### 一、简介

在分布式系统中，spring cloud config 提供一个服务端和客户端去提供可扩展的配置服务。我们可用用配置服务中心区集中的管理所有的服务的各种环境配置文件。配置服务中心采用**[Git](http://lib.csdn.net/base/git" \o "Git知识库" \t "http://blog.csdn.net/forezp/article/details/_blank)**的方式存储配置文件，因此我们很容易部署修改，有助于对环境配置进行版本管理。

### 二、构建Config Server

创建一个spring-boot项目，取名为config-server,其pom.xml:

<?xml version="1.0" encoding="UTF-8"?><project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.forezp</groupId>

<artifactId>config-server</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>jar</packaging>

<name>config-server</name>

<description>Demo project for Spring Boot</description>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>1.5.2.RELEASE</version>

<relativePath/> <!-- lookup parent from repository -->

</parent>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>

<java.version>1.8</java.version>

</properties>

<dependencies>

<dependency>

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

<artifactId>spring-cloud-config-server</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

<dependency>

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

<artifactId>spring-cloud-starter-eureka</artifactId>

</dependency>

</dependencies>

<dependencyManagement>

<dependencies>

<dependency>

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

<artifactId>spring-cloud-dependencies</artifactId>

<version>Camden.SR6</version>

<type>pom</type>

<scope>import</scope>

</dependency>

</dependencies>

</dependencyManagement>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

<repositories>

<repository>

<id>spring-milestones</id>

<name>Spring Milestones</name>

<url>https://repo.spring.io/milestone</url>

<snapshots>

<enabled>false</enabled>

</snapshots>

</repository>

</repositories>

</project>

在程序的入口Application类加上@EnableConfigServer注解开启配置服务器。

@SpringBootApplication

@EnableConfigServer

public class ConfigServerApplication {

public static void main(String[] args) {

SpringApplication.run(ConfigServerApplication.class, args);

}

}

需要在配置中心配置下：

spring.application.name=config-server

server.port=8888

spring.cloud.config.server.git.uri=https://github.com/forezp/SpringcloudConfig/

spring.cloud.config.server.git.searchPaths=respo

spring.cloud.config.label=master

spring.cloud.config.server.git.username=your username

spring.cloud.config.server.git.password=your password

* spring.cloud.config.server.git.uri：配置git仓库地址
* spring.cloud.config.server.git.searchPaths：配置仓库路径
* spring.cloud.config.label：配置仓库的分支
* spring.cloud.config.server.git.username：访问git仓库的用户名
* spring.cloud.config.server.git.password：访问git仓库的用户密码

远程仓库[https://github.com/forezp/SpringcloudConfig/](https://github.com/forezp/SpringcloudConfig/" \t "http://blog.csdn.net/forezp/article/details/_blank) 中又个文件config-client-dev.properties文件中有一个属性：

foo = foo version 3

启动程序：访问[http://localhost:8888/foo/dev](http://localhost:8888/foo/dev" \t "http://blog.csdn.net/forezp/article/details/_blank)

{"name":"foo","profiles":["dev"],"label":"master",

"version":"792ffc77c03f4b138d28e89b576900ac5e01a44b","state":null,"propertySources":[]}

证明配置服务中心可以从远程程序获取配置信息。

http请求地址和资源文件映射如下:

* /{application}/{profile}[/{label}]
* /{application}-{profile}.yml
* /{label}/{application}-{profile}.yml
* /{application}-{profile}.properties
* /{label}/{application}-{profile}.properties

### 三、构建一个config client

重新创建一个springboot项目，取名为config-client,其pom文件：

<?xml version="1.0" encoding="UTF-8"?><project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.forezp</groupId>

<artifactId>config-client</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>jar</packaging>

<name>config-client</name>

<description>Demo project for Spring Boot</description>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>1.5.2.RELEASE</version>

<relativePath/> <!-- lookup parent from repository -->

</parent>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>

<java.version>1.8</java.version>

</properties>

<dependencies>

<dependency>

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

<artifactId>spring-cloud-starter-config</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

</dependencies>

<dependencyManagement>

<dependencies>

<dependency>

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

<artifactId>spring-cloud-dependencies</artifactId>

<version>Dalston.RC1</version>

<type>pom</type>

<scope>import</scope>

</dependency>

</dependencies>

</dependencyManagement>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

<repositories>

<repository>

<id>spring-milestones</id>

<name>Spring Milestones</name>

<url>https://repo.spring.io/milestone</url>

<snapshots>

<enabled>false</enabled>

</snapshots>

</repository>

</repositories>

</project>

其配置文件：

spring.application.name=config-client

spring.cloud.config.label=master

spring.cloud.config.profile=dev

spring.cloud.config.uri= http://localhost:8888/

server.port=8881

spring.cloud.config.label 指明远程仓库的分支

spring.cloud.config.profile

* + dev开发环境配置文件
  + test测试环境
  + pro正式环境

spring.cloud.config.uri= [http://localhost:8888/](http://localhost:8888/" \t "http://blog.csdn.net/forezp/article/details/_blank) 指明配置服务中心的网址。

程序的入口类：

@SpringBootApplication

@RestController

public class ConfigClientApplication {

public static void main(String[] args) {

SpringApplication.run(ConfigClientApplication.class, args);

}

@Value("${foo}")

String foo;

@RequestMapping(value = "/hi")

public String hi(){

return foo;

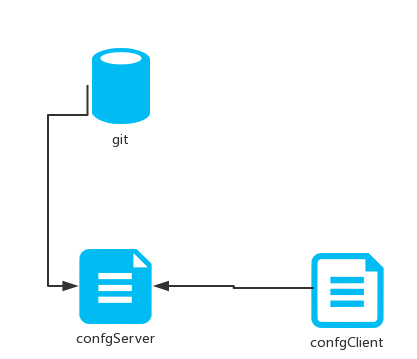
}

}

打开网址访问：[http://localhost:8881/hi](http://localhost:8881/hi" \t "http://blog.csdn.net/forezp/article/details/_blank)，网页显示：

foo version 3

这就说明，config-client从config-server获取了foo的属性，而config-server是从**[git](http://lib.csdn.net/base/git" \o "Git知识库" \t "http://blog.csdn.net/forezp/article/details/_blank)**仓库读取的,如图：



本文源码下载：   
[https://github.com/forezp/SpringCloudLearning/tree/master/chapter6](https://github.com/forezp/SpringCloudLearning/tree/master/chapter6" \t "http://blog.csdn.net/forezp/article/details/_blank)

### 四、参考资料

[spring\_cloud\_config](http://projects.spring.io/spring-cloud/spring-cloud.html" \l "_spring_cloud_config" \t "http://blog.csdn.net/forezp/article/details/_blank)

Git同步用tortoiseGit