



# 21天微服务实战营-Day14

华为云DevCloud & ServiceStage服务联合出品



# Day14 CSE实战之其他服务何如接入CSE

## 大纲

- ❑ 创建springboot服务(默认熟练运用springboot)
- ❑ 为Springboot服务绑定mesher
- ❑ CSEGoSDK开发的provider和consumer

## □创建springboot服务

- 创建一个maven项目，引入springboot所需的jar包，为了方便测试，该服务需要具有两个接口一个作为服务提供者接口，另外一个作为客户端接口，该服务将和使用CSEGoSDK开发的服务端和客户端进行访问。

```
8  @RestController
9  @RequestMapping("/consumer/v0")
10 public class ConsumerController {
11     @RequestMapping("/hello/{name}")
12     public String ProviderHello(@PathVariable(value = "name") String name) {
13         RestTemplate restTemplate = new RestTemplate();
14         String s = restTemplate.getForObject( url: "http://provider/provider/v0/hello/" + name, String.class);
15         return s;
16     }
17 }

ConsumerController > ConsumerGreeting()

ProviderController.java x
8  @RestController
9  @RequestMapping("/provider/v0")
10 public class ProviderController {
11     @RequestMapping("/hello/{name}")
12     public String ConsumerHello(@PathVariable(value = "name") String name) { return "hello , " + name; }
15 }
```

# □创建springboot服务

在创建好springboot服务后，我们需要为刚刚创建好的绑定一个mesher，在绑定mesher之前，我们需要将springboot的代理设置为mesher的监听地址。本文通过Java的ProxySelector进行代理设置

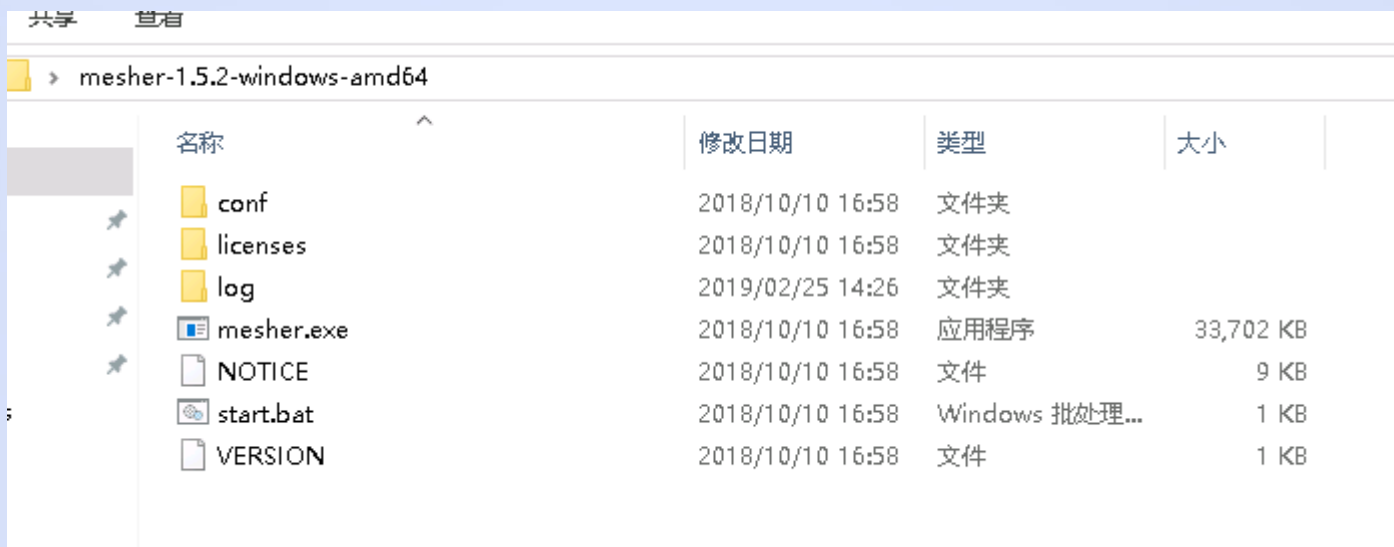
```
public class MyProxySelector extends ProxySelector {  
    @Override  
    public List<Proxy> select(URL uri) {  
        List<Proxy> list = new ArrayList<>();  
        // 设置需要进行代理的地址，此处为mesher的host和port  
        String address = "127.0.0.1";  
        int port = 30101;  
        InetSocketAddress socketAddress = new InetSocketAddress(address, port);  
        Proxy proxy = new Proxy(Proxy.Type.HTTP, socketAddress);  
        list.add(proxy);  
        return list;  
    }  
  
    @Override  
    public void connectFailed(URL uri, SocketAddress sa, IOException ioe) {  
    }  
}
```

```
@SpringBootApplication  
public class Application {  
    public static void main(String[] args) {  
        ProxySelector.setDefault(new MyProxySelector());  
        SpringApplication.run(Application.class, args);  
    }  
}
```

