

(/apps/redirect?  
utm\_source=side-  
banner-click)

# Scala中TypeTags和Manifests的用法



福克斯纪 (/u/13542edebea3) [+ 关注](#)

2015.08.26 07:53\* 字数 575 阅读 2219 评论 0 喜欢 8

(/u/13542edebea3)

Scala运行在JVM上，在JVM上有一种机制叫做类型擦除（type eraser）。类型擦除是说：在语言的编译阶段程序中所携带的泛型信息都会被擦除，最终生成的class文件中是不包含类型信息的。所以scala 2.8中引入了Manifests类来解决构建Array时遇到的一些问题。然而引入的Manifests并不完美，对于路径依赖类型，它拿到的信息是不精确的。所以后续引入了TypeTags。

展示一下各自的用法。看一个例子，利用manifest获取类型信息。

```
scala> class A[T]

scala> manifest[List[A[Int]]]
res1: Manifest[List[A[Int]]] = scala.collection.immutable.List[A[Int]]
```

manifest定义在Predef中，直接使用即可。与manifest功能类似的一个方法是classManifest，

```
scala> classManifest[List[A[Int]]]
warning: there was one deprecation warning; re-run with -deprecation for details
res2: ClassManifest[List[A[Int]]] = scala.collection.immutable.List[A[Int]]
```

两个方法都可以拿到相应传入的类型信息，但是观察返回结果：res1的类型是Manifest[List[A[Int]]]，res2的类型是ClassManifest[List[A[Int]]]，两个东西还是有一些区别。接着再看一个例子：

```
scala> manifest[List[A[_]]]
res3: scala.reflect.Manifest[List[A[_]]] = scala.collection.immutable.List[A[_ <: Any]]

scala> classManifest[List[A[_]]]
warning: there was one deprecation warning; re-run with -deprecation for details
res4: scala.reflect.ClassTag[List[A[_]]] = scala.collection.immutable.List[A[<?>]]
```

在这个例子中manifest会返回所有的类型信息，而classManifest返回了List[A]两层类型，丢失最内部的泛型信息。通过看源码注释对ClassManifest的解释，针对高阶类型它可能不会给出精确的类型信息。所以如果想要得到精确地类型信息使用manifest。

通常Manifest会以隐式参数和上下文绑定的形式使用。

```
def method[T](arg: T)(implicit m: Manifest[T]) = m //implicit parameter

def method[T: Manifest](arg: T) = m //context bound style
```

那么，有了Manifest，还要TypeTag干吗？



```
scala> class Foo { class Bar}
defined class Foo

scala> def m(f: Foo)(b: f.Bar)(implicit t: Manifest[f.Bar]) = t
m: (f: Foo)(b: f.Bar)(implicit t: scala.reflect.Manifest[f.Bar])scala.reflect.Manifest[f.Bar]

scala> val f = new Foo; val b = new f.Bar
f: Foo = Foo@6fe29d36
b: f.Bar = Foo$Bar@6bf7d9d

scala> val ff = new Foo; val bb = new ff.Bar
ff: Foo = Foo@2fdb85b8
bb: ff.Bar = Foo$Bar@75280b93

scala> m(f)(b)
res20: scala.reflect.Manifest[f.Bar] = Foo@6fe29d36.type#Foo$Bar

scala> m(ff)(bb)
res21: scala.reflect.Manifest[ff.Bar] = Foo@2fdb85b8.type#Foo$Bar

scala> res20 == res21 //not expected
res23: Boolean = true
```

(/apps/redirect?  
utm\_source=side-  
banner-click)

很明显f.Bar和ff.Bar是不相同的类型。所以，scala2.10 引入了TypeTags来解决上述问题，TypeTag可以看成是Manifest的升级版。

```
def m2(f: Foo)(b: f.Bar)(implicit ev: TypeTag[f.Bar]) = ev
val ev3 = m2(f1)(b1)
val ev4 = m2(f2)(b2)
ev3 == ev4 //false
```

