

simple build tool

Getting started with sbt

sbt - what?

- :: de facto standard build tool for Scala projects
- :: open source
- :: written in Scala
- :: based on Apache Ivy
- :: current version 0.10
- :: many tools/plugins still use 0.7.x
- :: <http://code.google.com/p/simple-build-tool/> (0.7.7)
- :: <https://github.com/harrah/xsbt/> (0.10)

sbt - why?

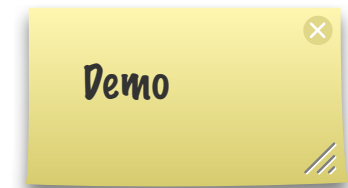
- :: cross compiling
- :: configuration, customization, and extension in Scala
- :: continuous compilation and testing
- :: Multiple project/subproject support
- :: Dependency management support
- :: Scala REPL with project classes and dependencies on the classpath
- :: supports testing with ScalaCheck, specs, scalatest, jUnit

Directory layout

:: Sources: same directory layout as Maven

```
src/  
  main/  
    resources/  
      <files to include in main jar here>  
    scala/  
      <main Scala sources>  
    java/  
      <main Java sources>  
  test/  
    resources  
      <files to include in test jar here>  
    scala/  
      <test Scala sources>  
    java/  
      <test Java sources>
```

sbt - HelloWorld



:: Following demos and examples use sbt 0.10

:: Demo

```
Terminal — bash — 140x40
Last login: Sun Jun 12 21:23:11 on ttys005
[michaelkober@mike-laptop-2 ~]$ cd
[michaelkober@mike-laptop-2 ~/Documents/code/workspaces]$ cd sbt-hello/
[michaelkober@mike-laptop-2 ~/Documents/code/workspaces/sbt-hello]$ sbt run
[info] Set current project to default (in build file:/Users/michaelkober/.sbt/plugins/)
[info] Set current project to default (in build file:/Users/michaelkober/Documents/code/workspaces/sbt-hello/)
[info] Running HelloWorld
Hello World!
[success] Total time: 0 s, completed Jun 12, 2011 9:34:50 PM
[michaelkober@mike-laptop-2 ~/Documents/code/workspaces/sbt-hello]$
```

Directory layout - continued

- :: manually managed dependencies

 - :: `lib`

- :: managed dependencies

 - :: `lib_managed` (optional in 0.10)

- :: generated files (classes, jars, doc,..)

 - :: `target`

- :: build configuration

 - :: `build.sbt`

```
project/  
  boot  
  build/  
    MyProject.scala  
    build.properties  
    plugins
```

build.sbt (Light config)

```
name := "HelloWorld"

version := "1.0"

organization := "com.jayway"

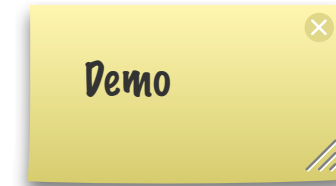
scalaVersion := "2.9.0"
```

- :: contains expression of type Settings[T]
- :: must be separated by blank lines

Using sbt - basic actions

:: Basic build actions

- :: clean
- :: update (done automatically in 0.10)
- :: compile
- :: run
- :: test
- :: doc
- :: package
- :: console



Using sbt - basic commands

:: Basic build commands

- :: sbt: starts interactive mode
- :: exit or quit: end current interactive mode
- :: projects: list available projects (in multiproject setup)
- :: project <projectid>: change current project
- :: actions: list available actions
- :: current: print current project and log level
- :: ~ <command>: execute action when sourcefile changed
- :: reload



Full configuration

- :: project definition in 'project' directory
- :: extend sbt.Build
- :: internal project is defined by creating an instance of Project
- :: example:

```
import sbt._

object MyBuild extends Build
{
  // Declare a project in the root directory of the build with ID "root".
  // Declare an execution dependency on sub1.
  lazy val root = Project("root", file(".")) aggregate(sub1)

  // Declare a project with ID 'sub1' in directory 'a'.
  // Declare a classpath dependency on sub2 in the 'test' configuration.
  lazy val sub1 = Project("sub1", file("a")) dependsOn(sub2 % "test")

  // Declare a project with ID 'sub2' in directory 'b'.
  // Declare a configuration dependency on the root project.
  lazy val sub2 = Project("sub2", file("b"), delegates = root :: Nil)
}
```

