Spring Cloud Function

Spring Cloud Function致力于促进函数作为主要的开发单元，提供一个通用的模型，用于在各种平台上部署基于函数的软件，例如Amazon AWS Lambda这样的FaaS平台。该项目致力于将函数变成开发人员所使用的主要理念，具有以下特征：

* 业务逻辑以函数的方式实现
* 业务逻辑和Runtime Target相解耦，相同的代码可以以函数的形式暴露为RESTFul服务、流处理或者Task
* 提供标准的编程模型，从而可以支持不同的Serverless Provider，部署在不同FaaS平台上，也可以部署为独立的应用
* 使用Spring boot特性，auto-configuration,dependency injection,metric等

# 快速指南

1. 编译

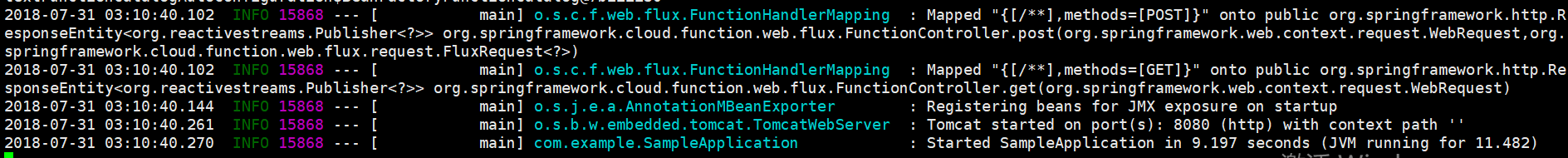
*git clone <https://github.com/spring-cloud/spring-cloud-function.git>*

*./mvnw clean install*

1. 启动Sample

*$ java -jar spring-cloud-function-samples/function-sample/target/\*.jar*

输出信息如下：



1. 测试

*# curl -H "Content-Type: text/plain" localhost:8080/uppercase -d Hello*

*HELLO*

*curl -H "Content-Type: text/plain" localhost:8080/hello*

*Hello*

1. 函数的定义

* 业务逻辑可以使用java.util.function包中所定义的核心接口：Function,ConSumer及Supplier的实现进行开发，开发人员只需要关心业务实现及参数
* 继承Function接口

1. 在项目中提供Function的方式有两种

* 通过注释Bean，定义如下：

*@FunctionScan*

*@SpringBootApplication*

*public class SampleApplication {*

*@Bean*

*public Function<String, String> uppercase() {*

*return value -> value.toUpperCase();*

*}*

*@Bean*

*public Supplier<String> hello() {*

*return () -> "hello";*

*}*

*}*

在类定义中添加@FunctionScan

* 自定义Function类，在配置中application.properties中添加参数：

*spring.cloud.function.scan.packages: com.example.functions*

在该package中定义Function类，如戏

*package com.example.functions;*

*import java.util.function.Function;*

*public class Greeter implements Function<String, String> {*

*@Override*

*public String apply(String name) {*

*return "Hello " + name;*

*}*

*}*

访问如下：

*# curl -H "Content-Type: text/plain" localhost:8080/greeter -d "Tom"*

*Hello Tom*

1. Function可以封装为多种Runtime，本处以RESTful的形式提供，在pom.xml中添加：

*<dependency>*

*<groupId>org.springframework.cloud</groupId>*

*<artifactId>spring-cloud-starter-function-web</artifactId>*

*</dependency>*

封装成Task及Stream会在后文中进行介绍

# **Spring-Cloud-Function Dynamic Compilation**

在Spring-Cloud Function中支持将lambda表达式动态的编译成Function实例，这对于实现FaaS平台至关重要，其使用如下（Function Sample），可以通过项目中script进行测试

1. 启动function registry服务

*$./function-registry.sh*

1. 注册函数

*./registerFunction.sh -n uppercase -f "f->f.map(s->s.toString().toUpperCase())"*

在service端输出入下：



在默认目录/tmp/function-registry的数据目录中生成uppercase.fun文件

*# tree -L 2 function-registry/*

├─*─ consumers*

├─*─ functions*

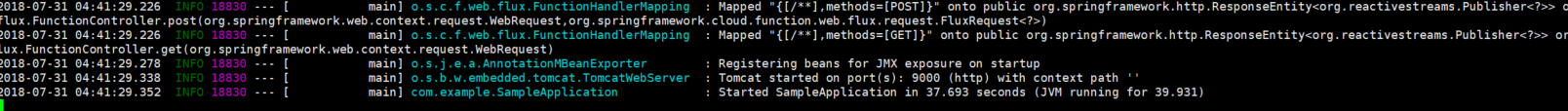
│└── *uppercase.fun*

└─*─ suppliers*

1. 启动REST微服务使用注册的Function

* 测试Function

*# ./web.sh -f uppercase -p 9000*



使用命令如下：

*curl -H "Content-Type: text/plain" -H "Accept: text/plain" localhost:9000/uppercase -d foo*

*输出结果：*

*FOO*

* 测试Supplier

*./web.sh -s words -p 9010*

*curl -H "Accept: application/json" localhost:9001/words*

*["foo","bar"]*

* 测试Consumer

*./web.sh -c print -p 9002*

*curl -X POST -H "Content-Type: text/plain" -d foo localhost:9002/print*

*https://github.com/spring-cloud/spring-cloud-function/blob/master/docs/src/main/asciidoc/spring-cloud-function.adoc*

*https://spring.io/blog/2017/07/05/introducing-spring-cloud-function*