和TypeTag一同引入的还有ClassTag和WeakTypeTag。ClassTag和ClassManifest一样，也不会返回完整的类型信息。但是它好classManifest还是有区别的。

```
scala> classTag[List[_]]
res9: scala.reflect.ClassTag[List[_]] = scala.collection.immutable.List

scala> classManifest[List[_]]
warning: there was one deprecation warning; re-run with -deprecation for details
res10: scala.reflect.ClassTag[List[_]] = scala.collection.immutable.List[<?>]

scala> classTag[List[List[_]]]
res11: scala.reflect.ClassTag[List[List[_]]] = scala.collection.immutable.List

scala> classManifest[List[List[_]]]
warning: there was one deprecation warning; re-run with -deprecation for details
res12: scala.reflect.ClassTag[List[List[_]]] = scala.collection.immutable.List[scala.collection.immutable.List]
```

classTag只拿到了基础类型信息，没有尝试去拿任何泛型信息，而classManifest拿到了所有已知的类型信息。WeakTypeTag相较TypeTag可以处理带有泛型的参数。看下边的例子：

```
scala> :pas
// Entering paste mode (ctrl-D to finish)

def weakParamInfo[T](x: T)(implicit tag: WeakTypeTag[T]): Unit = {
  val targs = tag.tpe match { case TypeRef(_, _, args) => args }
  println(s"type of $x has type arguments $targs")
}

// Exiting paste mode, now interpreting.

weakParamInfo: [T](x: T)(implicit tag: reflect.runtime.universe.WeakTypeTag[T])Unit

scala> def foo[T] = weakParamInfo(List[T]())
foo: [T] => Unit

scala> foo[Int]
type of List() has type arguments List(T)
```



参考文档：

typetags-manifests (https://link.jianshu.com?t=%5Bhttp://docs.scala-lang.org/overviews/reflection/typetags-manifests.html%5D(http://docs.scala-lang.org/overviews/reflection/typetags-manifests.html))  
scala what is a typetag and how do i use it (https://link.jianshu.com?t=%5Bhttp://stackoverflow.com/questions/12218641/scala-what-is-a-typetag-and-how-do-i-use-it#%5D(http://stackoverflow.com/questions/12218641/scala-what-is-a-typetag-and-how-do-i-use-it#))  
scala-type-system-manifest-vs-typetag (https://link.jianshu.com?t=%5Bhttp://hongjiang.info/scala-type-system-manifest-vs-typetag/%5D(http://hongjiang.info/scala-type-system-manifest-vs-typetag/))  
What-s-the-difference-between-ClassManifest-and-Manifest-td (https://link.jianshu.com?t=%5Bhttp://scala-programming-language.1934581.n4.nabble.com/What-s-the-difference-between-ClassManifest-and-Manifest-td2125122.html%5D(http://scala-programming-language.1934581.n4.nabble.com/What-s-the-difference-between-ClassManifest-and-Manifest-td2125122.html))

(/apps/redirect?utm\_source=side-banner-click)

如果觉得文章对您有用，或者有共鸣，请打赏一瓶加多宝。谢谢支持...

赞赏支持

程序世界 (/nb/161580) 举报文章 © 著作权归作者所有



福克斯纪 (/u/13542edebea3)

写了 64822 字，被 182 人关注，获得了 243 个喜欢 (/u/13542edebea3)


+ 关注

喜欢 | 8




更多分享

(http://cwb.assets.jianshu.io/notes/images/170334f



下载简书 App ▶

随时随地发现和创作内容



(/apps/redirect?utm\_source=note-bottom-click)



登录后发表评论 (/sign\_in?utm\_source=desktop&utm\_medium=not-signed-in-comment-form)

评论

智慧如你，不想发表一点想法 (/sign\_in?utm\_source=desktop&utm\_medium=not-signed-in-nocomments-text)咩~



