



# 21天微服务实战营-Day6

华为云DevCloud & ServiceStage服务联合出品



# Day6 CSE实战之开发网关

## 大纲

- 使用DefaultEdgeDispatcher开发网关服务
- 使用URLMappedEdgeDispatcher开发网关服务

# EdgeService网关介绍

EdgeService是CSEJavaSDK提供的网关服务。网关服务可以作为微服务系统的边界，对外提供REST接口，将用户发送的RESTful请求转发给内部微服务。

EdgeService预置了默认的请求转发模块，可以根据简明的规则将请求转发到后端微服务，也允许用户开发更复杂的自定义规则转发机制。

EdgeService本身也是一个微服务，开发和配置方式与普通的微服务类似。用户可以通过扩展机制在EdgeService中定制各种业务功能。

# 使用DefaultEdgeDispatcher开发网关服务

开发一个EdgeService网关服务所需引入的maven配置与开发普通微服务基本一致，只是需要多引入一个edge-core包的依赖：

```
<dependencyManagement>
  <dependencies>
    <dependency>
      <groupId>com.huawei.paas.cse</groupId>
      <artifactId>cse-dependency</artifactId>
      <version>${cse.version}</version>
      <type>pom</type>
      <scope>import</scope>
    </dependency>
  </dependencies>
</dependencyManagement>

<dependencies>
  <dependency>
    <groupId>com.huawei.paas.cse</groupId>
    <artifactId>cse-solution-service-engine</artifactId>
  </dependency>
  <dependency>
    <groupId>org.apache.servicecomb</groupId>
    <artifactId>edge-core</artifactId>
  </dependency>
</dependencies>
```

# 使用DefaultEdgeDispatcher开发网关服务

定义服务的main类和microservice.yaml配置文件：

- Main类的定义与普通微服务完全相同
- microservice.yaml配置文件与普通微服务基本相同，只是需要额外增加一些请求转发规则的配置

```
cse:
  http:
    dispatcher:
      edge:
        default: ..... # 使用DefaultEdgeDispatcher开发网关服务
        enabled: true ..... # 开启DefaultEdgeDispatcher
        prefix: rest ..... # 匹配请求路径前缀为/rest
        withVersion: true ..... # 请求带版本号，例如v1表示[1.0.0,2.0.0)范围内的微服务版本
        # prefixSegmentCount默认值就是1，这里只是展示一下，实际上可以省略该配置
        prefixSegmentCount: 1 ..... # 前缀长度，例如/rest/provider/v0/hello/Bob，去掉第一段，将/provider/v0/hello/Bob转发到后端
```

# 使用DefaultEdgeDispatcher开发网关服务

启动edge、provider、consumer服务，通过edge分别调用provider和consumer，调用成功：

The image displays two screenshots of a REST client interface, likely Swagger UI or Postman, showing successful GET requests.

**Top Screenshot:**

- Method: GET
- URL: 127.0.0.1:8000/rest/provider/v0/hello/Bob
- Status: 200 OK
- Time: 8 ms
- Size: 97 B
- Body: "Hello Bob"

**Bottom Screenshot:**

- Method: GET
- URL: 127.0.0.1:8000/rest/consumer/v0/hello?name=Bob
- Status: 200 OK
- Time: 12 ms
- Size: 97 B
- Body: "Hello Bob"

# 使用URLMappedEdgeDispatcher开发网关服务

除了DefaultEdgeDispatcher，CSEJavaSDK也提供了URLMappedEdgeDispatcher，允许用户指定请求URL和微服务的映射关系。

使用URLMappedEdgeDispatcher开发EdgeService的方法和使用DefaultEdgeDispatcher基本一致，只在microservice.yaml中的请求转发规则上有所不同。

```
cse:
  http:
    dispatcher:
      edge:
        url: ..... # 使用URLMappedEdgeDispatcher开发网关服务
        enabled: true ..... # 开启URLMappedEdgeDispatcher
        mappings:
          providerV0: ..... # 定义名为providerV0的映射规则
            prefixSegmentCount: 1 ..... # 前缀长度为1，例如接到url为/hello/provider/v0/hello/Bob的请求，截去第一段发给provider
            path: "/hello/.*" ..... # 映射规则匹配请求url的正则表达式
            microserviceName: provider ..... # 映射规则对应转发的后端微服务
            versionRule: 0.0.0-1.0.0 ..... # 匹配的微服务版本范围，大于或等于0.0.0，小于1.0.0
          consumerV0:
            prefixSegmentCount: 1
            path: "/client/.*"
            microserviceName: consumer
            versionRule: 0.0.0-1.0.0
```

# 使用URLMappedEdgeDispatcher开发网关服务

启动edge、provider、consumer服务，通过edge分别调用provider和consumer，调用成功：

The image displays two screenshots of a REST client interface, likely Swagger Client or Postman, showing successful GET requests.

**Top Screenshot:**

- Method: GET
- URL: 127.0.0.1:8000/hello/provider/v0/hello/Bob
- Status: 200 OK
- Time: 17057 ms
- Size: 97 B
- Body: "Hello Bob"

**Bottom Screenshot:**

- Method: GET
- URL: 127.0.0.1:8000/client/consumer/v0/hello?name=Alice
- Status: 200 OK
- Time: 3655 ms
- Size: 99 B
- Body: "Hello Alice"



# EdgeService网关转发机制

- 由前文可以看出，DefaultEdgeDispatcher的配置较为简单，但对于EdgeService接收的请求URL格式有一定的要求；URLMappedEdgeDispatcher的转发规则较为灵活，但配置相对更繁琐一点。
- 推荐用户在设计微服务的时候预先规划好各微服务的请求URL格式，可以减少后期服务演进过程中碰到的接口兼容性问题。

# Thank You

