SBT plugins

José Miguel Martínez Carrasco

Springer

May 9, 2012

What is a plugin?

A plugin extends the build definition, most commonly by adding new settings. The new settings could be new tasks.

Plugins provide a means of injecting and augmenting settings and commands. Since plugins are just Scala code, you can package and share plugins between projects.

What I like

Simplify your build chain with code that can be used in different projects.

Get the resources to build plugins that meet your needs and share them.

- Create *your project* folder
- build.sbt
- project/build.properties
- project/plugins.sbt
- Scala class extending Plugin class.

build.sbt

```
1   sbtPlugin := true
2
3   name := "sbt-xslt-plugin"
4
5   organization := "com.jm2dev"
6
7   version := "0.1.6"
8
9   scalaVersion := "2.9.1"
10
11  seq(scriptedSettings: _*)
12
13  libraryDependencies ++= Seq(
14    "org.scalatest" %% "scalatest" % "1.6.1" % "test",
15    "net.sf.saxon" % "Saxon-HE" % "9.4"
16  )
```

project/build.properties

```
1    sbt.version=0.11.2
```

project/plugins.sbt

```
1  libraryDependencies <+= (sbtVersion) { (version) =>
2    "org.scala-tools.sbt" %% "scripted-plugin" % version
3  }
4
5  resolvers ++= Seq (
6    "coda" at "http://repo.codahale.com"
7  )
```

src/main/scala/MyPlugin.scala

```
1   import sbt._
2   import Keys._
3
4   object MyPlugin extends Plugin
5   {
6     override lazy val settings = Seq(commands +=
          myCommand)
7
8     lazy val myCommand =
9       Command.command("hello") { (state: State) =>
10        println("Hi!")
11        state
12      }
13  }
```

idea

Generates Intellij Idea project configuration.

plugins.sbt

```
1  resolvers += "sbt-idea-repo" at
       "http://mpeltonen.github.com/maven/"
2
3  addSbtPlugin("com.github.mpeltonen" % "sbt-idea" %
       "1.0.0")
```

usage

```
1  gen-idea
2
3  gen-idea no-classifiers
```

assembly

Generates a big jar.

Web application with servlet container embedded.

plugins.sbt

```
addSbtPlugin("com.eed3si9n" % "sbt-assembly" %
    "0.7.2")
```

usage

```
assembly
```

web

Develop web applications easily.

plugins.sbt

```
libraryDependencies <+= sbtVersion(v =>
    "com.github.siasia" %% "xsbt-web-plugin" %
    (v+"-0.2.11"))
```

build.sbt

```
seq(webSettings :_*)

libraryDependencies += "org.mortbay.jetty" % "jetty" %
                        % "6.1.22" % "container"
```

usage

```
~;container:start; container:reload /

container:stop
```

implicitly

The SBT Organization is available for use by any SBT plugin.

Developers who contribute their plugins into the community organization will still retain control over their repository and its access. The Goal of the SBT organization is to organize SBT software into one central location.

A side benefit to using the SBT organization for projects is that you can use gh-pages to host websites in the http://scala-sbt.org domain.

Community Ivy repository

Typesafe, Inc. has provided a freely available Ivy Repository for SBT projects to make use of.

If you would like to publish your project to this Ivy repository, first contact Joshua.Suereth@typesafe.com and request privileges (we have to verify code ownership, rights to publish, etc.). After which, you can deploy your plugins using the following configuration:

You'll also need to add your credentials somewhere. I use a /.sbt/sbtpluginpublish.sbt file

build.sbt

```
publishTo := Some(Resolver.url("sbt-plugin-releases",
  new URL("http://scalasbt.artifactoryonline.com/scalasbt/
  sbt-plugin-releases/"))(Resolver.ivyStylePatterns))

publishMavenStyle := false
```

sbtpluginpublish.sbt

```
credentials += Credentials("Artifactory Realm",
    "scalasbt.artifactoryonline.com", "jsuereth", "@my
    encrypted password@")
```

plugins.sbt

```
addSbtPlugin("me.lessis" % "ls-sbt" % "0.1.1")

resolvers ++= Seq(
  "less is" at "http://repo.lessis.me",
  "coda" at "http://repo.codahale.com"
)
```

plugins.sbt

```
seq(lsSettings: _*)
```

usage

```
  curl -X POST http://ls.implicit.ly/api/1/libraries \
-d 'user=your-gh-user \
&repo=your-gh-repo \
&version=version-to-sync'
```

Reusability.

Convention over configuration.

Simplicity.

- https://github.com/harrah/xsbt/wiki/Plugins-Best-Practices
- http://tihlde.org/ eivindw/latex-listings-for-scala/
- https://github.com/harrah/xsbt/wiki/Plugins