

*Jimin Hsieh*

---

# IntelliJ with Scala

<https://gitter.im/jiminhsieh>  
<https://twitter.com/JiminHsieh>

---



---

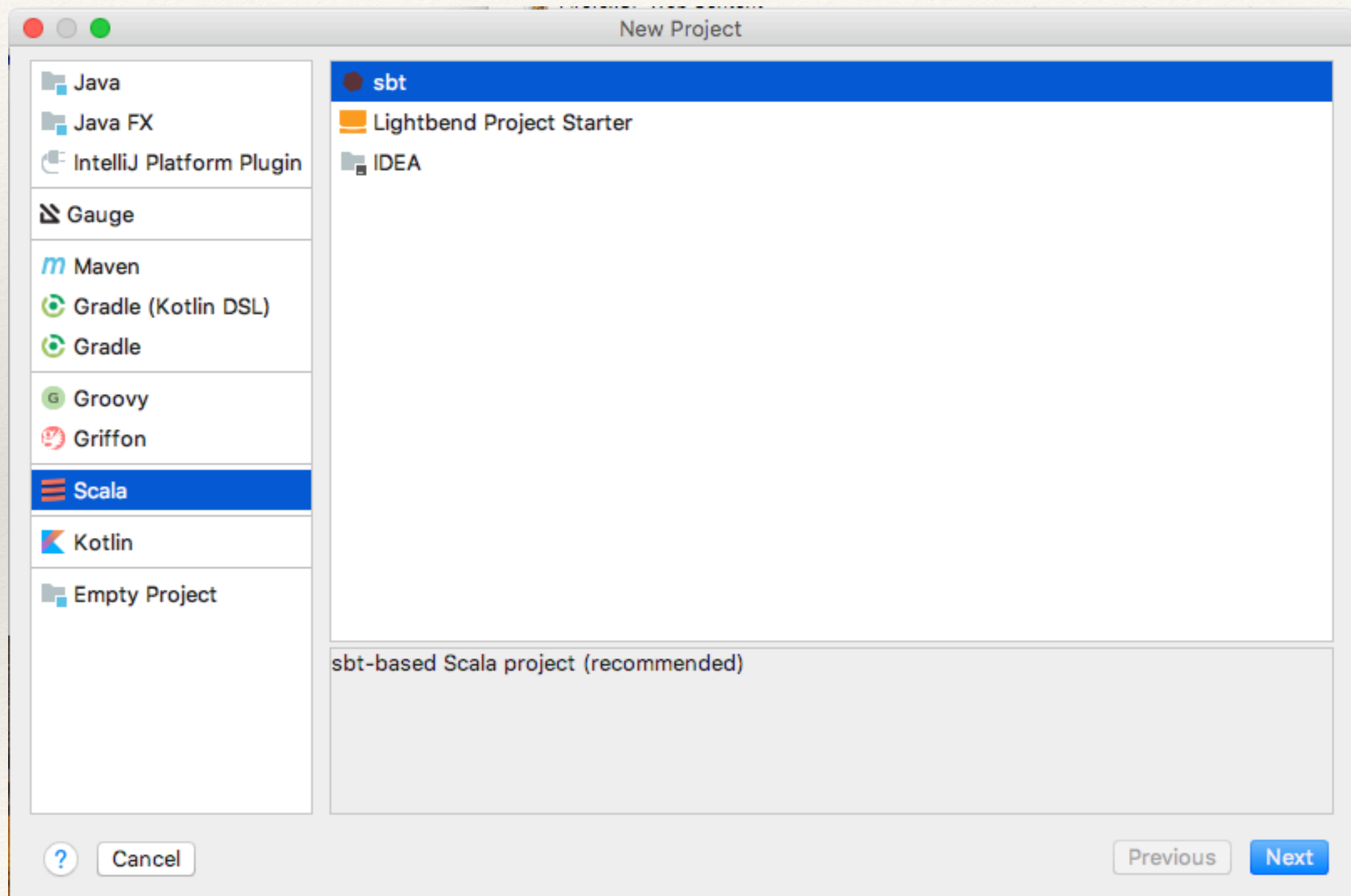
# Agenda

---

- ❖ Create new project
- ❖ Import new project
- ❖ Basic operation with IntelliJ
- ❖ How to development?
- ❖ Trace & Debug with IntelliJ



# Create new project - 01



# Create New Project - 02

**New Project**

Name:

Location:  ...

