

# The Last 10 Percent

Stefan Zeiger, Senior Software Engineer  
04/04/2017



## The Ninety-Ninety Rule

"The first 90 percent of the code accounts for the first 90 percent of the development time. The remaining 10 percent of the code accounts for the other 90 percent of the development time."

— Tom Cargill, [Bell Labs](#)

## Overview

- **Corollary: 90% of conference talks are about the first 90%**
- **We'll take 90% code and turn it into a 100% open source library**

# Overview

- The difficult part is not *how* to do it (follow instructions, copy & paste) but *what* to do
- Opinionated approach



Where do artifacts come from?

```
[info] Loading project definition from /home/travis/build/slick/slick/project
[info] Updating {file:/home/travis/build/slick/slick/project/}slick-build...
[info] Resolving com.typesafe#sbt-mima-plugin;0.1.14 ...
[info] Resolving com.typesafe#mima-reporter_2.10;0.1.14 ...
[info] Resolving org.scala-lang#scala-library;2.10.6 ...
[info] Resolving com.typesafe#mima-core_2.10;0.1.14 ...
[info] Resolving org.scala-lang#scala-compiler;2.10.6 ...
[info] Resolving org.scala-lang#scala-reflect;2.10.6 ...
[info] Resolving com.typesafe#config;1.0.0 ...
[info] Resolving com.typesafe.sbt#sbt-osgi;0.7.0 ...
[info] Resolving biz.aQute.bnd#bndlib;2.1.0 ...
[info] Resolving org.osgi#org.osgi.core;4.2.0 ...
[info] Resolving com.typesafe#sbt-sdlc;0.1 ...
[info] Resolving org.jsoup#jsoup;1.7.3 ...
[info] Resolving de.johoop#sbt-testng-plugin;3.0.2 ...
[info] Resolving com.novocode#sbt-ornate;0.4 ...
[info] Resolving com.novocode#sbt-ornate;0.4 ...
[info] Resolving org.scala-sbt#sbt;0.13.13 ...
[info] Resolving org.scala-sbt#main;0.13.13 ...
...
```



## Where do artifacts come from?

- **Maven Central** (Sonatype Nexus)
- **JCenter** (JFrog Artifactory / BinTray)
- **Your local repository** (~/.ivy2/local)
  
- **Can be changed in local sbt setup** (e.g. use repository proxy server) **or project** (add more repositories)

# sbt.boot.properties in sbt-launcher.jar

[repositories]

local

jcenter: <https://jcenter.bintray.com/>

typesafe-ivy-releases:

<https://repo.typesafe.com/typesafe/ivy-releases/>, ↵

[organization]/[module]/[revision]/[type]s/[artifact] ↵

(-[classifier]).[ext], bootOnly

maven-central





## left-pad in the node.js ecosystem

- [https://www.theregister.co.uk/2016/03/23/npm\\_left\\_pad\\_chaos/](https://www.theregister.co.uk/2016/03/23/npm_left_pad_chaos/)
- Tiny library (one function: left-pad)
- Author deleted it from NPM
- Chaos ensued



Let's Write *leftPad* in Scala

(The First 90 Percent)

```
object LeftPad {  
  
  def leftPad(s: String,  
              len: Int,  
              ch: Char = ' '): String =  
    ch.toString * (len-s.length) + s  
}
```

# Project Set-Up

## Building & Testing

- Command line first
- Use sbt
- Test framework of your choice
- I use JUnit

## Sharing

- README.md (documentation!)
- LICENSE (<https://opensource.org/licenses>)
- github (<https://github.com>)
- Issue Tracker
- Pull Requests
- Wiki? Problematic because not versioned!
  - Stick to README.md or use Ornate (<https://szeiger.github.io/ornate-doc/>)

Publish to Maven Central

# Publishing

- <http://www.scala-sbt.org/release/docs/Using-Sonatype.html>
- <http://www.scala-sbt.org/sbt-pgp/usage.html>
- Coordinates
- Signing
- Metadata
  
- Version Tag





# Continuous Integration / PR Validation

## Travis CI

- <https://docs.travis-ci.com/user/getting-started>
- Build matrix vs custom script

# Multi-Project Build

## Multi-Project Build

- sbt plugins

<http://www.scala-sbt.org/0.13/docs/Plugins.html>

- Advanced cross-versioning with sbt-doge

<https://github.com/sbt/sbt-doge>



# Bintray Publishing

- Publishing sbt plugins

<http://www.scala-sbt.org/0.13/docs/Bintray-For-Plugins.html>

- sbt-bintray

<https://github.com/sbt/sbt-bintray>

- Caveat: <https://github.com/sbt/sbt-bintray/issues/58>



# Tag-Based Publishing

## Travis CI

- <https://docs.travis-ci.com/user/encrypting-files/>
- Install Travis CI command-line tool (see above)
- Simplified scripts based on <https://github.com/scala/scala-parallel-collections/tree/master/admin>

Next Steps?



## Suggestions

- Add MiMa for automatic binary compatibility checks (<https://github.com/typesafehub/migration-manager>)
- Automatic publishing of generated manual and scaladocs in github pages
- Scaladoc link checking for generated manual (<https://github.com/typesafehub/sbt-sdlc>)

Links

# Skills

- **git**

<https://git-scm.com/doc>

- **sbt**

<http://www.scala-sbt.org/documentation.html>

- **Unix shell scripts**

<https://www.shellscript.sh/>

- **Markdown**

<https://guides.github.com/features/mastering-markdown/>

## Accounts

- github
- Sonatype OSS Repository Hosting  
(see <http://central.sonatype.org/pages/ossrh-guide.html> for how to sign up)
- Travis CI (via github account; see <https://docs.travis-ci.com/user/getting-started/>)
- BinTray



- **Sample Project:**

<https://github.com/szeiger/leftpad-template>

- **Twitter: @StefanZeiger**

