EPFL

scalacenter

Multiple files support

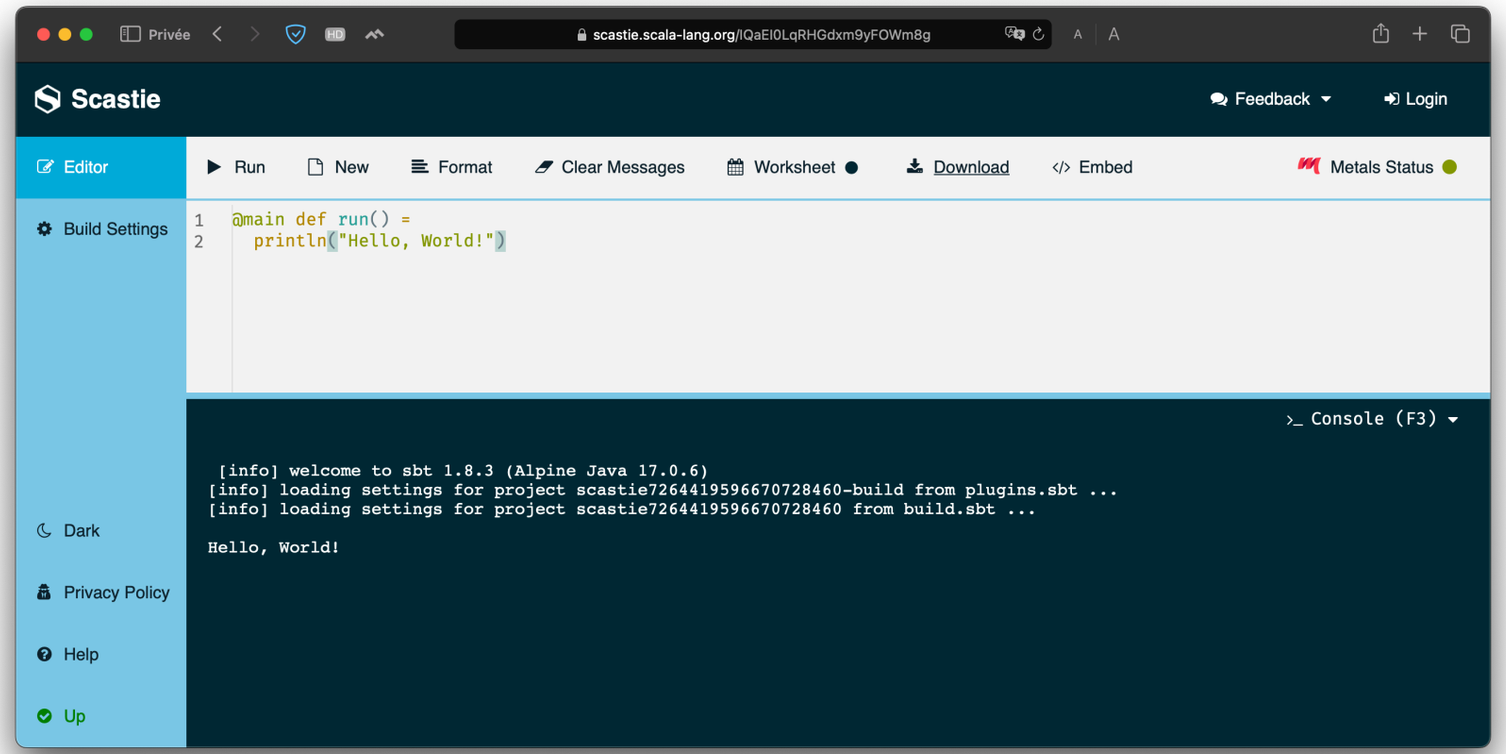
DAVID KALAJDZIC

Timeline

- Project definition
- Challenges
- Front end
- Backend
- Evaluation
- Conclusion & future work

