

springcloud 配置中心 (基于 svn)

作者: [biristone](#)

原文链接: <https://ld246.com/article/1520755235493>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

一、新建一个配置中心springboot工程

maven配置如下:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/MLSchema-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.biristone</groupId>
<artifactId>configure</artifactId>
<version>0.0.1-SNAPSHOT</version>
<packaging>jar</packaging>
```

```
<name>configure</name>
<description>biristone</description>
```

```
<parent>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-parent</artifactId>
  <version>2.0.0.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>
  <spring-cloud.version>Finchley.M8</spring-cloud.version>
  <eureka.version>1.4.0.RELEASE</eureka.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>
  <version>${eureka.version}</version>
</dependency>
<dependency>
```

```

        <groupId>org.tmatesoft.svnkit</groupId>
        <artifactId>svnkit</artifactId>
    </dependency>
</dependencies>

<dependencyManagement>
    <dependencies>
        <dependency>
            <groupId>org.springframework.cloud</groupId>
            <artifactId>spring-cloud-dependencies</artifactId>
            <version>${spring-cloud.version}</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>

```

主程序:

```

package com.biristone;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.cloud.config.server.EnableConfigServer;

@SpringBootApplication
@EnableConfigServer
public class ConfigureApplication {

    public static void main(String[] args) {

```

```
SpringApplication.run(ConfigureApplication.class, args);
}
}
```

application.yml内容:

server:

port: 9527

spring:

application:

name: xxx-service-config

cloud:

config:

server:

svn:

uri: <http://192.168.1.111/svn/code/cloude>

username: xxx

password: xxx

default-label: configsource #<http://192.168.1.111/svn/code/cloude/configsource> 为配置文件在路径,如果直接配置为uri是个坑

profiles:

active: subversion

二、客户端配置springboot配置

maven配置:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/
MLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/mav
n-4.0.0.xsd">
```

```
<modelVersion>4.0.0</modelVersion>
```

```
<groupId>com.biristone</groupId>
```

```
<artifactId>xxxboot</artifactId>
```

```
<version>1.0.0</version>
```

```
<packaging>jar</packaging>
```

```
<name>xxxboot</name>
```

```
<description>promotion project for Jmm</description>
```

```
<parent>
```

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

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

```

    <version>1.5.9.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>
  <start-class>com.jmm.drools.DroolsSpringbootApplication</start-class>
  <drools.version>6.5.0.Final</drools.version>
  <druid.version>1.1.5</druid.version>
  <druid.starter.version>1.1.5</druid.starter.version>
  <mybatis.version>3.4.1</mybatis.version>
  <mybatis-spring-boot.version>1.1.1</mybatis-spring-boot.version>
  <eureka.version>1.4.0.RELEASE</eureka.version>
  <archaius.version>0.6.1</archaius.version>
</properties>

<dependencies>
  <dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-eureka</artifactId>
    <version>${eureka.version}</version>
  </dependency>
  <dependency>
    <groupId>com.netflix.archaius</groupId>
    <artifactId>archaius-core</artifactId>
    <version>${archaius.version}</version>
  </dependency>
  <!--Drools-->
  <dependency>
    <groupId>org.kie</groupId>
    <artifactId>kie-api</artifactId>
    <version>${drools.version}</version>
  </dependency>
  <dependency>
    <groupId>org.drools</groupId>
    <artifactId>drools-core</artifactId>
    <version>${drools.version}</version>
  </dependency>
  <dependency>
    <groupId>org.drools</groupId>
    <artifactId>drools-compiler</artifactId>
    <version>${drools.version}</version>
  </dependency>
  <dependency>
    <groupId>org.drools</groupId>
    <artifactId>drools-decisiontables</artifactId>
    <version>${drools.version}</version>
  </dependency>
  <dependency>
    <groupId>org.drools</groupId>
    <artifactId>drools-templates</artifactId>
    <version>${drools.version}</version>
  </dependency>

```

```

</dependency>

<!--SpringBoot core-->
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter</artifactId>
  <exclusions>
    <exclusion>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-logging</artifactId>
    </exclusion>
  </exclusions>
</dependency>

<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-data-redis</artifactId>
</dependency>

<!--log4j-->
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-log4j2</artifactId>
</dependency>

<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-aop</artifactId>
</dependency>
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-web</artifactId>
  <exclusions>
    <exclusion>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-tomcat</artifactId>
    </exclusion>
  </exclusions>
</dependency>
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-undertow</artifactId>
</dependency>
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-actuator</artifactId>
</dependency>
<!--db-->
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
</dependency>
<!-- dependency>

