

# ZkConfig 使用说明

## 一、简介

ZkConfig 是为 zookeeper 开发的配置服务工具包,能与现有的 Java 系统进行良好的集成,也可以使用与非 java 系统以独立进程运行。提供与 spring 进行集成的插件。采用注解方式对需要动态更新的内存数据对象进行标注。

ZkConfig 用于解决在系统集群中配置文件的实时同步。当任意一台服务器配置文件发生变化的时候,所有集群服务器配置文件实现同步更新,并且在不启动 web 容器的情况下,实现内存配置对象的实时更新。

目前支持所有种类配置文件的同步更新,仅支持扩展名为.properties 与.cfg 结尾的键值对文件格式的内存数据对象实时同步。其余配置文件仅支持磁盘数据同步。

## 二、使用说明

### 使用步骤:

#### 1、增加虚拟机启动参数配置

运行 zkconfig 需要增加虚拟机运行参数 app.instance.config 的配置。app.instance.config 值为配置文件目录,将 zkconfig.properties 文件置于该目录下。

#### 2、为需要动态更新的配置类对象增加@ZkValue 注解

例:

```
//@ZkConfig(hosts = "127.0.0.1:2181")
public class TestObj {
```

```
    @ZkValue(key = "ip")
    private String ip;
    @ZkValue(key = "port")
    private int port;
    private String sth;
```

```
    //getter,setter 略
```

```
}
```

@ZkValue 注解中 key 值为配置 properties 文件中的 key 值,仅仅有此注解的属性会被实时更新。

@ZkConfig 注解与配置对象上,非必须,这里配置 zookeeper 服务器连接信息,此配置可以配置与 zkconfig.properties 文件中。

#### 3、配置 zkconfig.properties 文件

将 zkconfig.properties 文件复制到虚拟机运行参数 app.instance.config 所配置的目录下。

hosts=127.0.0.1:2181

Zookeeper 集群连接字符串,以逗号分隔 eg:127.0.0.1:2181,127.0.0.1:2182,127.0.0.1:2183

sessionTimeout=5000

会话超时时间

app\_name=cs

当前系统的应用名称，此项必填用于区分不同系统的配置文件

```
app_config_paths=${app.instance.config}
```

需要同步的系统配置文件目录集合，以逗号分隔 eg:/data/config/v-crm,/data/config/v-cs

```
dir_watch_period=3000
```

文件目录监控频率单位毫秒，用于监控配置文件的变化

#### 4、向 ZkConfig 注册内存配置对象实例（这里注册是实例，不是类型）

//创建注册中心

```
BeanRegisterCenter brc = new BeanRegisterCenter();
```

//注册实例，有多个就注册多个

```
brc.register(new TestObj());
```

此步骤非必须，如果不想更新内存数据，此步骤与步骤 2 都可以忽略。

#### 5、启动数据监听

//2、执行监听

```
MonitorType type = MonitorFactory.MonitorType.CONF_PROPERTIES_FILE_MONITOR;
```

```
ZkConfigExecutor.execute(type);
```

MonitorType 为监听类型枚举，目前支持通用文件类型监听以及 Properties 配置文件监听，后续可二次扩展。

示范代码（方式 1）：

```
/**
```

```
 * run local example
```

```
 * @param args
```

```
 */
```

```
public static void main(String[] args) {
```

```
    CountdownLatch connectedSignal = new CountdownLatch(1);
```

```
    try {
```

```
        //1、注册配置所需配置类
```

```
        BeanRegisterCenter brc = new BeanRegisterCenter();//执行监听
```

```
        brc.register(new TestObj());
```

```
        //2、执行监听
```

```
        MonitorType
```

```
        type
```

```
=
```

```
MonitorFactory.MonitorType.CONF_PROPERTIES_FILE_MONITOR;
```

```
        ZkConfigExecutor.execute(type);
```

```
        connectedSignal.await();
```

```
    } catch (Exception e) {
```

```
        e.printStackTrace();
```

```
    }
```

```
}
```

也可以采用方式二运行

```
public static void main(String[] args) throws IOException, InterruptedException, KeeperException {
```

```
    List<Object> confInstances = new ArrayList<Object>();
```

```
    confInstances.add(new TestObj());
```

```

        ZkConfigScannerConfigurer zk = new ZkConfigScannerConfigurer();
        zk.setMonitorType(2);
        zk.setConfInstances(confInstances);
        zk.startWatch();
    }
}