被以下专题收入，发现更多相似内容



程序员 (/c/NEt52a?utm\_source=desktop&utm\_medium=notes-included-collection)



首页投稿 (暂停... (/c/bDHhpK?utm\_source=desktop&utm\_medium=notes-included-collection)

**Scala** Scala编程与实践 (/0f6ebeae2d34?utm\_source=desktop&utm\_medium=notes-included-collection)

(/apps/redirect?utm\_source=side-banner-click)

推荐阅读

更多精彩内容 > (/)

**乌托邦09|三世与君同 (/p/50be338321c5?utm\_campa...**

(/p/50be338321c5?

"嫂子，今天你和俊澜的大喜日子，不喝不行啊。"紫烟不会喝酒，我替她喝。"俊澜拦住他那帮同学。"还是我自己喝吧，看你，都快站不住了。"我心...

utm\_campaign=maleskine&utm\_content=note&utm

金银银环 (/u/e75b1b264f20?

utm\_campaign=maleskine&utm\_content=user&utm\_medium=pc\_all\_hots&utm\_source=recommendation)

**94后没背景也一路开挂，普通人如何复制逆袭之道 (/p/f...**

(/p/fb6721b73596?

一个没背景的萌妹子，却在高三和好友创业开平面设计公司，业务触及海外。一个多年没写文的人，9篇简书却2次被评为今日榜样，1次登上剿悍江湖。一...

utm\_campaign=maleskine&utm\_content=note&utm

仗剑执笔的红娘子 (/u/c39e3ebffa3?

utm\_campaign=maleskine&utm\_content=user&utm\_medium=pc\_all\_hots&utm\_source=recommendation)

**心有光芒 必有远芳——“龙学苑杯”教师岗位技能暨应用...**

(/p/f5c4c9ed9c01?

2018年5月2日，阳春三月绿草如茵，百花若蝶晴空向暖，在龙潭区教师进修学校二楼多功能教室，隆重召开“龙学苑杯”教师岗位技能暨应用综合大赛总决赛...

utm\_campaign=maleskine&utm\_content=note&utm

杨春英的小故事 (/u/c216aa071f71?

utm\_campaign=maleskine&utm\_content=user&utm\_medium=pc\_all\_hots&utm\_source=recommendation)

**胡歌12年后首谈车祸：既然活下来了，就不能白白活着...**

(/p/4dd8e8bf469f?

文/麦大人 01 胡歌又刷屏了。近日他上了《朗读者》，而这一期的主题是“生命”，他用磁性的嗓音，朗读了一段《哈姆雷特》中的经典独白，相当震撼：...

utm\_campaign=maleskine&utm\_content=note&utm

麦大人 (/u/2b3ad4f2a058?

utm\_campaign=maleskine&utm\_content=user&utm\_medium=pc\_all\_hots&utm\_source=recommendation)

**露天电影成为我童年快乐的记忆 (/p/f477911b1310?ut...**

(/p/f477911b1310?

今年五一假期，在东郡水乡游玩，白天与伙伴们下塘浑水摸鱼，晚上看了一场儿时看的那种露天电影，这两项活动将我尘封已久的童年记忆从内心深处涌出来...

utm\_campaign=maleskine&utm\_content=note&utm

张碧龙 (/u/f2a44f21dbf9?

utm\_campaign=maleskine&utm\_content=user&utm\_medium=pc\_all\_hots&utm\_source=recommendation)

**Android - 收藏集 (/p/dad51f6c9c4d?utm\_campaign=maleskine&utm\_c...**

用两张图告诉你，为什么你的 App 会卡顿？ - Android - 掘金 Cover 有什么料？从这篇文章中你能获得这些料：知道setContentView()之后发生了什么？... Android 获取 View 宽高的常用正确方式，避免为零 - 掘金...



passiontim (/u/e946d18f163c?


utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

**掘金 Android 文章精选合集 (/p/5ad013eb5364?utm\_campaign=maleski...**

用两张图告诉你，为什么你的 App 会卡顿？ - Android - 掘金 Cover 有什么料？从这篇文章中你能获得这些




料：知道setContentView()之后发生了什么？ ... Android 获取 View 宽高的常用正确方式，避免为零 - 掘金...

