

# 当配置遇到Scala宏

又一个 Internal DSL 的应用案例

# 关于我



**Lunfu Zhong**

zhongl



wacai.com



HangZhou, China



[zhong.lunfu@gmail.com](mailto:zhong.lunfu@gmail.com)



<http://weibo.com/cafusic>

Scala, Java, Big Data, Message Middleware

这些年我们用过的配置  
都有哪些？

**.conf**

**.yaml**

**.properties**

**.xml**

相较而言...

# 醉人的命名长度

```
1 // application.properties
2 db1.jdbc.url = jdbc:mysql:10.0.0.1/demo
3 db1.jdbc.driver = com.mysql.jdbc.Driver
4 db1.jdbc.username = jushi
5 db1.jdbc.password = *****
```

# 啰嗦的对称包围

```
1 <!-- logback.xml -->
2 <configuration scanscan="true" scanPeriod="3 seconds" >
3
4     <appender name="STDOUT" class="ch.qos.logback.core.ConsoleAppender">
5         <encoder>
6             <pattern>%d{HH:mm:ss.SSS} [%thread] %-5level %logger{36} - %msg%n</pattern>
7         </encoder>
8     </appender>
9
10    <root level="INFO">
11        <appender-ref ref="STDOUT" />
12    </root>
13 </configuration>
```

# 优雅地近乎完美

```
1 #.travis.yml
2 language: scala
3 scala:
4     - 2.11.5
5 jdk:
6     - openjdk7
7 script:
8     - sbt ++$TRAVIS_SCALA_VERSION test
9 notifications:
10    email:
11        - jushi@wacai.com
```



# 强大到没有朋友

```
1 // kafka.conf
2 kafka {
3     server {
4         host = localhost
5         port = 9092
6     }
7     socket {
8         timeout = 5s
9         buffer = 64K
10    }
11    client = id
12    debug = false
13    delays = 2s
14 }
```

//@邓草原: Typesafe Config 不愧是最好的 Java/Scala 配置库。终于写好了一个 universal  
conf 文件，并且将长连接集群布到了树莓派集群上。

@邓草原

写了一天的配置文件了，想把各种配置归一，只靠一个 hostname 来分支。真痛苦。#最讨厌些配置#

3月12日 02:15 来自 微博 weibo.com

转发 5 | 评论 1 | 点赞 2

3月14日 09:31 来自 微博 weibo.com

只是...

# 我讨厌 G.E.T

```
1 // Consumer.scala
2 class Consumer extends Actor {
3   @inline def conf = context.system.settings.config
4
5   val client = new SimpleConsumer(
6     conf.getString("kafka.server.host"),
7     conf.getInt("kafka.server.port"),
8     conf.getDuration("kafka.socket.timeout", SECOND),
9     conf.getLong("kafka.socket.buffer"),
10    conf.getString("kafka.client")
11  )
12
13  ...
14 }
```



别跟我说话我想静静

然后，我想到了宏

Method or Annotation

# Method

```
1 def conf(key: String) = macro impl
2
3 def impl(c: Context)(key: c.Expr[String]) : c.Expr[Unit] = {
4   // TODO
5 }
6
7
8 val host = conf("kafka.server.host")
9
10 val host = conf.getString("kafka.server.host")
```



Method

# Annotation

```
1 class conf extends StaticAnnotation {
2   def macroTransform(annottees: Any*): Any = macro impl
3
4   private def impl(c: Context)(annottees: c.Expr[Any]*): c.Expr[Any] = {
5     import c.universe._
6
7     annottees.map(_.tree) match {
8       case ClassDef(...) :: Nil =>
9         // TODO
10
11       case _ =>
12         c.abort(c.enclosingPosition, "Invalid annottee")
13     }
14   }
15 }
```

# Annotation

```
1 @conf trait kafka {  
2     val server = new {  
3         val host: String  
4     }  
5 }  
6  
7 trait kafka {  
8     val server = new {  
9         val host: String = conf.getString("kafka.server.host")  
10    }  
11 }
```

如果仅仅只是映射装载  
就太 TYTSSN 了

# 由代码生成配置

```
1 @conf trait kafka {  
2     val server = new {  
3         val host = "wacai.com"  
4         val port = 12306  
5     }  
6  
7     val socket = new {  
8         val timeout = 3 seconds  
9         val buffer = 1024 * 64L  
10    }  
11  
12    val client = "wacai"  
13 }
```

```
1 kafka {  
2     server {  
3         host = wacai.com  
4         port = 12306  
5     }  
6  
7     socket {  
8         timeout = 3s  
9         buffer = 64K  
10    }  
11  
12    client = wacai  
13 }
```

# 之前提到的又一案例

```
1 import lego.dsl._
2
3 new Server {
4
5     def name = "ping-pong"
6
7     tcp.bind(port = 12306) {
8         Codec() <=> Handle {
9             case Request(_, hs, _) => Response(200, hs)
10        }
11    }
12 }
```



# 参阅

[github.com/wacai/config-annotation](https://github.com/wacai/config-annotation)

使用 Scala Macro Annotation 实现配置项绑定

[首篇](#)

[续篇](#)

[终篇](#)

What are macros good for ?



scala macro annotations  
real world example



[jushi@wacai.com](mailto:jushi@wacai.com)