JDK:  ... New...

sbt:  ... ☒ Sources

Scala:  ...  ... ☒ Sources

▼ More Settings

Module name:

Content root:  ...

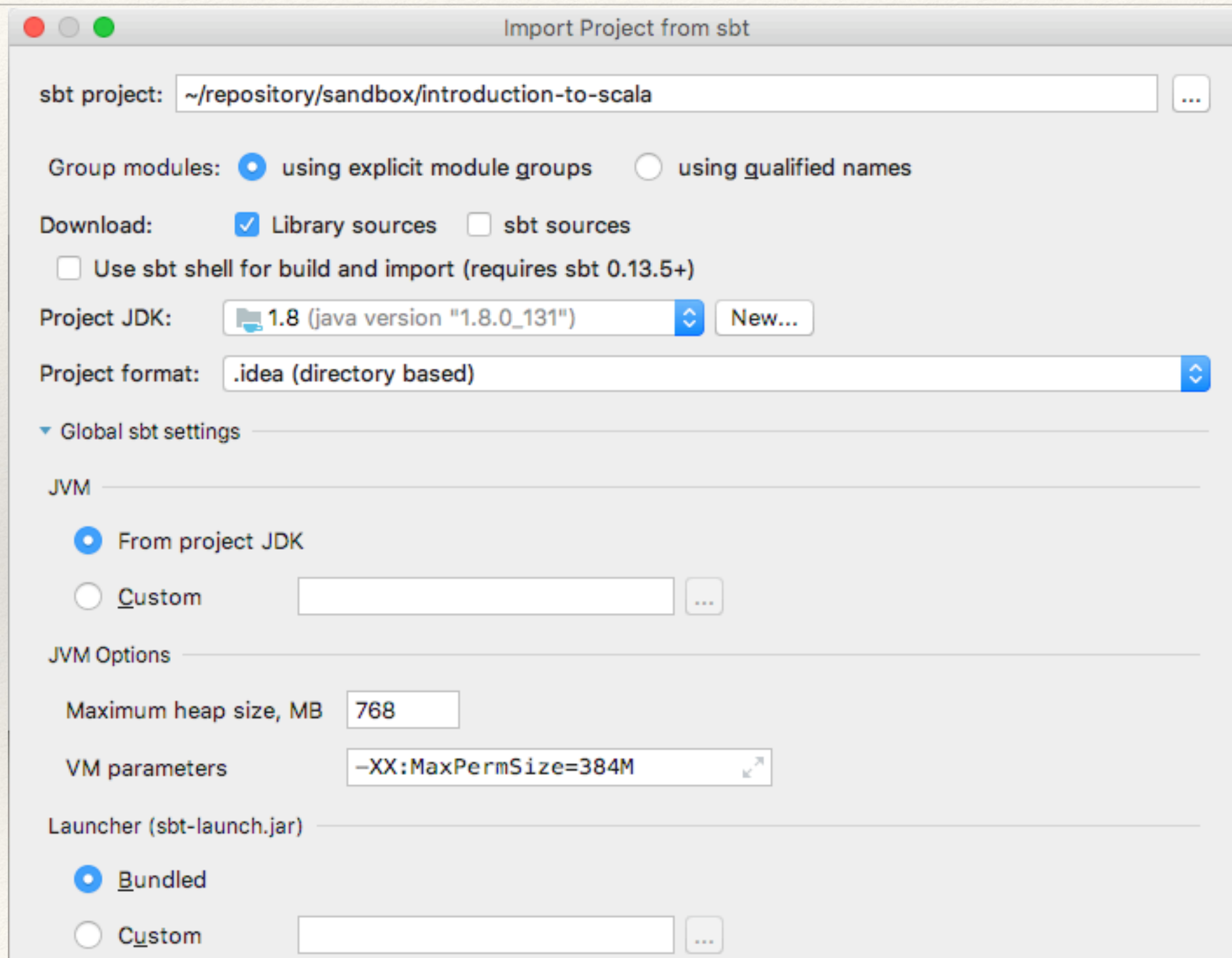
Module file location:  ...

Project format:  ...

