

数字营销业务部 / ... / 配置中心接入

系统接入配置中心详细过程

由 张铁凡 创建, 最终由 lijingjing12 修改于 2015-08-04

注：以下接入环境为测试环境，上线时，请记得更改配置

一、配置中心配置导入（添加）

1. 1. 配置host

192.168.146.115 config.ads.jd.com

#192.168.146.116 config.ads.jd.com

1. 2. 访问config.ads.jd.com (admin/000000)

注意：不支持IE9及以下浏览器

a.项目管理—>添加项目

新建项目

项目编码：

客户端订阅时使用的编码

项目名称：

用户可读的名称

项目管理者：

admin

保存

b.首页-->选择上一步添加的项目-->development-->添加Module-->添加配置或者导入配置

首页 >> javademo - development

模块：全部

添加Module 删除Module

添加配置 导入配置 预览

二、应用接入配置中心

依赖spring的接入方式

1.处理项目中的jar冲突：

已知冲突zookeeper,配置中心依赖3.4.6版本，saf依赖3.4.5；
protobuf冲突；

2.加入依赖

<dependency>

<groupId>com.jd.ads.config</groupId>

<artifactId>client</artifactId>

<version>1.1.8-releases</version>

</dependency>

3.实现监听器接口（根据自己的业务特点，实现接收到变更通知后的逻辑）：

推荐方式：

```
public class ConfigurationListenerTest1 implements ProjcodeConfigurationListener {
    private String projCode;

    @Override
    public void configurationChanged(ConfigurationEvent event) {
        System.out.println("ConfigurationListenerTest2"+event.getProjCode()
        +" "+event.getType().name()
        +" " + event.getPropertyName() + " " + event.getPropertyValue());
    }
}
```

```

@Override
public String getProjCode() {
    return this.projCode;
}

@Override
public void setProjCode(String projCode) {
    this.projCode = projCode;
}
}

```

兼容之前的方式:

```

public class ConfigurationListenerTest implements ConfigurationListener {

    @Override
    public void configurationChanged(ConfigurationEvent event) {
        System.out.println("ConfigurationListenerTest"+event.getProjCode()
+ " "+event.getProfile()+ " "+event.getType().name()
+ " " + event.getPropertyName() + " " + event.getPropertyValue());
    }
}

```

4. 在Spring中配置:

<!--推荐配置方式-->

```

<bean class="com.jd.ads.config.client.PropertiesConfigurationFactoryBean">
    <constructor-arg index="0"
value="192.168.146.112:2181,192.168.146.113:2181,192.168.146.114:2181,192.168.146.115:2181,192.168.146.116:2181"/>
    <constructor-arg index="1">
        <list>
            <bean class="com.jd.ads.config.client.ConfigurationListenerTest1">
                <property name="projCode" value="xxxlog4j"/>
            </bean>
        </list>
    </constructor-arg>
    <constructor-arg index="2" value="development"/>
</bean>

```

<!--兼容之前的配置方式-->

```

<bean class="com.jd.ads.config.client.PropertiesConfigurationFactoryBean">
    <constructor-arg index="0"
value="192.168.146.112:2181,192.168.146.113:2181,192.168.146.114:2181,192.168.146.115:2181,192.168.146.116:2181"/>
    <constructor-arg index="1">
        <list>
            <bean class="com.jd.ads.config.client.ConfigurationListenerTest"/>
        </list>
    </constructor-arg>
    <constructor-arg index="2" value="development"/>
    <constructor-arg>
        <array>
            <value>xxxlog4j</value>
            <value>javademo</value>
        </array>
    </constructor-arg>

```

```
</constructor-arg>

</bean>
```

不依赖于**Spring**的接入方式

1.处理项目中的jar冲突:

2.加入依赖

3.实现监听器接口（根据自己的业务特点，实现接收到变更通知后的逻辑）：

注：前三点同上，此处不列出详细方式

4. 编写普通的Java类，连接配置中心

```
public class TestPropertiesConfiguration {

    public static void main(String[] args) throws Exception {

        ConfigurationListenerTest2 listenerTest2 = new ConfigurationListenerTest2();

        listenerTest2.setProjCode("javademo");

        List list = new ArrayList();

        list.add(listenerTest2);

        PropertiesConfiguration config = new
PropertiesConfiguration("192.168.146.112:2181,192.168.146.113:2181,192.168.146.114:2181",list,"development");

    }

}
```

注：应用上线时，需在访问配置中心的线上环境，导入（添加）配置,并修改应用中的配置

线上host:


172.16.164.47 config.ads.jd.com

#172.16.164.48 config.ads.jd.com

线上zk连接字符串（connectionString）：

172.16.164.46:2181,172.16.164.47:2181,172.16.164.49:2181,172.16.164.50:2181

订阅环境：production

 赞同 成为第一个赞同者