

Day5 教你如何配置你的微服务

1 打卡任务

作业：

- 1、将provider服务打成可执行jar包，体验磁盘上的配置文件覆盖jar包内配置和环境变量覆盖配置文件的功能。
- 2、在hello.sayHelloPrefix配置项上加上回调方法，在配置项改变时打印提示日志。
- 3、开启accesslog功能，并将accesslog日志内容输出到业务日志中。

打卡：

- 1、截取provider刷新动态配置时，触发hello.sayHelloPrefix配置项回调方法打印日志的截图。
- 2、截取日志中accesslog打印的内容。

打卡任务基于Day5的demo项目：



Demo-Day5.zip

2 准备工作

- 1、正常运行Day5的demo

3 验证微服务配置优先级关系

- 1、在provider服务的pom文件中加上复制依赖包和打可执行jar包的插件配置

```

<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-dependency-plugin</artifactId>
      <version>2.10</version>
      <executions>
        <execution>
          <id>copy-dependencies</id>
          <phase>package</phase>
          <goals>
            <goal>copy-dependencies</goal>
          </goals>
          <configuration>
            <outputDirectory>target/lib</outputDirectory>
          </configuration>
        </execution>
      </executions>
    </plugin>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-jar-plugin</artifactId>
      <version>2.6</version>
      <configuration>
        <archive>
          <manifest>
            <addClasspath>true</addClasspath>
            <classpathPrefix>./lib/</classpathPrefix>
            <mainClass>${main.class}</mainClass>
            <addDefaultImplementationEntries>true</addDefaultImplementationEntries>
            <addDefaultSpecificationEntries>true</addDefaultSpecificationEntries>
          </manifest>
          <manifestEntries>
            <Class-Path>./</Class-Path>
          </manifestEntries>
        </archive>
      </configuration>
    </plugin>
  </plugins>
</build>

```

注意在properties中增加一个main.class的配置，指向provider服务的Main类

```

<properties>
  <cse.version>2.3.62</cse.version>
  <main.class>microservice.demo.training21days.provider.AppMain</main.class>
</properties>

```

- 2、执行mvn clean package，在target目录下可以看到打好的jar包和对应的lib目录。实际部署到执行环境的时候需要把jar包和lib目录同时复制过去。

在命令行中执行java -jar helloworld-provider-0.0.1-SNAPSHOT.jar，启动服务。

classes	2019/2/17 17:54	文件夹
lib	2019/2/17 17:55	文件夹
maven-archiver	2019/2/17 17:54	文件夹
maven-status	2019/2/17 17:54	文件夹
helloworld-provider-0.0.1-SNAPSHOT.jar	2019/2/17 17:54	JAR 文件 10 KB

```

C:\Windows\System32\cmd.exe
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

D:\doc\21Days\Demo\helloworld-provider\target>java -jar helloworld-provider-0.
  
```

通过consumer调用该provider服务的sayHello方法，此时返回的前缀是Hello

```

GET http://127.0.0.1:9090/consumer/v0/hello?name=Alice

Pretty Raw Preview JSON
1 "Hello Alice"
  
```

3、在jar包所在目录下放置一份microservice.yaml配置文件，内容如下

classes	2019/2/17 17:54	文件夹
lib	2019/2/17 17:55	文件夹
maven-archiver	2019/2/17 17:54	文件夹
maven-status	2019/2/17 17:54	文件夹
helloworld-provider-0.0.1-SNAPSHOT.jar	2019/2/17 17:54	JAR 文件
microservice.yaml	2019/2/17 18:08	YAML 文件

```

D:\doc\21Days\Demo\helloworld-provider\target\microservice.yaml -
File Edit Selection Find View Goto Tools Project Preferences Help

microservice.yaml x
1 hello:
2   sayHelloPrefix: "Hello(from disk file)"
  