? Cancel Previous Finish



# Import new project



The screenshot shows the 'Import Project from sbt' dialog box. The 'sbt project' field is set to '~/repository/sandbox/introduction-to-scala'. Under 'Group modules', 'using explicit module groups' is selected. Under 'Download', 'Library sources' is checked. The 'Project JDK' is set to '1.8 (java version "1.8.0\_131")'. The 'Project format' is set to '.idea (directory based)'. The 'Global sbt settings' section is expanded, showing 'JVM' settings where 'From project JDK' is selected. 'JVM Options' are set to '-XX:MaxPermSize=384M'. The 'Launcher (sbt-launch.jar)' is set to 'Bundled'.

Import Project from sbt

sbt project: ~/repository/sandbox/introduction-to-scala

Group modules: ☒ using explicit module groups ☐ using qualified names

Download: ☒ Library sources ☐ sbt sources

☐ Use sbt shell for build and import (requires sbt 0.13.5+)

Project JDK: 1.8 (java version "1.8.0\_131") New...

Project format: .idea (directory based)

▼ Global sbt settings

JVM

☒ From project JDK

☐ Custom

JVM Options

Maximum heap size, MB 768

VM parameters -XX:MaxPermSize=384M

Launcher (sbt-launch.jar)

☒ Bundled

☐ Custom



# Basic Operation with IntelliJ

	macOS	Win
Show type	[Ctrl][Shift][p]	[Alt][Equals]
Build project	[⌘][F9]	[Ctrl][Shift][F9]
Find Class	[⌘][o]	[Ctrl][n]
Find File	[⌘][Shift][o]	[Ctrl][Shift][n]
Search current text	[⌘][f]	[Ctrl][f]
Search global text	[⌘][Shift][f]	[Ctrl][Shift][f]
Line Comment	[⌘][/]	[Ctrl][/]
Block Comment	[⌘][Shift][/]	[Ctrl][Shift][/]



---

# How to development?

---

- ❖ Large project or project with advanced Scala features
  - ❖ IntelliJ as an advanced editor.
    - ❖ Be aware that syntax highlight in IntelliJ doesn't always correct.
  - ❖ **Terminal with sbt to build/compile and test.**
- ❖ Without using advanced Scala features
  - ❖ IntelliJ is fine for most use cases.



---

# Trace & Debug with IntelliJ

---

- ❖ Breakpoint
  - ❖ Step into (F7)
  - ❖ Step over (F8)
  - ❖ Step out (Shift + F8)



