如果你不能接受我最差的一面，
毫无疑问你将不配得到最好的我.

—— 玛丽莲.梦露

聚石@taobao.com

https://github.com/zhongl

<u>Real-World Scala</u>

Get Started

Create Project

```
$ g8 typesafehub/scala-sbt
Scala Project Using sbt

organization [org.example]: me.zhongl
name [Scala Project]: demo
scala_version [2.9.2]:
version [0.1-SNAPSHOT]:

Template applied in ./demo
```

Project Structure

```
$ tree demo
demo
├── README
├── project
│   └── DemoBuild.scala
└── src
    └── main
        └── scala
            └── me
                └── zhongl
                    └── Demo.scala
```

Build Spec

```
import sbt._
import sbt.Keys._

object DemoBuild extends Build {

  lazy val demo = Project(
    id = "demo",
    base = file("."),
    settings = Project.defaultSettings ++ Seq(
      name := "demo",
      organization := "me.zhongl",
      version := "0.1-SNAPSHOT",
      scalaVersion := "2.9.2"
      // add other settings here
    )
  )
}
```

Hello, demo

```
$ sbt run
[info] Loading global plugins from ~/.sbt/plugins
[info] Loading project definition from ~/demo/project
[info] Set current project to demo (in build file:~demo/)
[info] Running me.zhongl.Demo
Hello, demo
[success] Total time: 0 s, completed 2013-5-24 9:38:47
```

IDE Plugins

```
$ cat ~/.sbt/plugins/build.sbt   // Global
addSbtPlugin("com.github.mpeltonen" % "sbt-idea" %
"1.2.0")

$ cat ~/demo/project/plugins.sbt // Project
addSbtPlugin("com.github.mpeltonen" % "sbt-idea" %
"1.2.0")

$ sbt gen-idea                   // Create IDE Files
```

Dependencies

```
settings = Project.defaultSettings ++ Seq(
  name := "demo",
  organization := "me.zhongl",
  version := "0.1-SNAPSHOT",
  scalaVersion := "2.9.2",
  libraryDependencies := Seq(
    "org.scala-lang" % "scala-library" % "2.9.2",
    "org.scalatest" %% "scalatest" % "1.7.2" % "test"
//  "org.scalatest" % "scalatest_2.9.2" % "1.7.2" % "test"
  )
)
```

Resolver

```
# ~/.sbt/local.sbt

resolvers <<= resolvers {rs =>
  val localMaven = "Local Maven Repository" at "file://"
+Path.userHome.absolutePath+"/.m2/repository"
  localMaven +: rs
}
```

Package

```
$ sbt package
$ sbt package-bin
$ sbt package-doc
$ sbt package-src
```

Publish

```
$ sbt publish           // central repos
$ sbt publish-local     // local repos
```

References

- giter8
- sbt
- sbt-idea
- sbteclipse
- nbsbt
- Typesafe Activator
- Scala Maven Plugin
- Buildr
- Gradle Scala Plugin

Behavior-Drive Development

Demo Spec

```
package me.zhongl

import org.scalatest.FunSpec
import org.scalatest.matchers.ShouldMatchers

class DemoSpec extends FunSpec with ShouldMatchers {
  describe("Demo") {
    it("should sum two integers") {
      Demo sum (1, 2) should be (3)
    }
  }
}
```

Continue Test

```
$ sbt
> ~ test
[info] Compiling 1 Scala source to ~/demo/target/scala-
2.9.2/test-classes...
[error] ~/demo/src/test/scala/me/zhongl/DemoSpec.scala:9:
value sum is not a member of object me.zhongl.Demo
[error]       Demo sum (1, 2) should be (3)
[error]            ^
[error] one error found
[error] (test:compile) Compilation failed
[error] Total time: 2 s, completed 2013-5-24 11:19:08
1. Waiting for source changes... (press enter to
interrupt)
```

Implement

```
package me.zhongl

object Demo extends App {
  println("Hello, demo")

  def sum(x: Int, y: Int) = x + y
}
```

Continue Test

```
[info] Compiling 1 Scala source to ~/demo/target/scala-
2.9.2/classes...
[info] DemoSpec:
[info] Demo
[info] - should sum two integers
[info] Passed: : Total 1, Failed 0, Errors 0, Passed 1,
Skipped 0
[success] Total time: 1 s, completed 2013-5-24 11:23:16
2. Waiting for source changes... (press enter to
interrupt)
```

Test Only

```
> test-only me.zhongl.DemoSpec
[info] DemoSpec:
[info] Demo
[info] - should sum two integers
[info] Passed: : Total 1, Failed 0, Errors 0, Passed 1,
Skipped 0
[success] Total time: 1 s, completed 2013-5-24 11:30:06
```