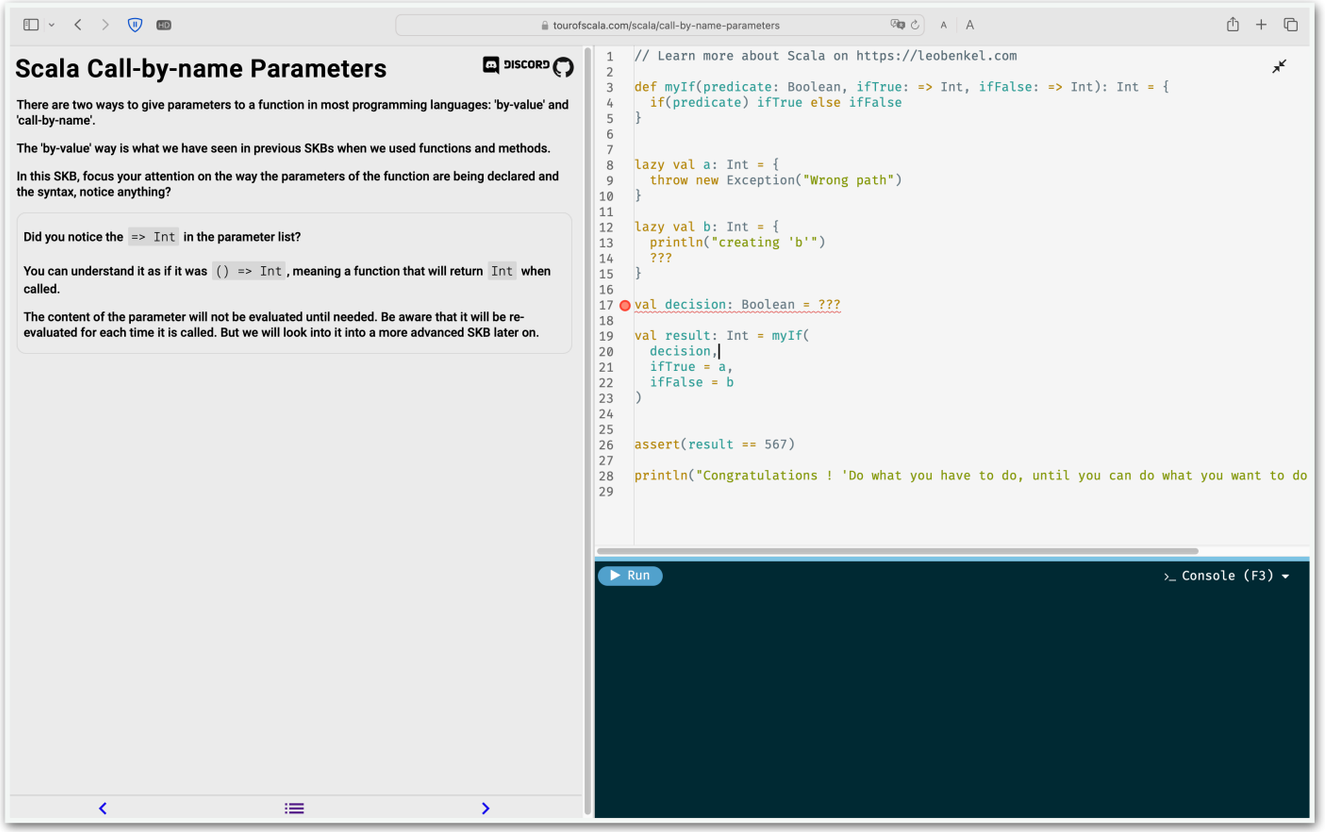
What is Scastie?

- https://scastie.scala-lang.org/U6hM9iXwT0KiAMH7DJJXmQ
- https://github.com/scalacenter/scastie



