1. 结构的字段也可不初始化结构就赋值

当分配一个对象，会收到一个包含值在内存中位置的指向的引用，另外，对象数据保存在内存堆中。 object.ReferenceEquals 方法提供了判断两个对象是否在同一个内存空间。

**托管资源**一般是指被CLR控制的内存资源,这些资源的管理可以由CLR来控制,例如程序中分配的对象,作用域内的变量等，大部分对象都是托管资源。

**非托管资源**是CLR不能控制或者管理的部分，这些资源有很多，比如文件流，数据库的连接，系统的窗口句柄，打印机资源，需要调用Dispose方法

垃圾回收的时候：

内存移动，

地址变更，

阻塞线程，

继续运行

System.Timers.Timer

System.Threadings.Timer 注意有两个Timer

对象越老，生存期越长

回收部分快过全部回收

垃圾回收只有0，1，2代,之后还是为2代

可以使用AttributeUsage定义属性的单一用途或多用途

[AttributeUsage(AttributeTargets.All, AllowMultiple=true)]  
 public class SomethingAttribute : Attribute  
 {  
   public SomethingAttribute(String str)  
 {  
 }  
 }

public enum AttributeTargets

{

Assembly = 0x0001, Module = 0x0002, Class = 0x0004,

Struct = 0x0008, Enum = 0x0010, Constructor = 0x0020,

Method = 0x0040, Property = 0x0080, Field = 0x0100,

Event = 0x0200, Interface = 0x0400, Parameter = 0x0800,

Delegate = 0x1000,

All = Assembly │ Module │ Class │ Struct │ Enum │ Constructor │

Method │ Property │ Field │ Event │ Interface │ Parameter │

Delegate,

ClassMembers = Class │ Struct │ Enum │ Constructor │ Method │

Property │ Field │ Event │ Delegate │ Interface,

}

1. Dynamic，JObject和HttpContext.Request.Form["User[UserID]"]接收参数

$.ajax({ url: "/api/users/RegisterObjectDynamic", type: "post", data: { "User": user, "Info": info }, success: function (data) { alert(data); }, datatype: "json", contentType: 'application/json' });

//POST api/Users/register

[HttpPost]

public string RegisterObject(JObject jData)//可以来自FromBody FromUri

{

string idParam = base.HttpContext.Request.Form["User[UserID]"];

string nameParam = base.HttpContext.Request.Form["User[UserName]"];

string emailParam = base.HttpContext.Request.Form["User[UserEmail]"];

string infoParam = base.HttpContext.Request.Form["info"];

dynamic json = jData;

JObject jUser = json.User;

string info = json.Info;

var user = jUser.ToObject<Users>();

return string.Format("{0}\_{1}\_{2}\_{3}", user.UserID, user.UserName, user.UserEmail, info);

}

[HttpPost]

public string RegisterObjectDynamic(dynamic dynamicData)//可以来自FromBody FromUri

{

string idParam = base.HttpContext.Request.Form["User[UserID]"];

string nameParam = base.HttpContext.Request.Form["User[UserName]"];

string emailParam = base.HttpContext.Request.Form["User[UserEmail]"];

string infoParam = base.HttpContext.Request.Form["info"];

dynamic json = dynamicData;

JObject jUser = json.User;

string info = json.Info;

var user = jUser.ToObject<Users>();

return string.Format("{0}\_{1}\_{2}\_{3}", user.UserID, user.UserName, user.UserEmail, info);

}

1. Consul 作用：负载均衡 ，服务注册与开发，健康检查

Cmd 运行consul\_1.6.2.exe agent –dev

Nuget 安装consul

ConsulClient client = new ConsulClient(c =>

{

c.Address = new Uri("http://localhost:8500/");

c.Datacenter = "dc1";

});

string ip = configuration["ip"];

int port = int.Parse(configuration["port"]);//命令行参数必须传入

//int weight = string.IsNullOrWhiteSpace(configuration["weight"]) ? 1 : int.Parse(configuration["weight"]);//命令行参数必须传入

client.Agent.ServiceRegister(new AgentServiceRegistration()

{

ID = "service" + Guid.NewGuid(),//唯一的

Name = "ZhaoxiUserService",//组名称-Group

Address = ip,//其实应该写ip地址

Port = port,//不同实例

//Tags = new string[] { weight.ToString() },//标签

Check = new AgentServiceCheck()//配置心跳检查的

{

Interval = TimeSpan.FromSeconds(12),

HTTP = $"http://{ip}:{port}/Api/Health/Index",

Timeout = TimeSpan.FromSeconds(5),

DeregisterCriticalServiceAfter = TimeSpan.FromSeconds(5)

}

});