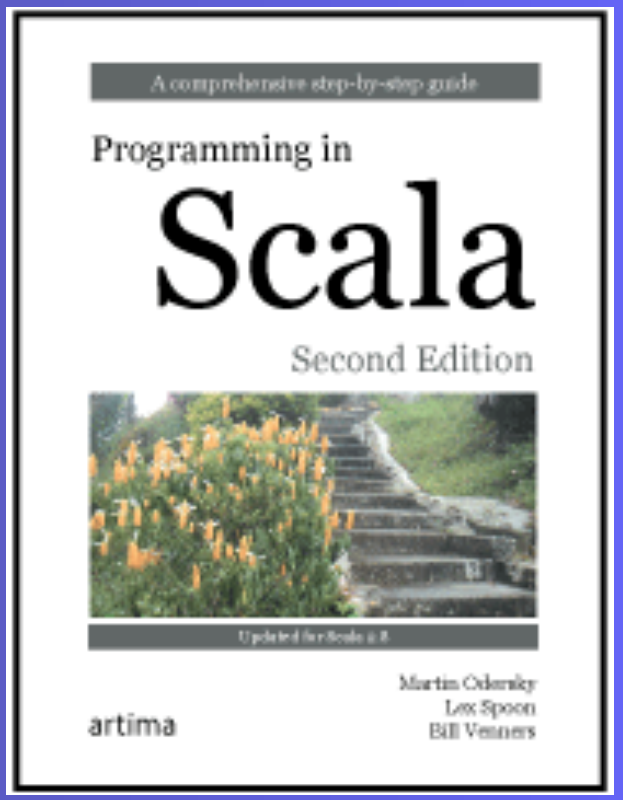
artima

Stairway to Scala - Flight 16

Build tools, web apps, integrating with Java

Bill Venners
Dick Wall



www.artima.com

Copyright (c) 2010 Artima Inc. All Rights Reserved.

Flight 16 goal

A high level tour of Build Tools, Web Application Frameworks, and Integration with Java

Maven

- Most universal tool support at present
- Fastest way to get a new Scala project

```
mvn archectype:generate
```

Check/update versions

```
mvn clean test
```

- Many archetypes, including web apps

Ant

```
<target name="init">
  <property name="scala-library.jar" value="${scala.home}/lib/scala-library.jar" />
  <path id="build.classpath">
    <pathelement location="${scala-library.jar}" />
    <pathelement location="${build.dir}" />
  </path>
  <taskdef resource="scala/tools/ant/antlib.xml">
    <classpath>
      <pathelement location="${scala.home}/lib/scala-compiler.jar" />
      <pathelement location="${scala-library.jar}" />
    </classpath>
  </taskdef>
</target>

<target name="compile" depends="init">
  <mkdir dir="${class.dir}" />
  <scalac srcdir="${src.dir}" destdir="${class.dir}" classpathref="build.classpath">
    <include name="**/*.scala" />
    <exclude name="test/**/*.scala" />
  </scalac>
</target>
```

SBT (Simple Build Tool)

- Written in Scala
- Fast Compile/Test, also Continuous
- Can integrate with Maven
- http://code.google.com/p/simple-build-tool/

```
dwall-MPRO:Koans dickwall$ sbt
Project does not exist, create new project? (y/N/s) y
Name: Flight13
Organization: com.escalatesoft
Version [1.0]:
Scala version [2.7.7]: 2.8.1
sbt version [0.7.4]:
....
```

Simple Build Tool (Continued)

- Need to add project file in: project/build/SomeName.scala

```
import sbt._

class SomeProject(info: ProjectInfo) extends DefaultProject(info) {
  val mavenLocal = "Local Maven Repository" at
    "file://"+Path.userHome+"/.m2/repository"
}
```

```
sbt update
sbt clean
sbt test
sbt ~test
```

Lift

- Functional Web Framework written by David Pollak
- http://liftweb.net
- Very complete (ORM, Templates, Components)
- Comet with Actors
- Data Binding
- Maven archetype - includes SBT support

Play

- RAD Web Framework with Scala support
- http://www.playframework.org/
- A little less radical than Lift
- Integration with major IDEs
- Early days, but promising

```
play install scala
play new SomeApp --with scala
play run SomeApp
```

- Not yet updated to Scala 2.8 final (still on RC7)