---

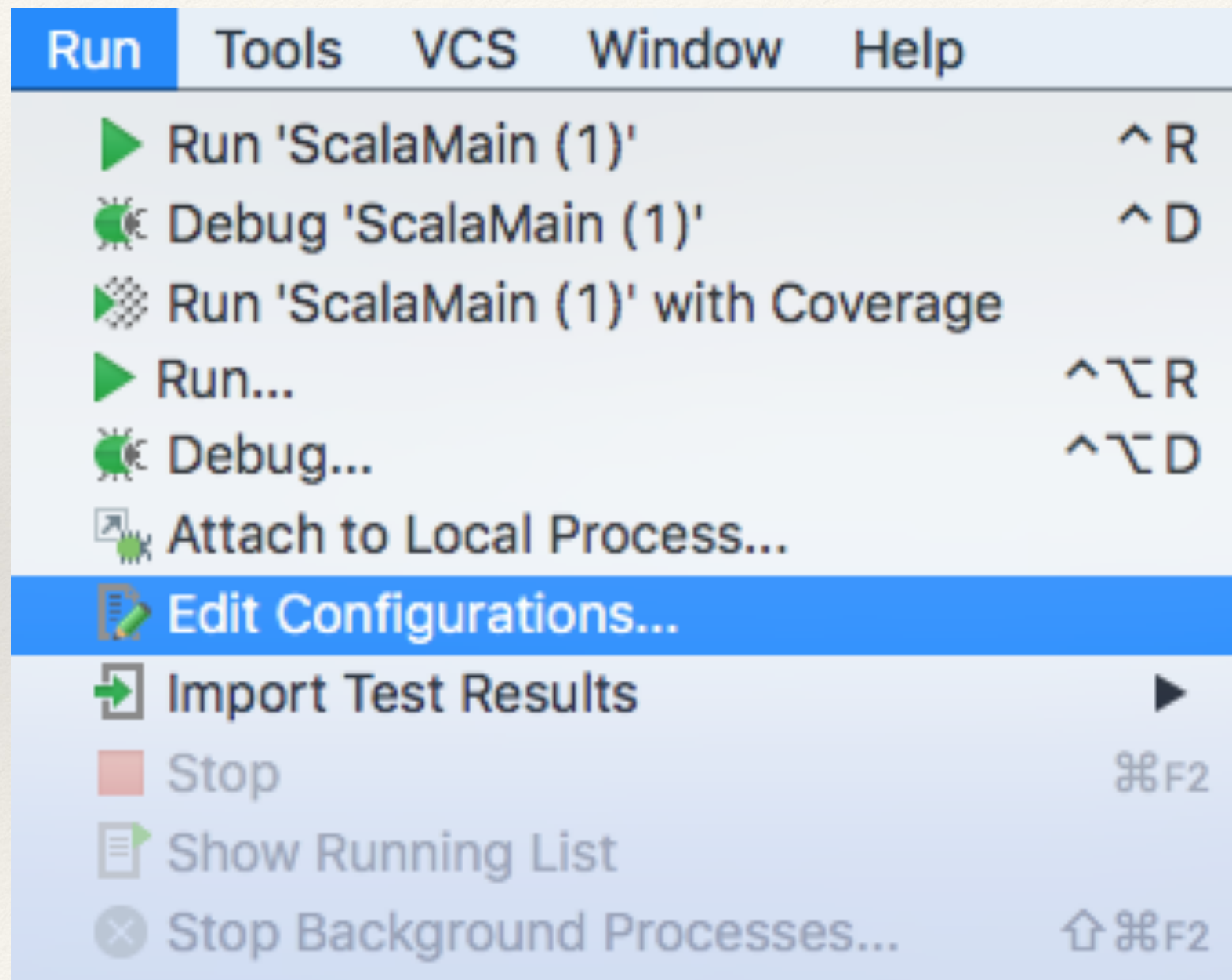
# Remote Debug

---

- ❖ Why remote debug?
  - ❖ When we do not start application through IntelliJ.
  - ❖ For instance, we start application through sbt.



# How to debug remotely?





# How to debug remotely?

The screenshot shows the 'Add New Configuration' dialog in IntelliJ IDEA. The 'Remote' option is selected in the left sidebar. The main panel is titled 'Configuration' and shows settings for remote JVM debugging.

**Name:** Unnamed ☐ Share ☐ Single instance only

**Configuration** | **Logs**

Command line arguments for running remote JVM

`-agentlib:jdwp=transport=dt_socket,server=y,suspend=n,address=5005`

For JDK 1.4.x

`-Xdebug -Xrunjdwp:transport=dt_socket,server=y,suspend=n,address=5005`

For JDK 1.3.x or earlier

`-Xnoagent -Djava.compiler=NONE -Xdebug  
-Xrunjdwp:transport=dt_socket,server=y,suspend=n,address=5005`

**Settings**

Transport: ☒ Socket ☐ Shared memory

Debugger mode: ☒ Attach ☐ Listen

Host:  Port:

Search sources using module's classpath:



---

# sbt debug port

---

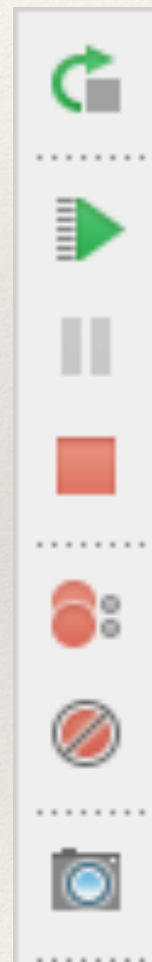
- ❖ Application and sbt are in the same process (default).
  - ❖ `$ sbt -jvm-debug 5005`
- ❖ Application and sbt are different process.
  - ❖ Pass JVM options to forked process.
    - ❖ `-agentlib:jdwp=transport=dt_socket,server=y,suspend=n,address=5005`



---

# IntelliJ Debugger

---





---

# Reference

---

❖ IntelliJ Debugging