```

重新启动provider服务的jar包，此时得到的应答如下

```

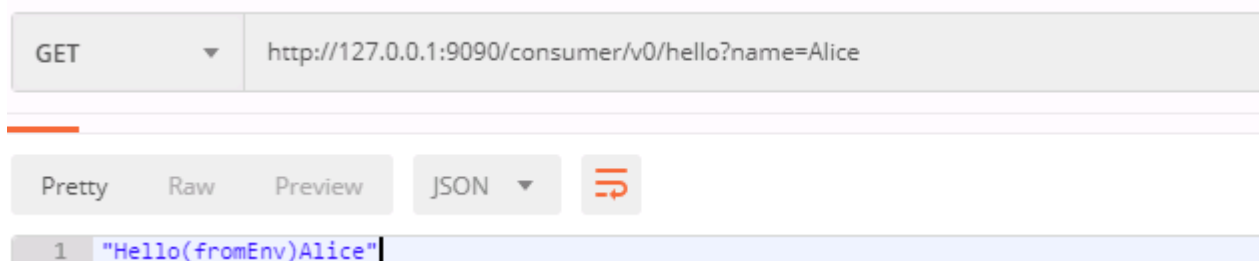
GET http://127.0.0.1:9090/consumer/v0/hello?name=Alice

Pretty Raw Preview JSON
1 "Hello(from disk file) Alice"
  
```

4、设置环境变量hello_sayHelloPrefix=Hello(fromEnv)，再次启动服务

```
D:\doc\21Days\Demo\helloworld-provider\target>set hello_sayHelloPrefix=Hello(f
D:\doc\21Days\Demo\helloworld-provider\target>java -jar helloworld-provider-0.
```

此时调用sayHello方法得到的应答如下



4 给配置项增加回调方法

- 1、在provider服务的HelloService类中增加一个sayHelloPrefix配置项的回调，使其在配置项被刷新时打印一行日志

```
..private static final Logger LOGGER = LoggerFactory.getLogger(HelloService.class);

..private DynamicStringProperty sayHelloPrefix = DynamicPropertyFactory
.....getInstance().getStringProperty( propName: "hello.sayHelloPrefix", defaultValue: ""
....., notifyConfigRefreshed());

..private Runnable notifyConfigRefreshed() {
..    return () -> LOGGER.info("config[hello.sayHelloPrefix] changed to [{}]", sayHelloPre
..}
```

- 2、启动服务，在provider服务的动态配置页面增加hello.sayHelloPrefix的配置

创建配置

作用域

provider@Training21Days-Hello...

* 配置项

hello.sayHelloPrefix

* 值

Hello(fromCC)

确定

取消

待provider服务实例从配置中心刷新配置项后，就会触发回调方法打印日志

```
[INFO] config[hello.sayHelloPrefix] changed to [Hello(fromCC)]! microservice.demo.trainin
[WARN] Config value cache changed: action:create; item:[hello.sayHelloPrefix] org.apache.
[INFO] Updating remote config is done. revision has changed from default94785601 to defau
```

5 启用和配置 accesslog

1、开启accesslog功能

Accesslog功能的说明文档在这里：[https://docs.servicecomb.io/java-](https://docs.servicecomb.io/java-chassis/zh_CN/build-provider/access-log-configuration.html)

[chassis/zh_CN/build-provider/access-log-configuration.html](https://docs.servicecomb.io/java-chassis/zh_CN/build-provider/access-log-configuration.html)。