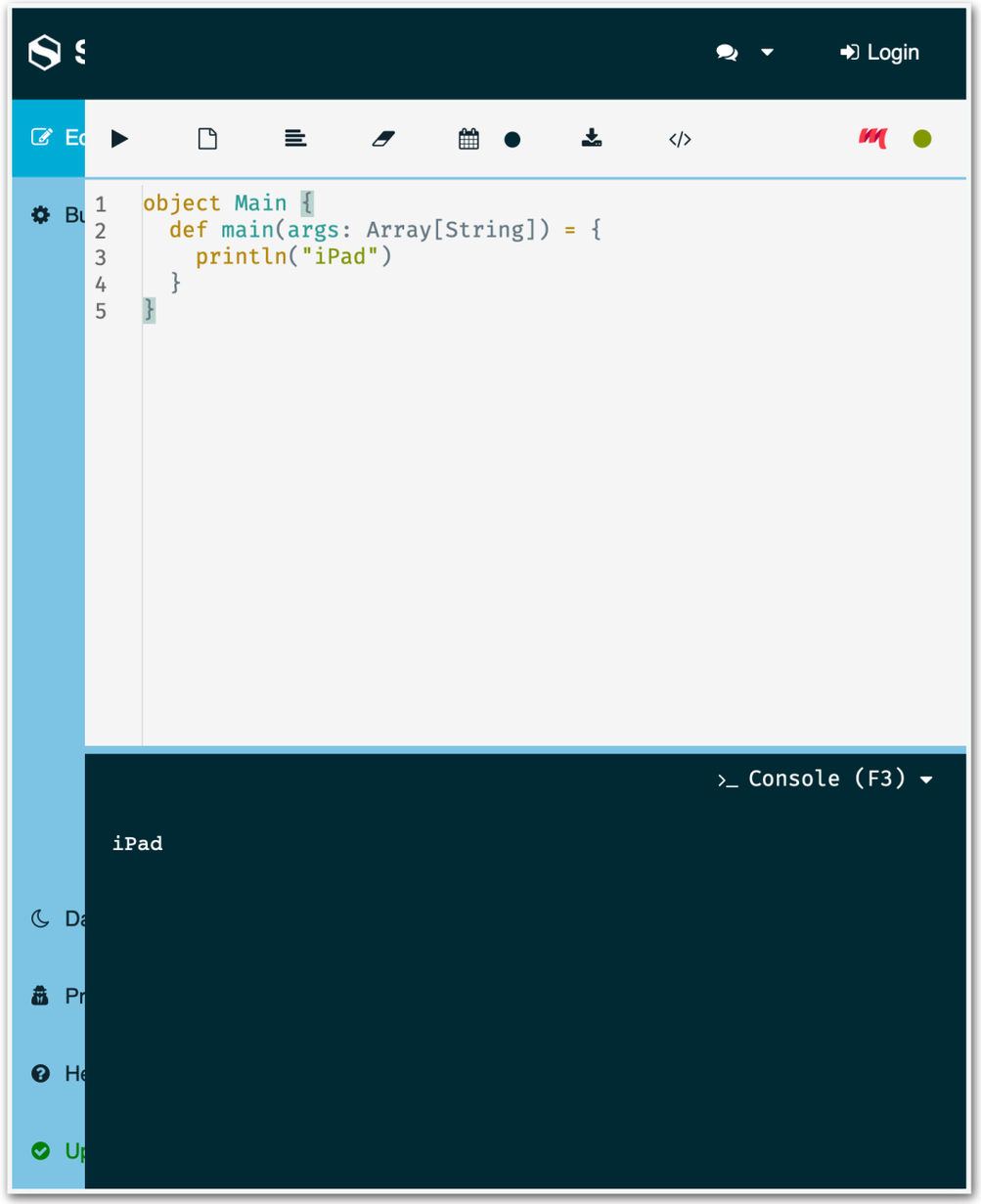
Current main features

- Embed feature



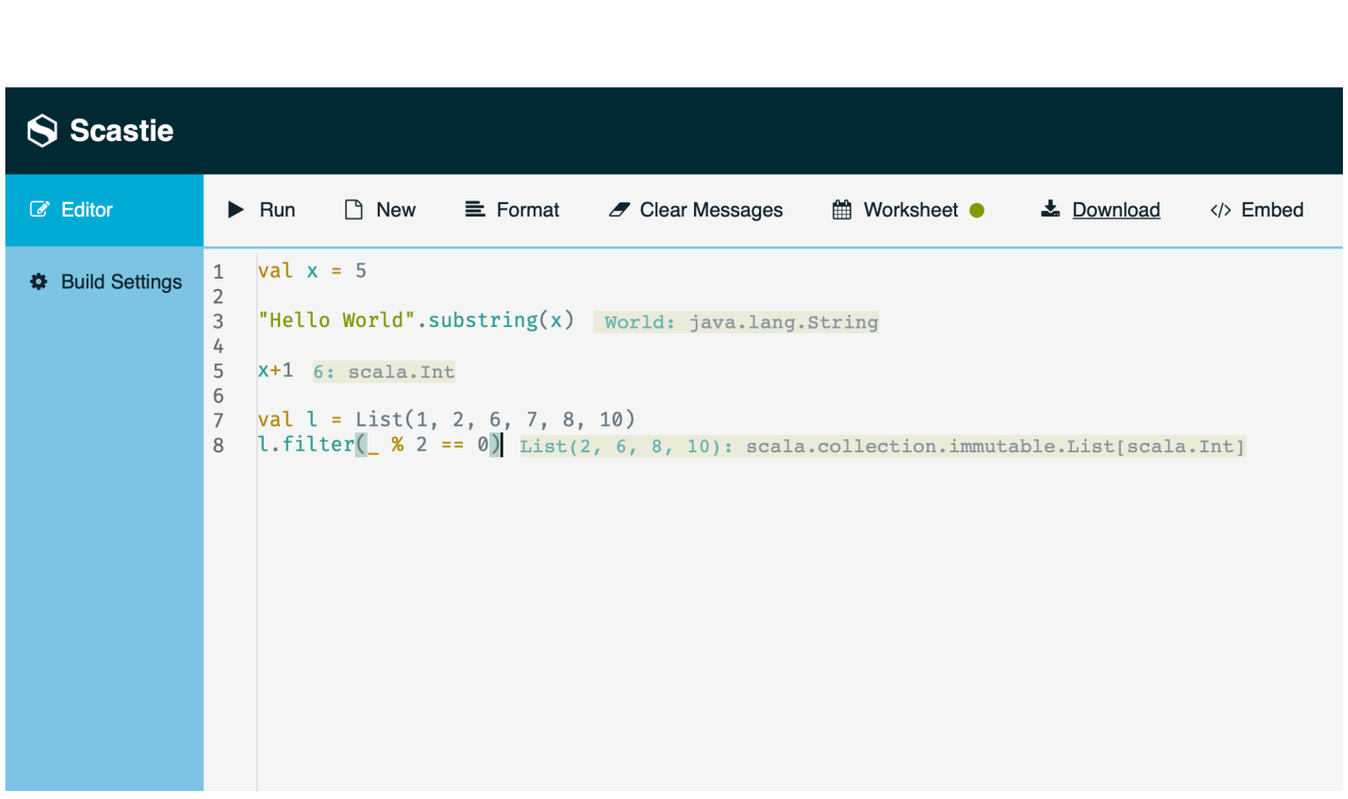
Current main features

- Mobile/Tablet modes

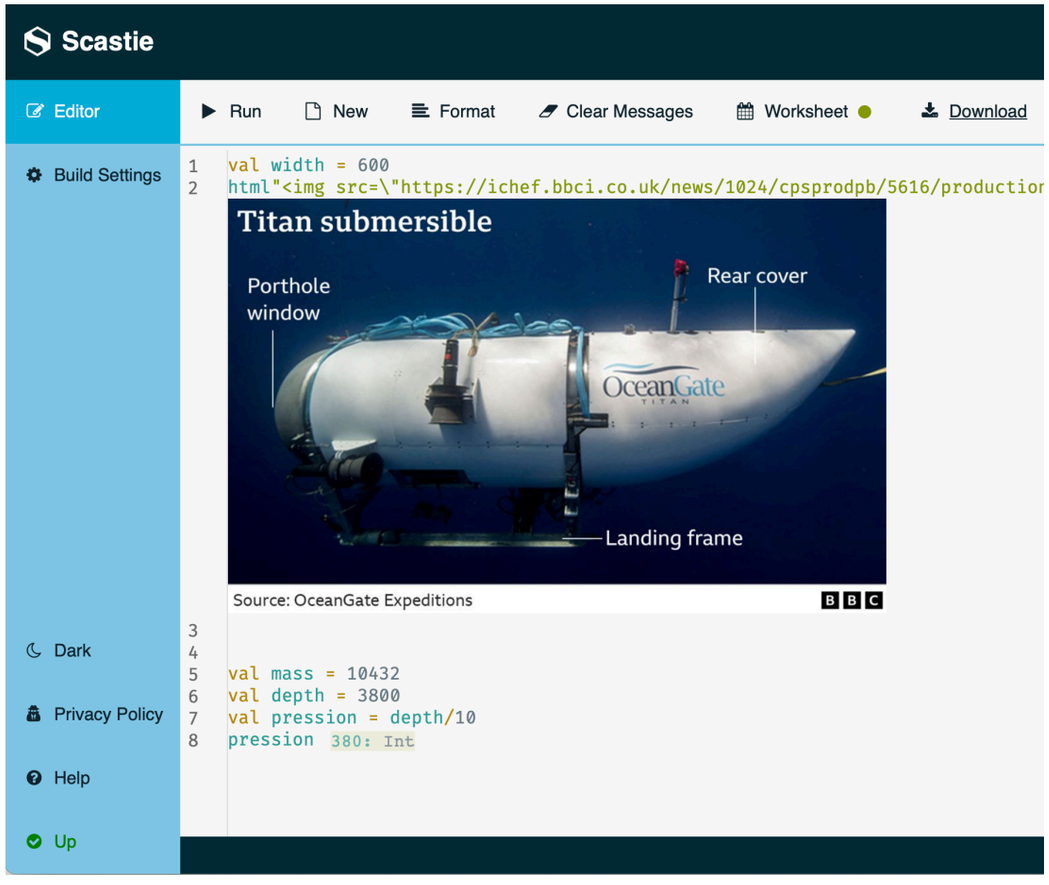


Current main features

- Worksheet mode



Instrumentation



HTML tags

Problem

Scastie

Feedback

DavidKalajdzic

Editor | Run | New | Format | Clear Messages | Worksheet | Download | Embed | Metals Status

Build Settings

```
class Kangaroo(val name: String) {
  var energy: Int = 100

  def jump(): Unit = {
    if (energy >= 10) {
      println(s"$name is jumping!")
      energy -= 10
    } else {
      println(s"$name is too tired to jump.")
    }
  }

  def eat(): Unit = {
    println(s"$name is eating.")
    energy += 20
  }
}

class Human(val name: String) {
  var hunger: Int = 50

  def walk(): Unit = {
    if (hunger <= 70) {
      println(s"$name is walking.")
      hunger += 10
    } else {
```

Console (F3)

```
Starting the Kangaroo Simulation!
------------------------
Kangy is jumping!
John is walking.
Kangy is eating.
John is eating.
------------------------
Simulation complete!
```