Adding repositories

:: Resolvers

- :: add maven style repositories like this

```
resolvers += "Scala-Tools Maven2 Snapshots" at "http://scala-tools.org/repo-snapshots"
```

:: Predefined

- :: DefaultMavenRepository <http://repo1.maven.org/maven2/> (included by default)
- :: ScalaToolsReleases <http://scala-tools.org/repo-releases/> (included by default)
- :: ScalaToolsSnapshots <http://scala-tools.org/repo-snapshots/>
- :: JavaNet1Repository <http://download.java.net/maven/1/>

```
resolvers += JavaNet1Repository
```



Adding dependencies

:: support for manual and automatic dependency management

:: libraryDependencies += groupId % artifactID % revision % configuration

```
libraryDependencies += Seq(  
  "org.apache.camel" % "camel-core" % "2.5.0" % compile,  
  "org.specs" % "specs" % "1.6.1" % test  
)
```

configuration for apache camel is equivalent to the following Maven2 snippet:

```
<dependency>  
  <groupId>org.apache.camel</groupId>  
  <artifactId>camel-core</artifactId>  
  <version>2.5.0</version>  
</dependency>
```



Plugins and Integration

:: sbt eclipse plugin - sbt 0.10

:: <https://github.com/typesafehub/sbteclipse>

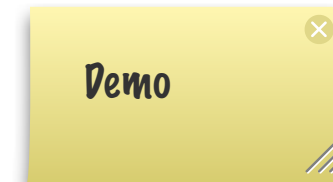
:: create eclipse project from sbt

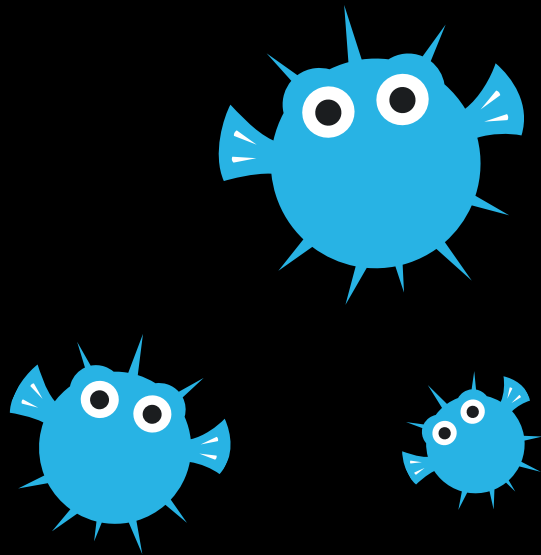
:: sbt idea sbt - 0.7.x (forks for 0.10 in progress)

:: <https://github.com/mpeltonen/sbt-idea>

:: plugin/processor to create project files for IntelliJ Idea

:: bnd4sbt, Android, scalariform, and more...

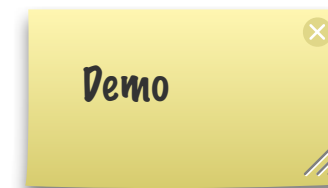




Create Actions

- :: construct task instance and assign it to a lazy val
- :: name for the task will be transformed from camel-case to dash seperated, i.e. MyCopyTask is transformed to my-copy-task
- :: val is required so that sbt can detect the task via reflection
- :: specify dependencies to other tasks with 'dependsOn'

Create Actions - examples



// example 1

```
lazy val sayHello = task { println("Hello sbt!"); None }
```

// example 2

```
def recursiveListFiles(f: File): Array[File] = {  
  val these = f.listFiles  
  these ++ these.filter(_.isDirectory).flatMap(recursiveListFiles)  
}
```

```
lazy val printDeps = task {  
  recursiveListFiles(new File("lib_managed")).filter(!_.isDirectory).foreach(println(_))  
  None  
} dependsOn(update) describedAs("print all dependencies to console")
```