefault(i => i.UserID == id);
}
///  
/// 添加  
///  
/// 表单对象，它是唯一的  
///  
public Users Post([FromBody] Users entity)
{
    _userList.Add(entity);
    return entity;
}
///  
/// 更新  
///  
/// 主键  
/// 表单对象，它是唯一的  
///  
public Users Put(int id, [FromBody]Users entity)
{
    var user = _userList.FirstOrDefault(i => i.UserID == id);
    if (user != null)
    {
        user.UserName = entity.UserName;
        user.UserEmail = entity.UserEmail;
    }
    return user;
}
///  
/// 删除  
///  
/// 主键
```

```
    ///
    public void Delete(int id)
    {
        _userList.Remove(_userList.FirstOrDefault(i=>i.UserID==id));
    }
}
```

下面我们看一下各个操作调用的结果：

添加操作（POST）

```
{"UserID":4,"UserName":"test","UserEmail":"Parry@cnblogs.com"}
```

确定

POST http://localhost:7543/api/Test/ 200 OK 8ms

头信息 Post 响应 JSON Cookies

参数 application/x-www-form-urlencoded

UserEmail Parry@cnblogs.com
UserID 4
UserName test

源代码
UserID=4&UserName=test&UserEmail=Parry%40cnblogs.com

更新操作（PUT）

```
{"UserID":1,"UserName":"moditest","UserEmail":"Parry@cnblogs.com"}
```

确定

PUT http://localhost:7543/api/Test?id=1 200 OK 12ms

参数 头信息 Put 响应 JSON Cookies

参数 application/x-www-form-urlencoded

UserEmail Parry@cnblogs.com
UserID 1
UserName moditest

源代码
UserID=1&UserName=moditest&UserEmail=Parry%40cnblogs.com

最后，再强调一下，web api使用RESTful架构的，不像传统的（同是SOAP的）RPC方式的wcf和web

Service，它强调的是接口模块的概念，每个模块都是独立的，每个模块的接口方法都是统一的和单一的，即CRUD方法组成。

加入伯乐在线专栏作者。扩大知名度，还能得赞赏！详见《[招募专栏作者](#)》

1 赞 1 收藏 [评论](#)



合作联系

Email: bd@jobbole.com

QQ: 2302462408 （加好友请注明来意）

更多频道

[小组](#) - 好的话题、有启发的回复、值得信赖的圈子

[头条](#) - 分享和发现有价值的内容与观点

[相亲](#) - 为IT单身男女服务的征婚传播平台

[资源](#) - 优秀的工具资源导航

[翻译](#) - 翻译传播优秀的外文文章

[文章](#) - 国内外的精选文章

[设计](#) - UI, 网页, 交互和用户体验

[iOS](#) - 专注iOS技术分享

[安卓](#) - 专注Android技术分享

[前端](#) - JavaScript, HTML5, CSS

[Java](#) - 专注Java技术分享

[Python](#) - 专注Python技术分享