Dark

Privacy Policy

Help

Up

Project goal

- **Multi file support**
  - organise and navigate big snippets
  - Macros support
- Develop UI to edit multiple files (file hierarchy/tabs)
  - File/Folder creation/deletion/renaming/moving
  - Handling tab switching (Error messages per file)
- Server side logic for compilation
- Enable autocomplete across files
- Backward compatibility of all existing features
- Scastie is becoming an IDE…

https://github.com/scalacenter/student-projects/issues/12

48

.scala files in one folder

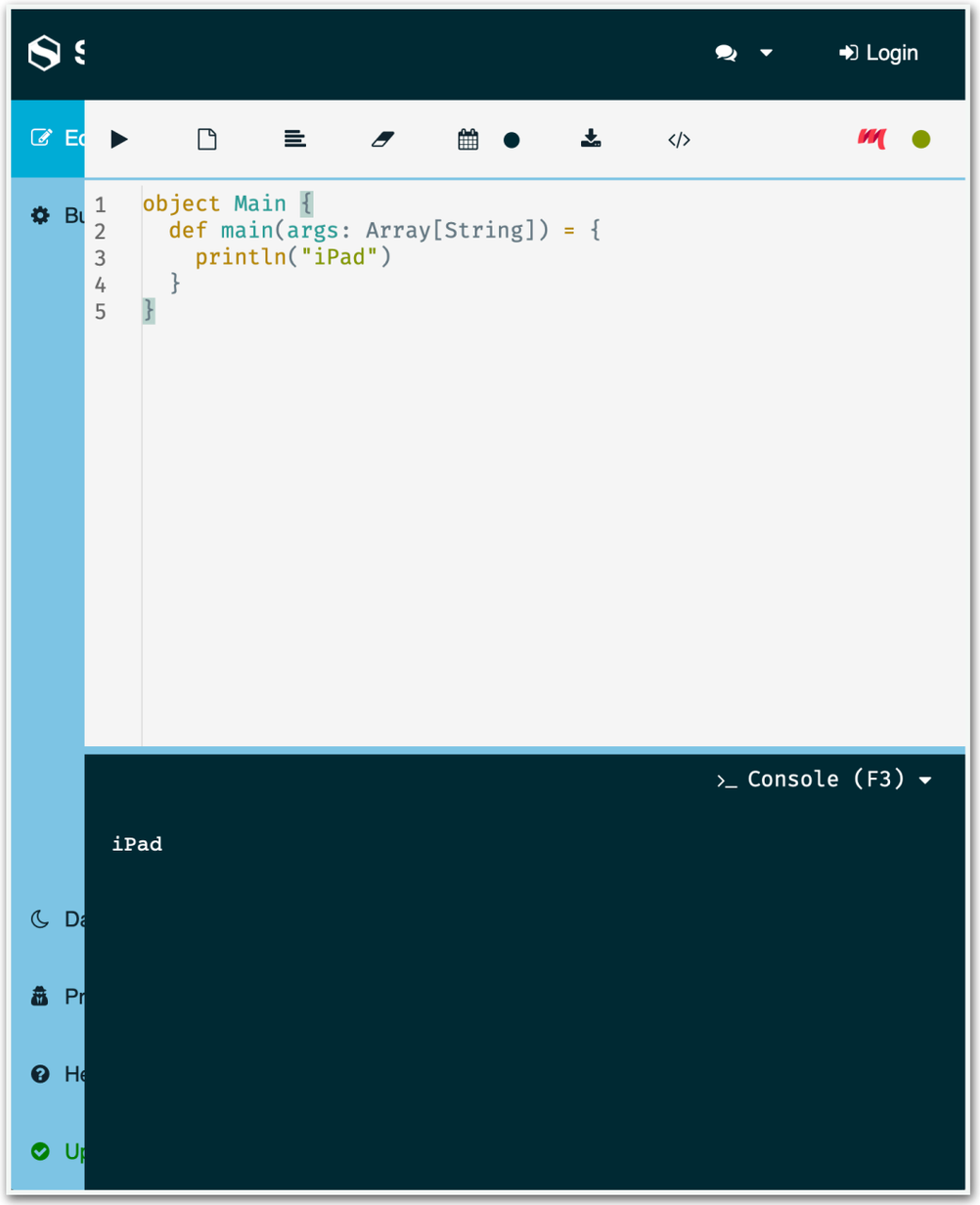
0

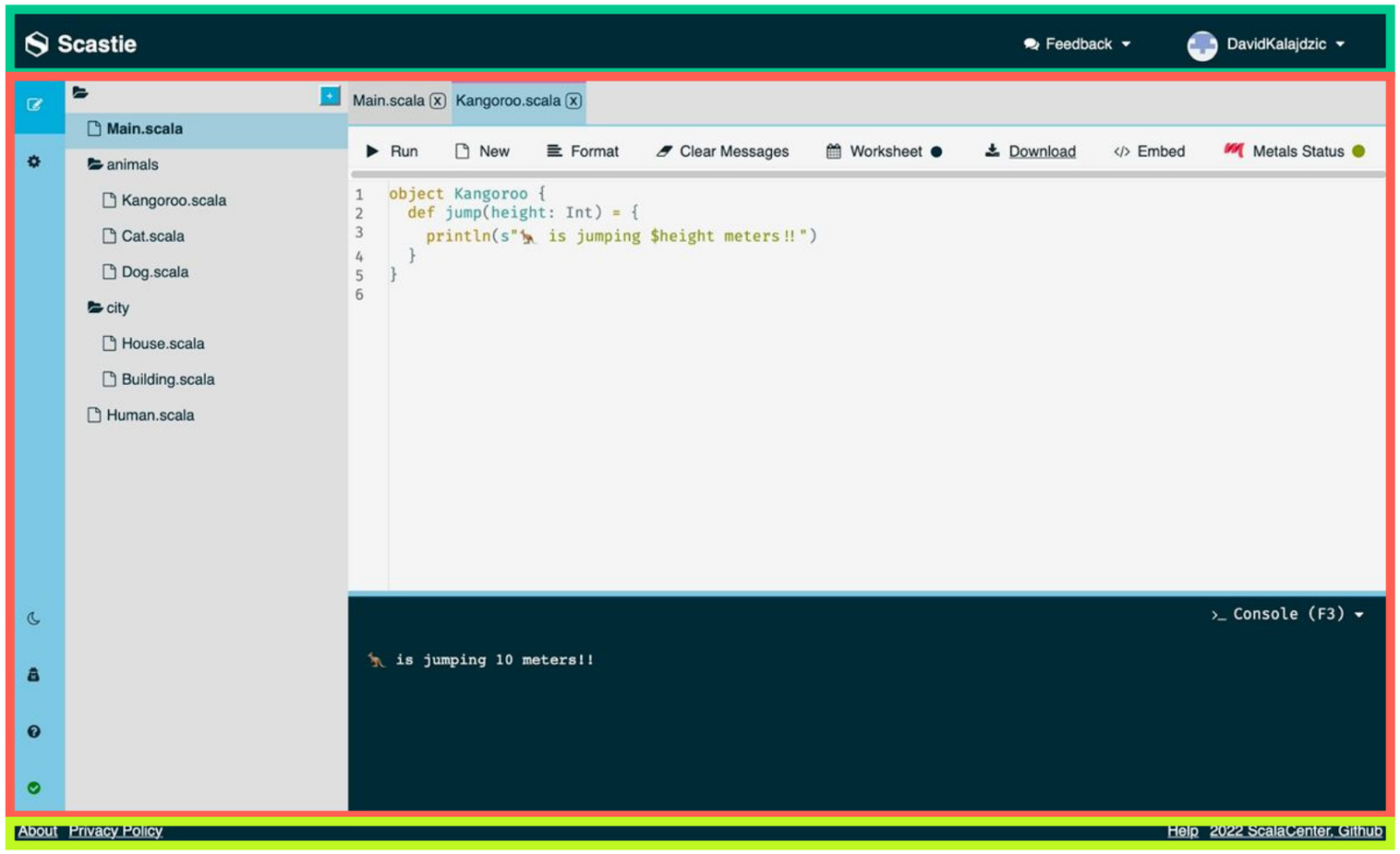
Documentation

Front end

- ScalaJS-React (https://github.com/japgolly/scalajs-react)
- Shared API definitions with Backend
- Challenges
  - Refactor the UI
  - FileHierarchy, TabStrip connected
  - Don't destroy existing features