More Matchers

```
List(1, 2, 3) should have size (3)

"Scala" should startWith ("Sc")

Map("K" -> "V") should contain key ("K")

book should have ('title ("Programming in Scala"))

evaluating { assert(1 < 0) } should produce
[AssertionError]
```

References

- http://www.scalatest.org/ (备梯)
- http://etorreborre.github.io/specs2/
- https://code.google.com/p/scalacheck/
- http://scalamock.org/

Coverage

Scct plugin

```
# project/plugins.sbt
resolvers += Classpaths.typesafeResolver

resolvers += "scct-github-repository" at "http://mtkopone.
github.com/scct/maven-repo"

addSbtPlugin("reaktor" % "sbt-scct" % "0.2-SNAPSHOT")

# project/DemoBuild.scala

settings = Project.defaultSettings ++ Seq(
  id := "demo"
  ...
) ++ ScctPlugin.instrumentSettings
```

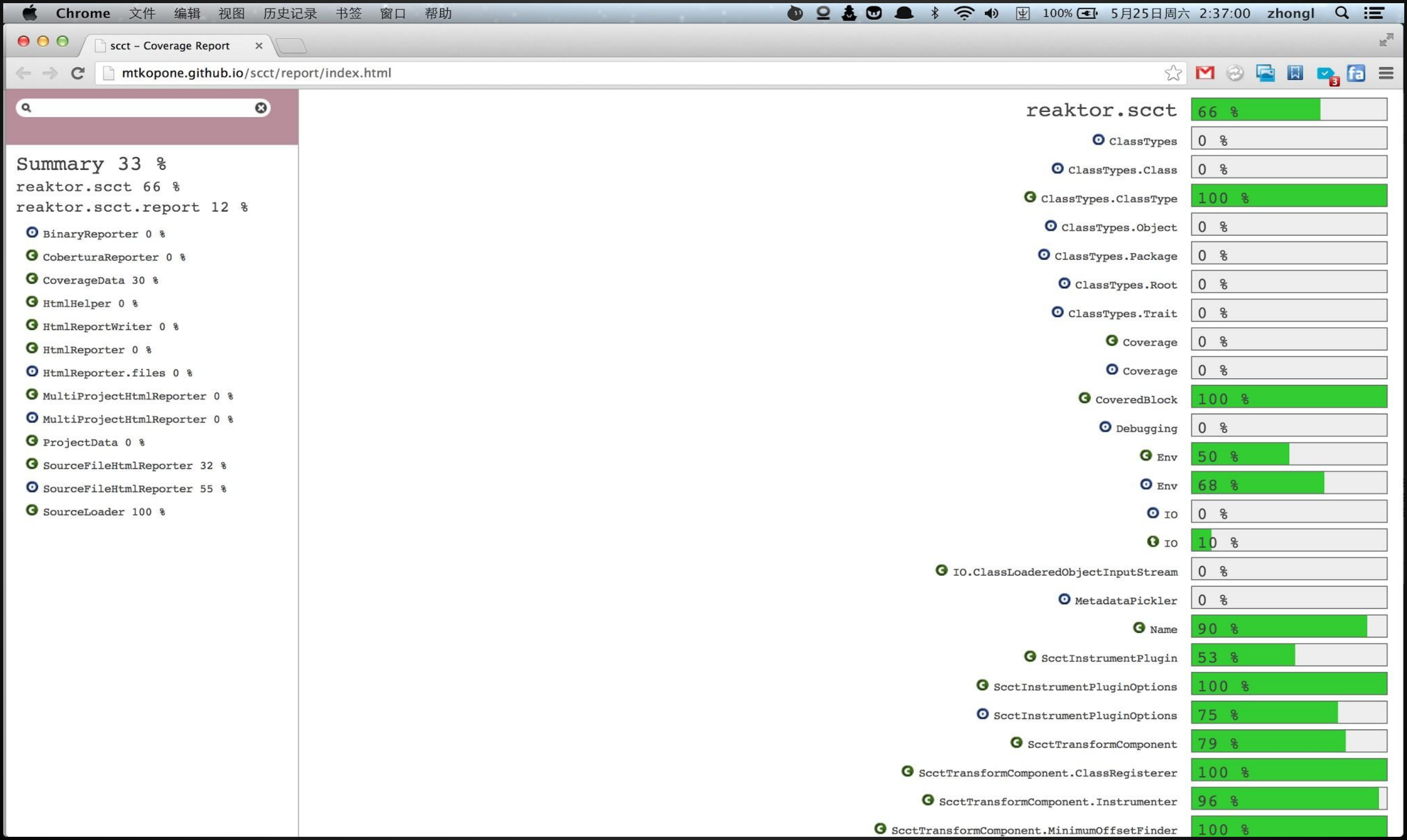
Scct plugin

```
$ sbt clean scct:test

$ sbt
> ;clean ;scct:test

# open
./target/scala_2.9.2/coverage-report/index.html
```

