

分布式配置管理中心

- 1 概述
- 2 代码配置
 - 2.1 客户端添加依赖
 - 2.2 重构配置文件
 - 2.3 配置文件内容
 - 2.4 修改单元测试配置
- 3 配置文件仓库
 - 3.1 配置仓库存放规则
 - 3.2 查看配置文件
 - 3.3 修改配置文件
 - 3.4 加密解密
 - 3.5 常见问题
- 4 自动刷新配置文件
 - 4.1 监听配置文件更新的事件
- 5 本地调试

概述

代码中不放置配置文件。配置文件统一通过GIT仓库管理。 dev, test, uat三个环境可以使用共同的配置管理服务器。

优点：代码打包后可以直接指定profile来启动不同环境。修改配置文件不用重新打包，非基础连接类的配置可以不用重启服务即可更新配置。

地址：

[opra-config-repo/browse](#)

代码配置

客户端添加依赖

```
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-config<
/artifactId>
</dependency>
```

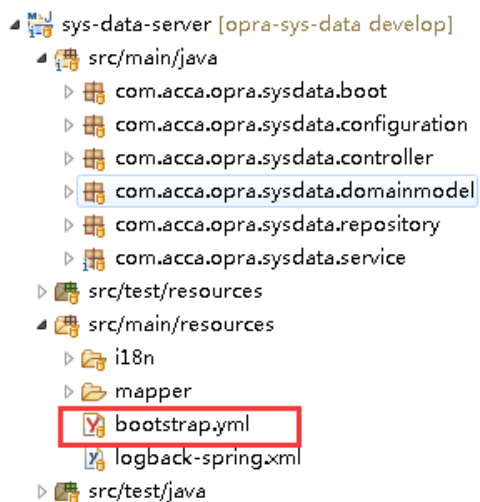
架构并没有添加这个依赖，是为了考虑不是所有项目都必须使用配置管理服务器。所以，这个依赖需要手工添加到自己的代码pom中。

重构配置文件

必须添加bootstrap.yml文件，本地resource目录中只需要保留一个bootstrap.yml文件。其余所有文件都通过配置管理服务器获取。

本地开发人员可以在自己的feature分支中，将bootstrap中的uri清空，本地保留一个application.yml文件，用于本地启动调试等工作。

如下图所示：



配置文件内容

```
spring:
  application:
    name: sys-data
  cloud:
    config:
      uri: http://10.1.17.234:8888,http://10.1.17.17:8888 #
      username: admin
      password: Acca@1234
```

修改单元测试配置

在单元测试resource文件目录中添加bootstrap.yml文件

内容如下:

```
spring:
  application:
    name: sys
  cloud:
    config:
      uri:
```

这里配置uri为空，默认仍然使用本地的application.yml文件。单元测试目录下的application.yml文件不变。

配置文件仓库

<http://git.acca.com.cn:7990/projects/OPRA-GIT/repos/opra-config-repo/browse>

配置仓库存放规则

根目录下第一级为大的项目名称例如：opra

第二级目录为子系统的名字，例如sys mas sal等

第三级为具体的配置文件，例如 application-dev.yml等

配置文件的profile中不能有-，例如uat-ca是不符合规范的，需要改成uatca。

配置文件的application-uatca.yml，文件名称中的 application可要可不要。

查看配置文件

地址	用户名	密码
http://10.1.17.234:8888	admin	Acca@1234
http://10.1.17.17:8888	admin	Acca@1234

访问下面的地址，可以查看当前配置文件的详细信息：

<http://10.1.17.234:8888/sys-data-dev.yml>

<http://10.1.17.234:8888/{spring.application.name}-{profile}.yml>

```
/{application}-{profile}.yml

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

{application}  spring.config.name
{profile}     spring.profiles.active
{label}      spring.cloud.config.label  (defaults to master.)
```

修改配置文件

将config-repo的git库clone到本地，并修改，提交。

加密解密

通过post方法对数据项进行加密。例如：

```
curl -u admin:Acca@1234 http://localhost:8888/encrypt -d mas
```

将加密的字符串配置到配置文件中

```
datasource:
  username: opramas
  password: '{cipher}
46ef87b2ba6ef295b5a0a19834f0befc43f036faa31b5b7be824d02c8279949a'
```

常见问题

1、之前在本地是正常的配置文件，改成配置管理服务器之后，产生问题。可能的原因

```
application
# loggingdebug
logging:
  level:
    com.acca.opra.sys.dao: DEBUG

local
# loggingdebug
logging:
  level:
    com.acca.opra.sys: DEBUG

config serverbug
```

2、如果本地启动报错，可以手工查询配置管理服务器，看看配置文件是否正常显示。如出现下面的错误，需要登录到配置管理服务器的机器，看一下日志。地址：/data/springboot/config-server/logs/araf-config-server.log

Whitelabel Error Page

This application has no explicit mapping for /error, so you are seeing this as a fallback.

Sun Apr 26 16:41:07 CST 2020

There was an unexpected error (type=Internal Server Error, status=500).

java.lang.String cannot be cast to java.util.Map

3、保证default配置文件，即application.yml中的配置项是齐全的。这样其他profile的配置文件中的项才能覆盖，否则，能出现只在某个profile中配置了但是无法生效的问题。

自动刷新配置文件

客户端代码中必须使用@RefreshScope注解标识，例如下面：

```
@Service
@Slf4j
@Getter
@RefreshScope
public class KafkaLogAutoCleanEsIndex {

    @Value("${datalog.retain.days: 7}")
    private int days;
```

配置文件修改并提交之后，可以访问一下配置管理服务器的刷新地址。

<http://10.1.17.234:8888/refresh>

自动通知所有注册上来的微服务，并进行配置文件数据的更新。

监听配置文件更新的事件

当配置文件更新后，如果代码中需要监听到配置文件更新的数据，触发一些操作时，可以如下设计：

实现一个spring的ApplicationListener，泛型RefreshEvent即可。

```
@Component
@Slf4j
@Data
public class JobQueuePartitionListener implements
ApplicationListener<RefreshEvent> {

    @Override
    public void onApplicationEvent(RefreshEvent event) {
        if (event.getKeys().contains("spring.adp.partitions")) { // key
            changePartitions(); //
        }
    }
}
```

访问配置管理服务器的刷新地址，可以与git，bitbucket等代码工具结合，增加hooks，做到提交配置文件，自动触发刷新。

但是我们的bitbucket没有购买hooks这个功能，只能先手工触发refresh地址。

Atlassian Marketplace for Bitbucket

ⓘ Atlassian Marketplace for Bitbucket 5.16.0 has been disabled by your system administrator. Visit the [Atlassian Marketplace](#) website to continue your search or [re-enable the connection](#). ✕

[Settings](#)

The Universal Plugin Manager (v2.22.14) by Atlassian

本地调试

开发人员本地需要修改配置，可以在自己的开发分支中，添加application.yml文件。配置为本地启动所需的配置。然后将bootstrap.yml中的uri值清空。类似单元测试中的配置。

```
spring:
  application:
    name: sys
  cloud:
    config:
      uri:
```