**.Net5+Ocelot+Consul**

实现API网关 服务注册 服务发现 负载均衡

**一、阐述网关**

简单来说，网关就是暴露给外部的请求入口。就和门卫一样，外面的人想要进来，必须要经过门卫。当然，网关并不一定是必须的，但对于业务复杂、规模庞大的项目来说，使用网关有很多无法舍弃的好处，比如可以进行统一的请求聚合来节省流量、降低耦合度，可以赋予项目熔断限流的能力提高可用性等等；明确一点，网关应该作为独立进程存在。

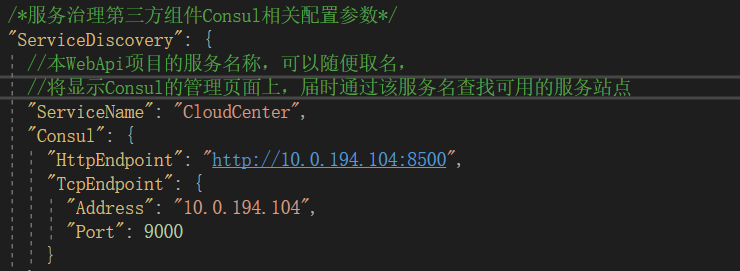
**二、具体使用**

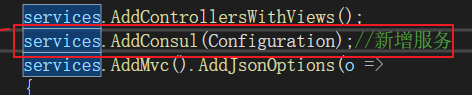
1、Ocelot(http://ocelot.readthedocs.io)是一个用.NET Core实现并且开源的API网关，它功能强大，包括了：路由、负载均衡、请求聚合、认证、鉴权、限流熔断等，这些功能只都只需要简单的配置即可完成

2、Consul(https://www.consul.io)是一个分布式，高可用、支持多数据中心的服务注册、发现、健康检查和配置共享的服务软件（https://www.tizi365.com/archives/535.html）。

3、Ocelot天生集成对Consul支持，在MySteelSoft.CloudCenter.APIGateway项目中Ocelot.json配置就可以开启ocelot+consul的组合使用，实现服务注册、服务发现、健康检查、负载均衡。

4、开发的过程中注册服务到Consul中可以使用代码进行控制，在<Middlewares>文件中有相关的注册服务代码；appsettings.json中需要配置consul的地址。（此方法不建议使用，部署到IIS遇到有问题。）







5、部署到IIS中一定要使用手动配置功能，或者使用api接口注入服务；在conf文件中放入需要手动注入的服务地址；数据中台的本机注入json如下：

{

"encrypt": "7TnJPB4lKtjEcCWWjN6jSA==",

"services": [

{

"id": "CloudCenter\_10.0.194.104:9000",

"name": "CloudCenter",

"tags": [ "CloudCenter9000" ],

"address": "10.0.194.104",

"port": 9000,

"checks": [

{

"id": "CloudCenter\_10.0.194.104:9000",

"name": "CloudCenter",

"http": "http://10.0.194.104:9000/HealthCheck",

"interval": "10s",

"tls\_skip\_verify": false,

"method": "GET",

"timeout": "5s"

}

]

},

{

"id": "CloudCenter\_10.0.194.104:9001",

"name": "CloudCenter",

"tags": [ "CloudCenter9001" ],

"address": "10.0.194.104",

"port": 9001,

"checks": [

{

"id": "CloudCenter\_10.0.194.104:9001",

"name": "CloudCenter",

"http": "http://10.0.194.104:9001/HealthCheck",

"interval": "12s",

"tls\_skip\_verify": false,

"method": "GET",

"timeout": "5s"

}

]

}

]

}

6、常用注入命令如下：

（1）consul agent -dev -client 10.0.194.104 -ui 指定ip可以访问

（2）consul agent -server -datacenter=dc1 -bootstrap -data-dir ./data -config-file ./conf -ui -node=10.0.194.104 -bind 10.0.194.104 -client=0.0.0.0