sbt Scala build tool

CONCEPTS

Start sbt once and stay in its shell: later commands skip the JVM startup and reuse compiled state.

build.sbt settings and tasks
↓ reload
sbt shell keeps state warm
↓ compile / test
target/ classes and reports

Tasks take a scope prefix: Test/compile compiles test sources, subproj/test runs one subproject.

RUN AND BUILD

sbt
Start the interactive shell

sbt *<command>*
Run one command and exit (batch mode)

compile
Compile main sources

run *<args>*
Run the main class in sbt's JVM

runMain *<class>* *<args>*
Run a specific main class

clean
Delete target directories

Test/compile
Compile test sources

package
Build the binary jar

doc
Generate Scaladoc

update
Resolve dependencies

exit
Leave the shell (or Ctrl+D)

help *<command>*
Show help for a command

TESTING

test
Run all tests

testOnly *<Suite>*
Run named suites, * works

TESTING (cont.)

testOnly *<Suite>* **--** **-z** "*<text>*"
Run tests matching a name

testQuick
Run new, failed or affected tests

WATCH MODE

~compile
Recompile on source change

~test
Re-run tests on change

~testQuick
Re-run only affected tests

~run
Restart the app on change

Enter
Stop a ~ watch and return to shell

PROJECTS

projects
List all subprojects

project *<id>*
Switch the current project

*<id>***/test**
Run a task in one subproject

*<id>***/compile**
Compile one subproject

project /
Switch back to the root project

clean test
Run tasks one after another

; *<a>* **;** *<b>*
Run b only if a succeeds

BUILD DEFINITION

reload
Reload build files after editing

set *<key>* **:=** *<value>*
Change a setting until reload

show *<key>*
Print a setting or task value

inspect *<key>*
Show scope, value and dependencies

tasks
List available tasks

CROSS BUILDING AND REPL

console
Scala REPL with project classpath

consoleQuick
REPL without compiling the project

+test
Run for every crossScalaVersions entry

++*<version>* **test**
Switch Scala version for one command

crossScalaVersions
Show the versions used by +

PUBLISH AND OPTIONS

publishLocal
Publish to the local Ivy repository

publish
Publish to the publishTo repository

dependencyTree
Print the dependency tree

sbt -J-Xmx2G
Pass a JVM option to sbt

BUILD.SBT BASICS

Minimal build.sbt

```
ThisBuild / scalaVersion :=
"3.3.4"

lazy val root = (project in
file("."))
  .settings(name := "app")
```

Add a dependency

```
libraryDependencies +=
"org.scalatest" %% "scalatest" %
"3.2.19" % Test
```

Operators

Operator	Meaning
:=	Set a value
+=	Append one element
++=	Append a collection
%	Group % artifact % version
%%	Add the Scala suffix to the artifact

RECIPES

Fast TDD loop on one suite

```
sbt
~testOnly com.example.UserSpec
```