Design a layout

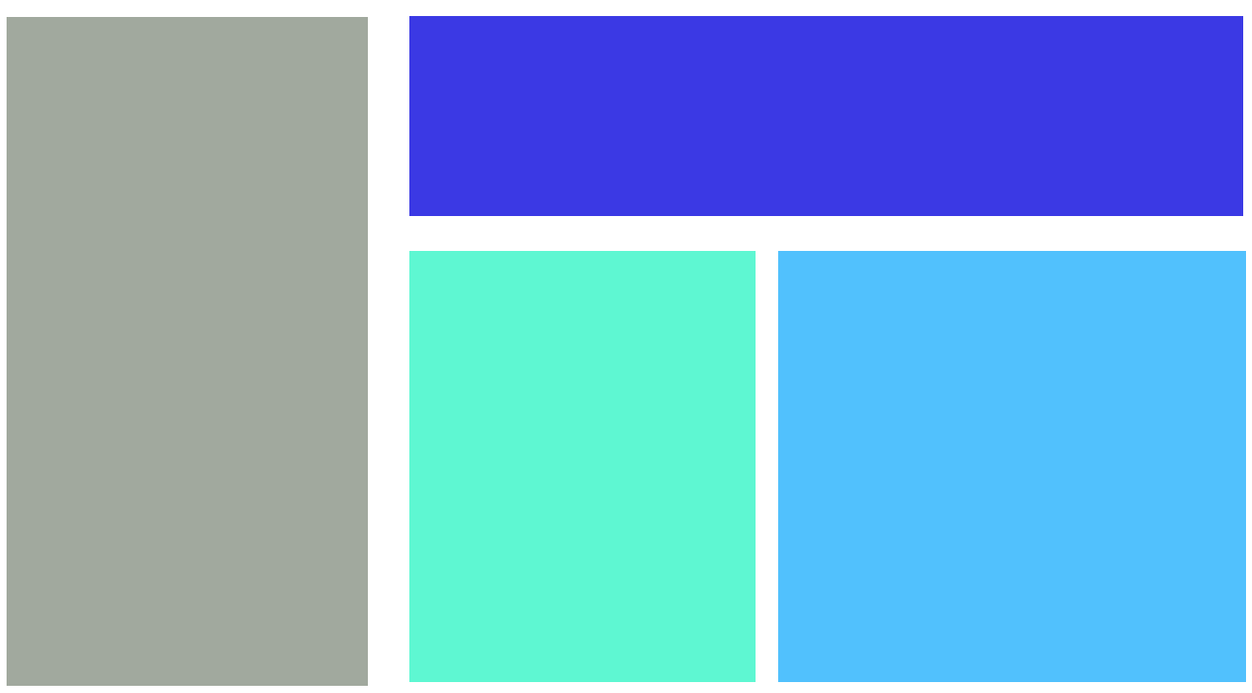




CSS Grid vs Flexbox

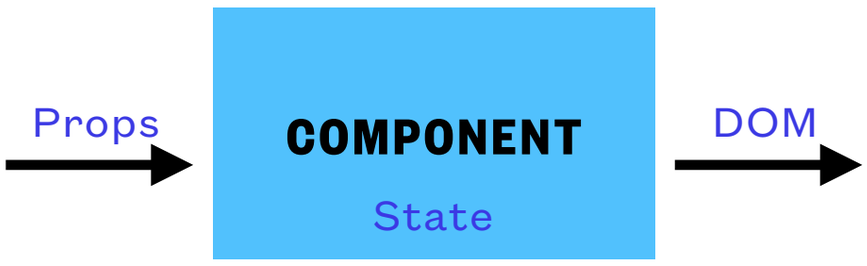
- Flexible box

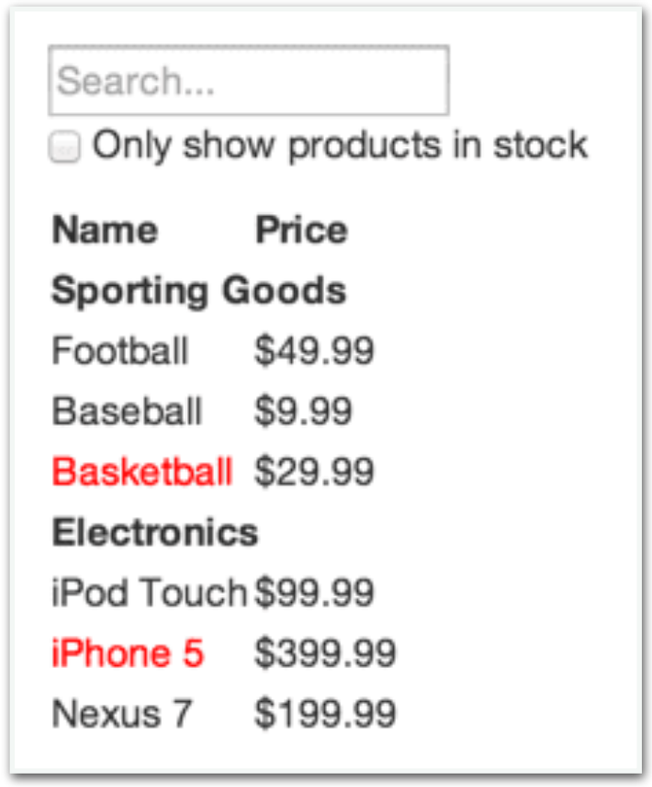
- Grid layout



React technology

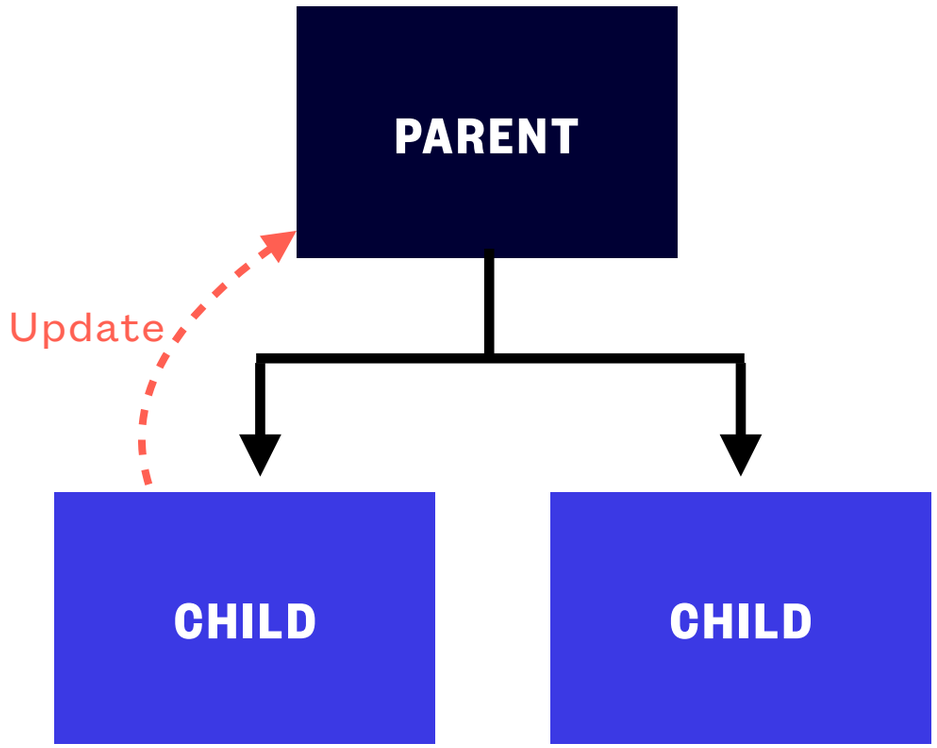
- Allows to write less code
- Reusable components (single responsibility principle)
- Components encapsulates State+UI behaviour
- Refreshing performant