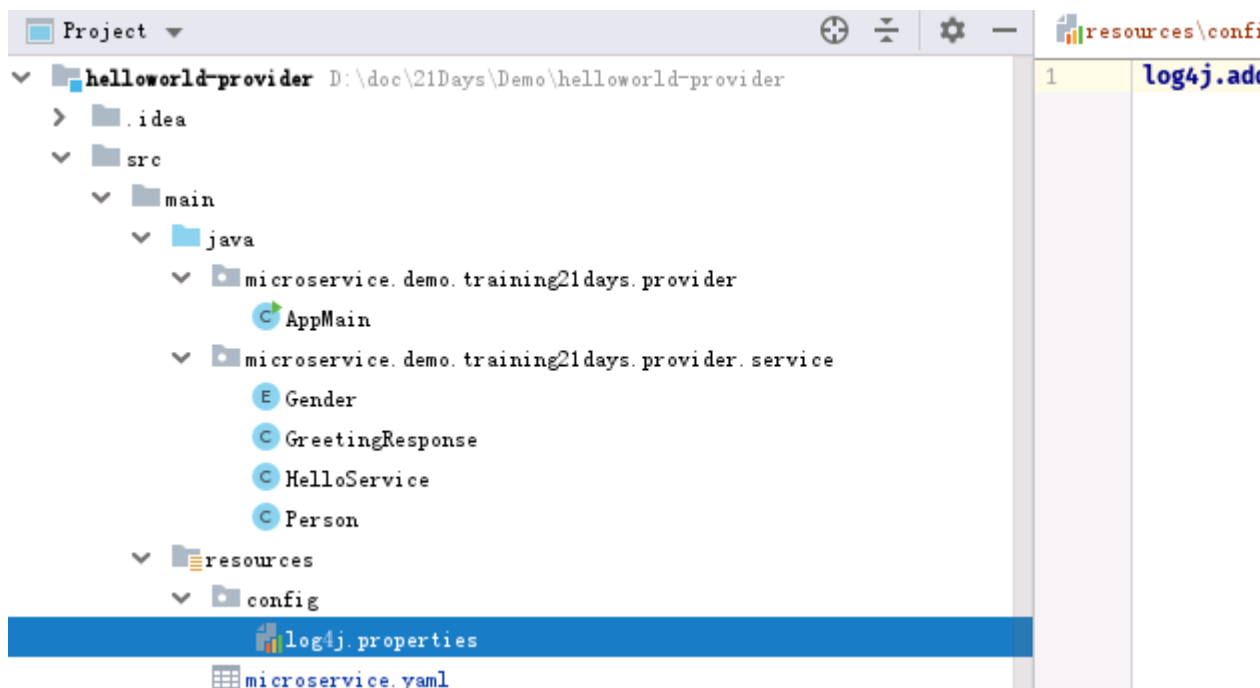
开启accesslog需要在microservice.yaml文件中做如下配置（这里使用了自定义的日志格式，没有用默认配置）：

```
..accesslog:
..  enabled: true
..  pattern: "%h - - %t cs-uri %s %B %D %SCB-traceId"
```

2、配置accesslog日志打印配置

默认的accesslog是单独打印到一个access.log日志文件中的。这里我们为了实验观察

的方便，在项目的resources/config目录下放置一份log4j.properties配置文件，覆盖默认的配置，令accesslog的内容合并到普通业务日志中输出。



3、启动服务，调用接口，可以看到日志中打印了accesslog

```
127.0.0.1 - - Sun, 17 Feb 2019 18:46:10 CST /provider/v0/hello/Bob.200 11.650 5c693b723b9
127.0.0.1 - - Sun, 17 Feb 2019 18:46:13 CST /provider/v0/hello/Bob.200 11.3 5c693b75185db
127.0.0.1 - - Sun, 17 Feb 2019 18:46:14 CST /provider/v0/hello/Bob.200 11.2 5c693b766b0e1
127.0.0.1 - - Sun, 17 Feb 2019 18:46:17 CST /provider/v0/hello/Bob.200 11.2 5c693b79a0d45
```

6 打卡截图

1.截取provider刷新动态配置时，触发hello.sayHelloPrefix配置项回调方法打印日志的截图。

2.截取日志中accesslog打印的内容。

1、Provider动态刷新配置项时的回调日志

```
[INFO] config[hello.sayHelloPrefix] changed to [Hello(fromCC)]! microservice.demo.trainin
[WARN] Config value cache changed: action:create; item:[hello.sayHelloPrefix] org.apache.
[INFO] Updating remote config is done. revision has changed from default94785601 to defau
```

2、Accesslog日志

```
127.0.0.1 - - Sun, 17 Feb 2019 18:46:10 CST /provider/v0/hello/Bob.200 11.650 5c693b723b9
127.0.0.1 - - Sun, 17 Feb 2019 18:46:13 CST /provider/v0/hello/Bob.200 11.3 5c693b75185db
127.0.0.1 - - Sun, 17 Feb 2019 18:46:14 CST /provider/v0/hello/Bob.200 11.2 5c693b766b0e1
127.0.0.1 - - Sun, 17 Feb 2019 18:46:17 CST /provider/v0/hello/Bob.200 11.2 5c693b79a0d45
```

7 小提示

- 1、对于CSEJavaSDK而言，框架自身的一些治理配置项和用户定义的业务配置项是同等地位的，即内部处理逻辑是完全一致的。用户也可以遵循一样的配置和使用方法。
- 2、日志配置是一个独立的体系，与业务配置不相关。

参考答案：