```

```
<groupId>org.apache.tomcat</groupId>
<artifactId>tomcat-jdbc</artifactId>
</dependency-->

<dependency>
  <groupId>com.alibaba</groupId>
  <artifactId>druid</artifactId>
  <version>${druid.version}</version>
</dependency>
<dependency>
  <groupId>com.alibaba</groupId>
  <artifactId>druid-spring-boot-starter</artifactId>
  <version>${druid.starter.version}</version>
</dependency>

<dependency>
  <groupId>com.alibaba</groupId>
  <artifactId>fastjson</artifactId>
  <version>1.2.12</version>
</dependency>

<dependency>
  <groupId>org.mybatis</groupId>
  <artifactId>mybatis</artifactId>
  <version>${mybatis.version}</version>
</dependency>

<dependency>
  <groupId>org.mybatis.spring.boot</groupId>
  <artifactId>mybatis-spring-boot-starter</artifactId>
  <version>${mybatis-spring-boot.version}</version>
</dependency>

<!--Json-->
<dependency>
  <groupId>com.hynnet</groupId>
  <artifactId>json-lib</artifactId>
  <version>2.4</version>
</dependency>
<dependency>
  <groupId>com.google.code.gson</groupId>
  <artifactId>gson</artifactId>
</dependency>

<!--swagger-->
<dependency>
  <groupId>io.springfox</groupId>
  <artifactId>springfox-swagger2</artifactId>
  <version>2.5.0</version>
</dependency>
<dependency>
  <groupId>io.springfox</groupId>
  <artifactId>springfox-swagger-ui</artifactId>
  <version>2.5.0</version>
```

```

</dependency>

<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-test</artifactId>
  <scope>test</scope>
</dependency>
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-jdbc</artifactId>
  <version>5.0.0.BUILD-SNAPSHOT</version>
</dependency>
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-configuration-processor</artifactId>
  <optional>true</optional>
</dependency>
<dependency>
  <groupId>org.apache.commons</groupId>
  <artifactId>commons-lang3</artifactId>
  <version>3.4</version>
</dependency>
<dependency>
  <groupId>org.quartz-scheduler</groupId>
  <artifactId>quartz</artifactId>
  <version>2.2.2</version>
</dependency>
<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-starter-config</artifactId>
  <version>1.4.0.RELEASE</version>
</dependency>
</dependencies>

  <dependencyManagement>
<dependencies>
  <dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-dependencies</artifactId>
    <version>Camden.SR5</version>
    <type>pom</type>
    <scope>import</scope>
  </dependency>
</dependencies>
</dependencyManagement>

<repositories>
  <repository>
    <id>spring-snapshots</id>
    <url>http://repo.spring.io/libs-snapshot</url>
  </repository>
</repositories>

<pluginRepositories>

```



```

    <pluginRepository>
      <id>spring-snapshots</id>
      <url>http://repo.spring.io/libs-snapshot</url>
    </pluginRepository>
  </pluginRepositories>

  <profiles>
    <profile>
      <id>fullProfile</id>
      <activation>
        <property>
          <name>full</name>
        </property>
      </activation>
    </profile>
  </profiles>

  <build>
    <plugins>
      <plugin>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-maven-plugin</artifactId>
        <configuration>
          <mainClass>${start-class}</mainClass>
        </configuration>
        <executions>
          <execution>
            <goals>
              <goal>repackage</goal>
            </goals>
          </execution>
        </executions>
      </plugin>
    </plugins>
  </build>

</project>

```

** 需要注意最好一定要配置到bootstrap.yml里,如果没有最好新建一个bootstrap.yml

** 内容如下:

spring:

cloud:

config:

uri: http://localhost:9527/

profile: dev

** 客户端主程序注入配置中心注解:

** package com.biristone.drools;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.boot.SpringApplication;

```

import org.springframework.boot.autoconfigure.EnableAutoConfiguration;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.cloud.client.discovery.EnableDiscoveryClient;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.ComponentScan;
import org.springframework.context.annotation.Configuration;
import org.springframework.core.env.Environment;
import org.springframework.scheduling.annotation.EnableScheduling;
import springfox.documentation.builders.PathSelectors;
import springfox.documentation.builders.RequestHandlerSelectors;
import springfox.documentation.spi.DocumentationType;
import springfox.documentation.spring.web.plugins.Docket;
import springfox.documentation.swagger2.annotations.EnableSwagger2;

@SpringBootApplication
@ComponentScan(basePackages = "com.biristone.drools")
@EnableScheduling
@EnableSwagger2
@Configuration
@EnableAutoConfiguration
@EnableDiscoveryClient
public class DroolsSpringbootApplication {

    public static void main(String[] args) {
        SpringApplication.run(DroolsSpringbootApplication.class, args);
    }

    @Bean
    public Docket createRestApi() {
        return new Docket(DocumentationType.SWAGGER_2)
            .select()
            .apis(RequestHandlerSelectors.basePackage("com.jmm.drools.web"))
            .paths(PathSelectors.any())
            .build();
    }

    @Autowired
    void setEnvironment(Environment env) {
        //appname为配置文件里的某个属性名
        System.out.println("++++++++++++++++++++++++++++++++++++++++++++++++++++++++env: " + env.getProperty(
            appname));
    }
}

```

```
}
```

Controller从配置中心读取属性:

```
@Value("${appname}")
private String appname;
@Value("${appvalue}")
private String appvalue;
@ApiOperation(value = "配置信息")
@RequestMapping(value = "/config", method = RequestMethod.POST, produces = "text/html; charset=UTF-8")
public String config() {
    return appname+"|" +appvalue;
}
```