## □绑定mesher

Springboot服务接口以及代理均已开发好时，我们就可以为springboot服务绑定一个mesher了

1. 从华为云下载mesher，请点击[mesher](#)下载  
下载解压如下：



名称	修改日期	类型	大小
conf	2018/10/10 16:58	文件夹	
licenses	2018/10/10 16:58	文件夹	
log	2019/02/25 14:26	文件夹	
mesher.exe	2018/10/10 16:58	应用程序	33,702 KB
NOTICE	2018/10/10 16:58	文件	9 KB
start.bat	2018/10/10 16:58	Windows 批处理...	1 KB
VERSION	2018/10/10 16:58	文件	1 KB

# □绑定mesher

## mesher配置

修改microservice.yaml,将服务名修改为springboot , 并新增APPLICATION\_ID : Training21Days-HelloWorld

```
1  ## microservice property
2  APPLICATION_ID: Training21Days-HelloWorld
3  service_description:
4    name: springboot
5    version: 0.0.1
6    environment: #microservice environment
7    properties:
8      allowCrossApp: false #whether to allow calls across applications
9
```

服务名

mesher版本

修改chassis.yaml,将监听地址改为机器的外部访问地址 , 如 : 192.168.0.1,  
并在该配置文件中添加ak/sk

```
1  ---
2  cse:
3    credentials:
4      accessKey: # you ak
5      secretKey: # you sk
6      project:
7      akskCustomCipher: default
```

## □绑定mesher

### ➤ 启mesher:

除了使用mesher提供的启动脚本启动外我们亦可以直接运行mesher.exe(mesher), 在运行之前, 我们需要设置环境变量 SPECIFIC\_ADDR, SPECIFIC\_ADDR支持格式: rest:port,grpc:port, http:port。可同时写两个如下, 设置后运行mesher.exe(mesher)。

e.g. export SPECIFIC\_ADDR=rest:80

# 绑定mesher

➤ 修改完并启动mesher后，可以在serverstage下看到如下



微服务控制台  
Cloud Service Engine

仪表盘

服务目录

服务治理

全局配置

事务看板

←

服务目录

您可以从应用、微服务和实例的维度查看微服务详细信息。如果微服务较多，您可以通过搜索功能查找目标微服务。[如何维护微服务？](#)

应用列表 微服务列表 实例列表

创建微服务

删除

全部应用(1)

微服务名称

Q

C

<input type="checkbox"/> 微服务名称	所属应用	版本号	实例数	创建时间	操作
<input type="checkbox"/> <a href="#">springboot</a>	default	1	1	2019/02/25 14:26:38 GMT+08:00	<a href="#">删除</a>
<input type="checkbox"/> <a href="#">client-demo</a>	default	1	1	2019/02/25 10:45:06 GMT+08:00	<a href="#">删除</a>
<input type="checkbox"/> <a href="#">server-demo</a>	default	1	1	2019/02/25 09:08:16 GMT+08:00	<a href="#">删除</a>



