IHttpActionResult类型是WebApi里面非常重要的一种返回值类型。下面博主就根据平时在项目里面使用最多的几种方式来讲解下这种类型的返回值的一些用法。

## **1、Json<T>(T content)**

public IHttpActionResult GetOrder()

{

var lstRes = new List<ORDER>();

//实际项目中，通过后台取到集合赋值给lstRes变量。这里只是测试。

lstRes.Add(new ORDER() { ID = "aaaa", NO = "111", NAME = "111", DESC = "1111" });

lstRes.Add(new ORDER() { ID = "bbbb", NO = "222", NAME = "222", DESC = "2222" });

return Json<List<ORDER>>(lstRes);

}

## **2、Ok()、 Ok<T>(T content)**

public IHttpActionResult GetOKResult()

{

return Ok();

}

如果返回Ok()，就表示不向客户端返回任何信息，只告诉客户端请求成功。

除了Ok()之外，还有另外一个重载Ok<T>(T content)。

public IHttpActionResult GetOKResult(string name)

{

return Ok<string>(name);

}

这种用法和Json<T>(T content)比较类似

## **3、NotFound()**

当需要向客户端返回找不到记录时，有时需要用到NotFound()方法。

public IHttpActionResult GetNotFoundResult(string id)

{

var lstRes = new List<ORDER>();

//实际项目中，通过后台取到集合赋值给lstRes变量。这里只是测试。

lstRes.Add(new ORDER() { ID = "aaaa", NO = "111", NAME = "111", DESC = "1111" });

lstRes.Add(new ORDER() { ID = "bbbb", NO = "222", NAME = "222", DESC = "2222" });

var oFind = lstRes.FirstOrDefault(x => x.ID == id) ;

if (oFind == null)

{

return NotFound();

}

else

{

return Json<ORDER>(oFind);

}

}

NotFound()方法会返回一个404的错误到客户端。

## **4、其他**

其他还有一些方法，都有它特定的用途。在此贴出来。

### **4.1、Content<T>(HttpStatusCode statusCode, T value)**

[HttpGet]

public IHttpActionResult GetContentResult()

{

return Content<string>(HttpStatusCode.OK, "OK");

}

向客户端返回值和http状态码。

### **4.2、BadRequest()**

[IMG_256](https://www.cnblogs.com/landeanfen/p/javascript:void(0);)

[HttpGet]

public IHttpActionResult GetBadRequest(ORDER order)

{

if (string.IsNullOrEmpty(order.ID))

return BadRequest();

return Ok();

}

[IMG_257](https://www.cnblogs.com/landeanfen/p/javascript:void(0);)

向客户端返回400的http错误。

### **4.3、Redirect(string location)**

[HttpGet]

public IHttpActionResult RedirectResult()

{

return Redirect("http://localhost:21528/api/Order/GetContentResult");

}

将请求重定向到其他地方。

## **5、自定义IHttpActionResult接口的实现**

上面介绍了一些系统内置的常用的实现IHttpActionResult接口的方法。如果我们需要自定义IHttpActionResult的返回呢？

在介绍之前，我们有必要先来看看IHttpActionResult类型的定义，将IHttpActionResult转到定义可以看到：

[IMG_256](https://www.cnblogs.com/landeanfen/p/javascript:void(0);)

namespace System.Web.Http

{

// 摘要:

// Defines a command that asynchronously creates an System.Net.Http.HttpResponseMessage.

public interface IHttpActionResult

{

// 摘要:

// Creates an System.Net.Http.HttpResponseMessage asynchronously.

//

// 参数:

// cancellationToken:

// The token to monitor for cancellation requests.

//

// 返回结果:

// A task that, when completed, contains the System.Net.Http.HttpResponseMessage.

Task<System.Net.Http.HttpResponseMessage> ExecuteAsync(CancellationToken cancellationToken);

}

}

[IMG_257](https://www.cnblogs.com/landeanfen/p/javascript:void(0);)

这个接口包含唯一的一个方法ExecuteAsync()，此方法将以异步方式创建一个**HttpResponseMessage**实例返回给客户端。

有了这个作为基础，下面，我们自定义一个bootstrapTable服务端分页的子类去展示自定义IHttpActionResult的用法。

首先，自定义一个实现类

[IMG_258](https://www.cnblogs.com/landeanfen/p/javascript:void(0);)

　　public class PageResult : IHttpActionResult

{

object \_value;

HttpRequestMessage \_request;

public PageResult(object value, HttpRequestMessage request)

{

\_value = value;

\_request = request;

}

public Task<HttpResponseMessage> ExecuteAsync(System.Threading.CancellationToken cancellationToken)

{

var response = new HttpResponseMessage()

{

Content = new ObjectContent(typeof(object), \_value, new JsonMediaTypeFormatter()),

RequestMessage = \_request

};

return Task.FromResult(response);

}

}

[IMG_259](https://www.cnblogs.com/landeanfen/p/javascript:void(0);)

然后，在API接口里面返回PageResult对象

[IMG_260](https://www.cnblogs.com/landeanfen/p/javascript:void(0);)

　　　　[HttpGet]

public IHttpActionResult GetPageRow(int limit, int offset)

{

var lstRes = new List<ORDER>();

//实际项目中，通过后台取到集合赋值给lstRes变量。这里只是测试。

lstRes.Add(new ORDER() { ID = "aaaa", NO = "111", NAME = "111", DESC = "1111" });

lstRes.Add(new ORDER() { ID = "bbbb", NO = "222", NAME = "222", DESC = "2222" });

var oData = new { total = lstRes.Count, rows = lstRes.Skip(offset).Take(limit).ToList() };

return new PageResult(oData, Request);

}

[IMG_261](https://www.cnblogs.com/landeanfen/p/javascript:void(0);)

最好，ajax调用

[IMG_262](https://www.cnblogs.com/landeanfen/p/javascript:void(0);)

$(function () {

$.ajax({

type: 'get',

url: 'http://localhost:21528/api/Order/GetPageRow',

data: { limit:1,offset:1},

success: function (data, status) {

alert(data);

}

});

});

[IMG_263](https://www.cnblogs.com/landeanfen/p/javascript:void(0);)