 掘金官方 (/u/5fc9b6410f4f?)

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)  
(/apps/redirect?utm\_source=side-banner-click)

## 掘金 Java 文章精选合集 (/p/893b21dce761?utm\_campaign=maleskine&...


Java 基础思维导图，让 Java 不再难懂 - 工具资源 - 掘金思维导图的好处 最近看了一些文章的思维导图，发现思维导图真是个强大的工具。了解了思维导图的作用之后，觉得把它运用到Java上应该是个不错的想法...

 掘金官方 (/u/5fc9b6410f4f?)

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

## 《编写高质量代码》学习笔记（2） (/p/bcd96ed136e4?utm\_campaign=m...


写着写着发现简书提醒我文章接近字数极限，建议我换一篇写了。建议52：推荐使用String直接量赋值 一般对象都是通过new关键字生成的，但是String还有第二种生成方式，也就是我们经常使用的直接声明方式，...

 我没有三颗心脏 (/u/a40d61a49221?)

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

## 漫谈Java的泛型机制 (/p/5a022eb2fbc4?utm\_campaign=maleskine&utm...

泛型是Java 1.5引入的新特性。泛型的本质是参数化类型，这种参数类型可以用在类、变量、接口和方法的创建中，分别称为泛型类、泛型变量、泛型接口、泛型方法。将集合声明参数化以及使用JDK提供的泛型...

 何时不晚 (/u/de8cc76a1f98?)


utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

(/p/30666e7ab8ae?)



utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)


## [灵异]骷髅玉（19） (/p/30666e7ab8ae?utm\_campaign=maleskine&utm...

 冰寒三尺 (/u/a59505f26884?)

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

## D89:昏沉、班级建设、乐斗 (/p/3b4359052e29?utm\_campaign=maleskin...

昨晚因为时间安排不得当，睡得比较晚，加上下午吃饭不规律和不科学，晚上睡眠质量也不高，所以早上就处于比较昏沉的状态，早点忘记吃，空腹上了两节课，这是对身体的极大不负责任，看来进入恶性循环是...

 小幸甫 (/u/b1a398f7491e?)

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)


(/p/76e77638142f?)



utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)

## 我爱你——讲述平安电销人的故事 (/p/76e77638142f?utm\_campaign=mal...

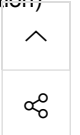
总有几分钟，其中的每一秒，你都愿意拿一年去换取。总有几颗泪，其中的每一次抽泣，你都愿意拿满手的承诺去代替。总有几段场景，其中的每幅画面，你都愿意拿全部的力量去铭记。——张嘉佳《从你的全世界路过》


 蒙初凝琪 (/u/d1293ed98793?)

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

## 雨 (/p/22cf0a344c72?utm\_campaign=maleskine&utm\_content=note&ut...


一整天，落雨没有完歇的节奏，不过，我的诗人朋友圈也因这场雨，一扫往日沉闷的气氛。他们纵论着古今中外雨中景、雨中事，白香山、李义山、晏几道、戴望舒、高尔基……让我觉得这个世界顿时空旷了许...



 洛阳慕岩子 (/u/4b633c19a074?utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

小木人 (/p/99bbc6fdebf6?utm\_campaign=maleskine&utm\_content=not...

小木人 经过它的第一眼，便由衷喜欢上了。小木人，形象似真人，关节各部都具有灵活转动性。普通的外表，不平凡的造姿。 小木人，多彩的人儿，小木人可塑不同的形象，动作，姿势，它可完成我不能施展的...

 师舆诗陈 (/u/b593cb3a5f80?utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

(/apps/redirect?utm\_source=side-banner-click)