```

### 三、与 spring 集成

在 spring.xml 配置文件中加入如下配置即可实现集成：

```

<bean id="testConfig" class="com.lashou.cs.TestConfig"></bean>
<bean
    id="zkConfigScannerConfigurer"
    class="com.lashou.v.zkconfig.springplugin.ZkConfigScannerConfigurer"
    scope="singleton" lazy-init="false">
    <property name="confInstanceNames">
        <list>
            <value>testConfig</value>
        </list>
    </property>
    <property name="monitorType" value="2" />
</bean>

<bean id="testConfig" class="com.lashou.cs.TestConfig"></bean>

<bean id="zkConfigScannerConfigurer"
    class="com.lashou.v.zkconfig.springplugin.ZkConfigScannerConfigurer"
    scope="singleton" lazy-init="false">
    <property name="confInstanceNames">
        <list>
            <value>testConfig</value>
        </list>
    </property>
    <property name="monitorType" value="2" />
</bean>

```

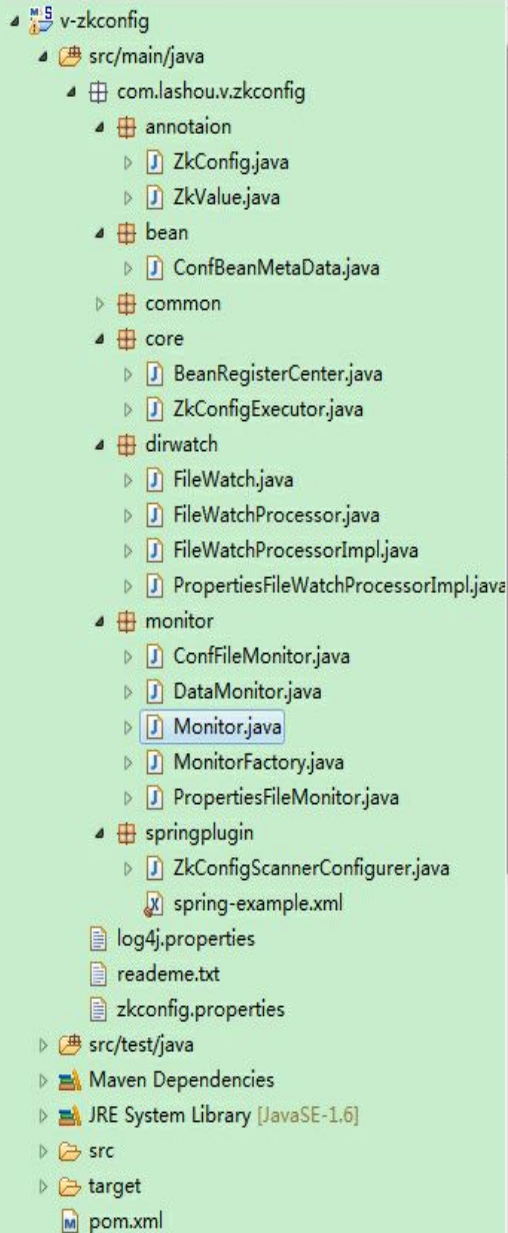
testConfig 为内存数据同步对象。confInstanceNames 为内存同步对象名称集合。monitorType 为监听文件类型，1：通用文件类型；2、properties 文件类型。

### 四、注意事项

在于 spring 集成时候，内存配置对象尽量采用配置文件的方式，因 spring 在类实例化时候会优先实例化配置文件，其次是注解方式。如果延迟实例化可能会导致无法获取的内存配置对象实例。

### 五、ZkConfig API 说明

## ZkConfig API设计



### common工具包

服务器连接工具  
文件处理工具  
系统初始化工具  
配置文件解析工具  
内存数据同步器

### 注解包

提供@ZkConfig @ZkValue注解支持

### 磁盘文件监控

通用文件监控  
Properties文件监控

### 实体对象包

配置属性元数据:  
ConfBeanMetaData

### ZK数据监控

监控工厂  
通用文件监控  
Properties文件监控

### 核心包

注册中心:  
BeanRegisterCenter  
启动器: ZkConfigExecutor