## □开发CSEGoSDK服务

我们在Day11时已经开发了provider和consumer，此时我们可以直接使用

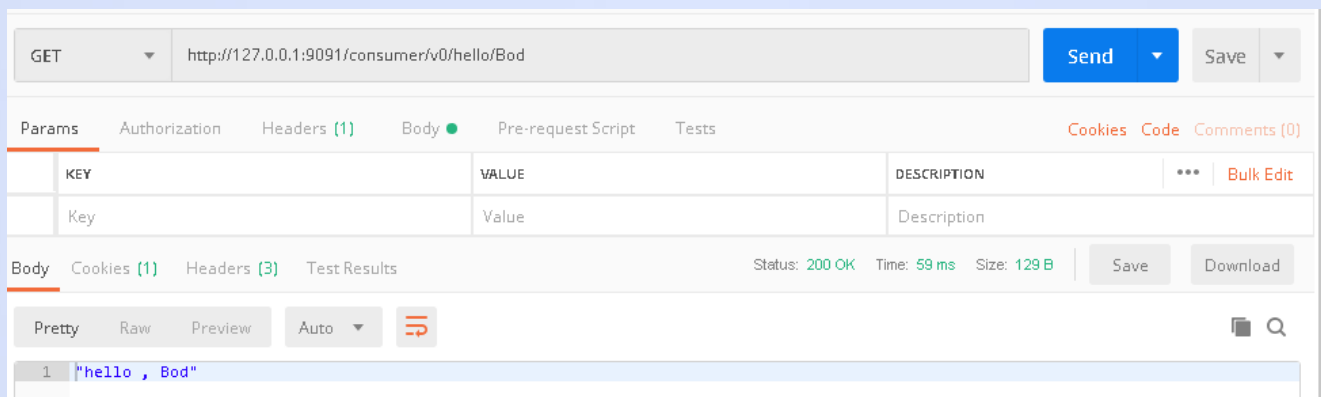
➤ 修改Day11中的consumer访问服务名为springboot

```
func (*Client) Hello(ctx *restful.Context) {  
    //可以使用 cse:// 和 http://作为前缀  
    //req, err := rest.NewRequest(http.MethodGet, fmt.Sprintf("http://server-demo/provider/v0/hello/%s", ctx.ReadPathParameter("name")), nil)  
    req, err := rest.NewRequest(http.MethodGet, fmt.Sprintf("cse://springboot/provider/v0/hello/%s", ctx.ReadPathParameter("name")), nil)  
    if err != nil {
```

同时按照Day11的启动方式将Day11的provider和consumer进行启动

# □访问服务

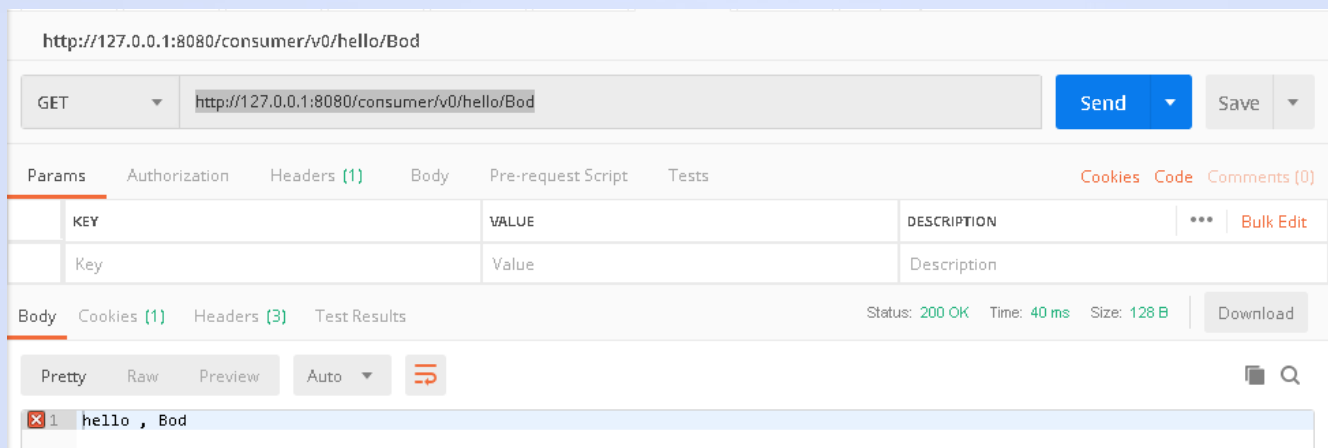
访问springboot服务的consumer接口 , /consumer/v0/{path}。  
调用 <http://127.0.0.1:9091/consumer/v0/hello/Bod> ,反馈结果



# □访问服务

访问Day11服务的consumer接口，/consumer/v0/{path}。

调用 <http://127.0.0.1:8080/consumer/v0/hello/Bod>,反馈结果



## □更多接入mesher-demo参考

- [nodejs接入CSE](#)
- [php接入CSE](#)
- [.Net接入CSE](#)
- [更多](#)

# Thank You