Scct plugin


Chrome 文件 编辑 视图 历史记录 书签 窗口 帮助
100% 5月25日周六 2:37:00 zhongl
scct – Coverage Report
mtkopone.github.io/scct/report/index.html

Summary 33 %
reaktor.scct 66 %
reaktor.scct.report 12 %

```
BinaryReporter 0 %
CoberturaReporter 0 %
CoverageData 30 %
HtmlHelper 0 %
HtmlReportWriter 0 %
HtmlReporter 0 %
HtmlReporter.files 0 %
MultiProjectHtmlReporter 0 %
MultiProjectHtmlReporter 0 %
ProjectData 0 %
SourceFileHtmlReporter 32 %
SourceFileHtmlReporter 55 %
SourceLoader 100 %
```

reaktor.scct	66 %
ClassTypes	0 %
ClassTypes.Class	0 %
ClassTypes.ClassType	100 %
ClassTypes.Object	0 %
ClassTypes.Package	0 %
ClassTypes.Root	0 %
ClassTypes.Trait	0 %
Coverage	0 %
Coverage	0 %
CoveredBlock	100 %
Debugging	0 %
Env	50 %
Env	68 %
IO	0 %
IO	10 %
IO.ClassLoaderedObjectInputStream	0 %
MetadataPickler	0 %
Name	90 %
ScctInstrumentPlugin	53 %
ScctInstrumentPluginOptions	100 %
ScctInstrumentPluginOptions	75 %
ScctTransformComponent	79 %
ScctTransformComponent.ClassRegisterer	100 %
ScctTransformComponent.Instrumenter	96 %
ScctTransformComponent.MinimumOffsetFinder	100 %

References

- http://mtkopone.github.io/scct/

Effective Scala

No statement

```
// Bad
def findPeopleIn(c: City, ps: Set[People]) = {
  val found = new mutable.HashSet[People]
  for (p <- ps) {
    for(a <- p.addresses) {
      if (a.city == c) found.put(p)
    }
  }
  return found
}
```

Be expression

```
// Good
def findPeopleIn(c: City, ps: Set[People]) = {
  for {
    p <- ps
    a <- p.addresses
    if a.city = c
  } yield p
}
```

Functional Magic

```
def firstPrimeGreatThan(num: Int): Int = {
  def prime(s: Stream[Int],f: Int => Boolean): Int =  s match
{
    case h #:: t if f(h) => h
    case h #:: t         => prime(t filter (_ % h > 0), f)
  }

  prime(Stream from 2, _ > num)
}

assert(firstPrimeGreatThan(20) == 23)
assert(firstPrimeGreatThan(100) == 101)
```

Use require

```
class Person(val name: String, val age: Int) {
  // Bad
  if (name == null || age <= 0)
    throw new IllegalArguemntException()

  // Good
  require(name != null, "name is required.")
  require(age > 0, "age should greater than zero.")
}
```

DSL

```
class Matcher(s: String) {
  def shouldMatch(regex: String) =
    require(s != null && s.matches(regex),
            "[" + s + "] should match " + regex)
}

implicit val str2Matcher = new Matcher(_:String)

class Person(val name: String, val age: Int) {
  name shouldMatch """\w+"""
}
```

Limit the scope of Implicits

References

- http://docs.scala-lang.org/
- http://twitter.github.io/effectivescala/
- http://twitter.github.io/scala_school/
- http://zh.scala-tour.com/
- http://stackoverflow.com/tags/scala/info
- https://github.com/languages/Scala

Books

- Scala for the Impatient (中文版)
- Programming Scala Tackle Multi-Core Complexity on the Java Virtual Machine(中文版)
- Scala in Depth (翻译中)
- Programming in Scala: A Comprehensive Step-by-Step Guide, 2nd Edition

Typesafe

@odersky
@jboner

LinkedIn amazon.com theguardian NHS

xerox SIEMENS tumblr. NASA CloudBees CSC

JUNIPER NETWORKS workday foursquare TOMTOM KLOUT

EDF TDC IGN UBS SONY livingsocial

Play

Akka

Slick

China Scala User Group

@邓草原
@fujohnwang
@hongjiang_wang

Thanks