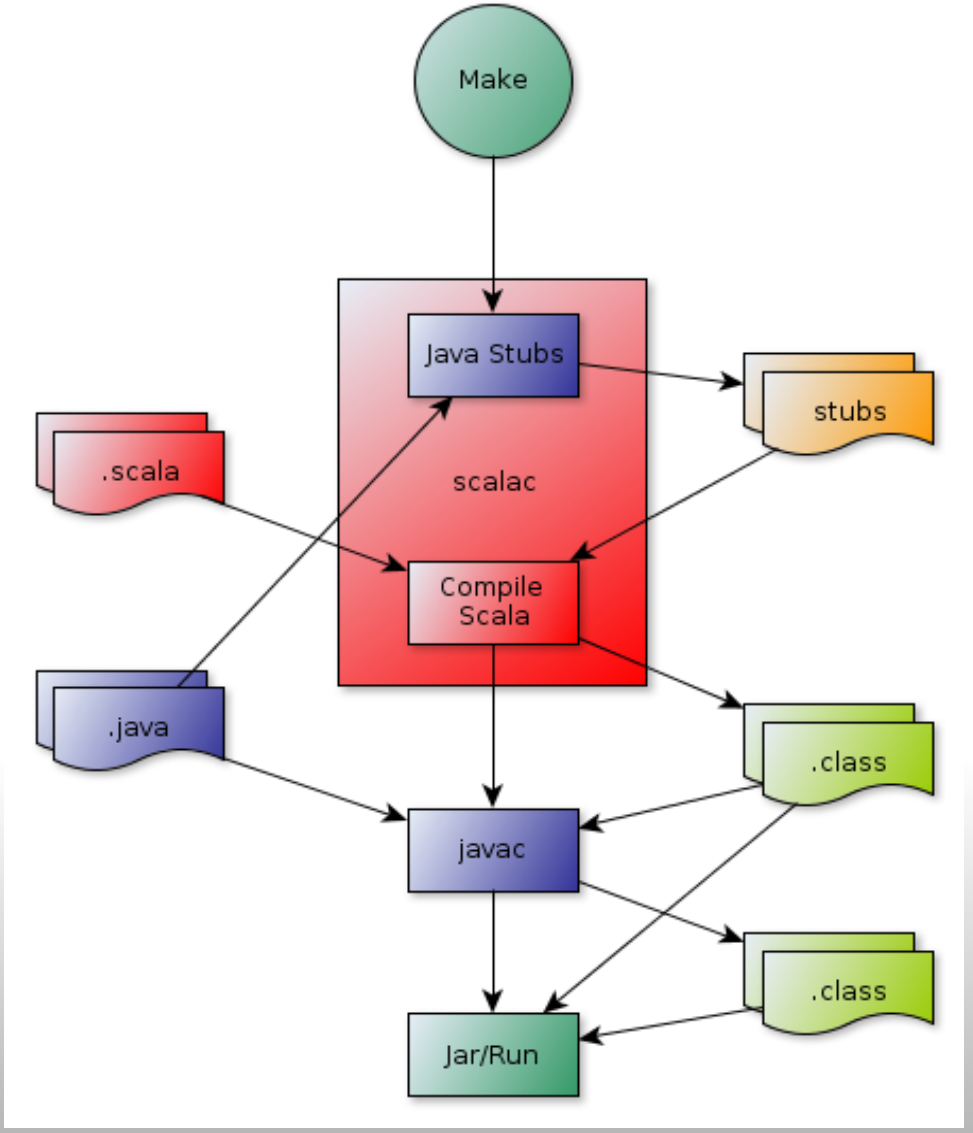
Wicket

- A popular Java web framework
- In Java -> many anonymous inner classes
- http://goo.gl/6arLs
- Less noise, but still wordy
- Good if you are familiar with Wicket already
- Relies on a lot of mutable state

Others

- Scala can use any Java library
- That includes (almost) any Java web framework
- But with mixed success
- A couple of other notables:
  - Scala/GWT - http://scalagwt.gogoego.com/
  - Scala/Vaadin - http://www.robertlally.com/blog/category/scala

Scala / Java Compile Cycle



Calling Java from Scala

- Import any Java library
- Call Java methods just like Scala
- Can leave off ()s for empty params
- Can call using infix notation
- Can extend or "with" Java interfaces
- Can instantiate Java classes
- Scala handles conversion to/from primitives

Nulls from Java

- Nulls discouraged in Scala

```
scala> val a = javaObj.methodCanReturnNull(x)

scala> a.toString                    // oops
java.lang.NullPointerException

scala> val b = Option(javaObj.methodCanReturnNull(x))

scala> b.toString                    // safe
None
```

Nulls to Java

- Methods that expect nulls?

```
scala> val a: Option[String] = Some("Hello")

scala> val b: Option[String] = None

scala> val r1 = javaObj.nullCapable(a.orNull)

scala> val r2 = javaObj.nullCapable(b.orNull)
```

Working with Java Collections

```
scala> val jl = new java.util.ArrayList[Int]
scala> jl.add(1); jl.add(2); jl.add(3)

scala> jl.map(_ * 2)
<console>:7: error: value map is not a member of java.util.
ArrayList[Int]
       jl.map(_ * 2)
          ^

scala> import scala.collection.JavaConversions._

scala> jl.map(_ * 2)
res1: scala.collection.mutable.Buffer[Int] = ArrayBuffer(2, 4, 6)
```

Implicit conversions not always enough?

```
// Java method signature:
public List<Integer> someJavaFunc(List<Integer> list) { ... }

scala> val l = List(1, 2, 3)
scala> val r1 = obj.someJavaFunc(l)
error: type mismatch;
found     : scala.List[Int]
required : java.util.ArrayList[java.lang.Integer]

scala> val jl = l.map( new java.lang.Integer(_) )
scala> val r2 = obj.someJavaFunc(jl)

(works)
```

Using Java Interfaces/Inner Classes

```
// java
public interface Predicate {
  boolean apply(o: Object);
}

scala> val isString = new Predicate {
     |   def apply(o: AnyRef): Boolean =
     |     o match {
     |       case s: String => true
     |       case _ => false
     |     }
     | }
scala> isString.apply("Hello")
res4: Boolean = true
```

Using Option from Java

```
// java

Option<String> something = Option.apply(it);
Option<String> nothing = Option.empty();

scalaObj.fnWithOptional(something);
scalaObj.fnWithOptional(nothing);
```

Using Scala Objects/Traits in Java

```
// scala

trait DoSomethingToString {
  def doIt(s: String): String
}

// Java

class Shout extends DoSomethingToString {
  public String doIt(String s) {
    return s.toUpperCase();
  }
}
```

General Advice

- Java calling Scala
  - Provide trait based API around Scala implementation
  - Avoid function literals
  - Convert between nullable and Option
- Scala calling Java
  - Remember scala.collection.JavaConversions
  - Use implicit conversions (respectfully)
  - Remember the REPL