Example of complex UI (1)

React technology



React tree (Parent as source of truth)

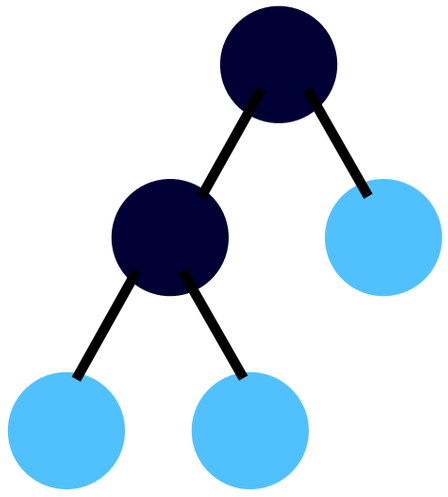
Functional programming

- In React we don't directly mutate vars
- We call "modState"
- Functional language enforces this concept

```
sealed trait FileOrFolder(name, path)

case class Folder(name, path, children: List[FileOrFolder])

case class File(name, path, content: String)
```

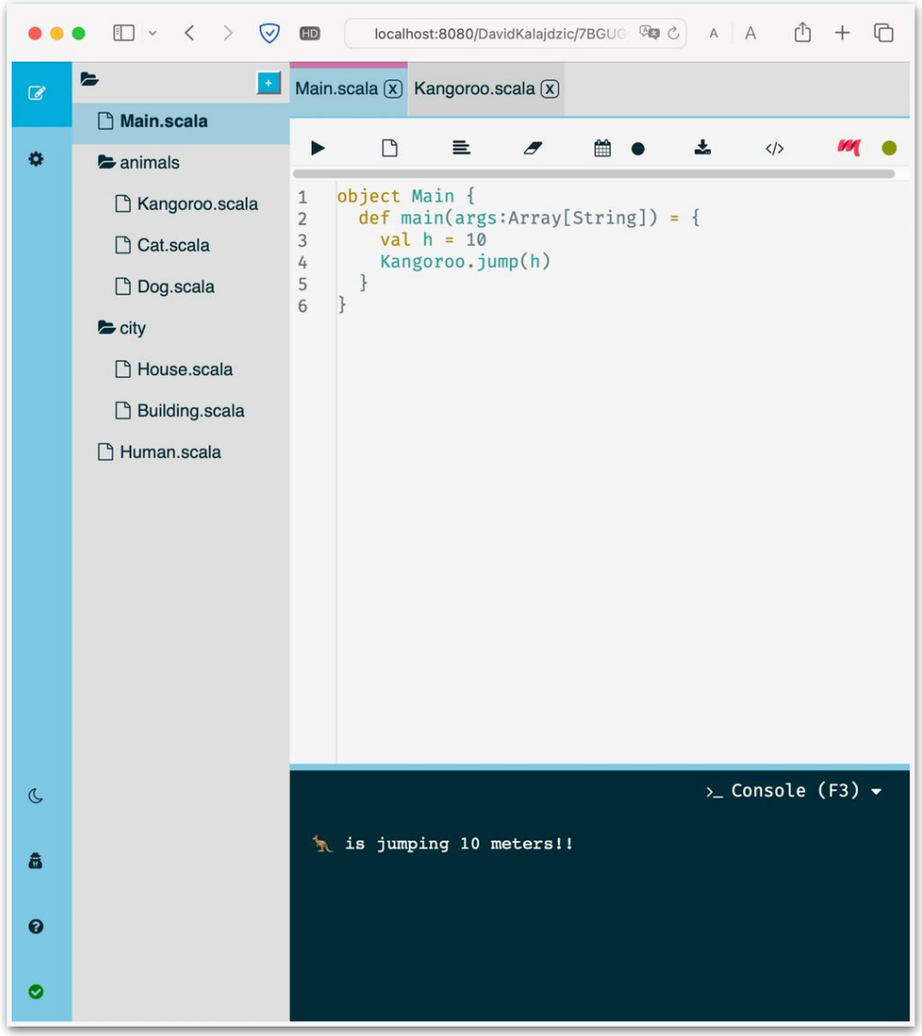


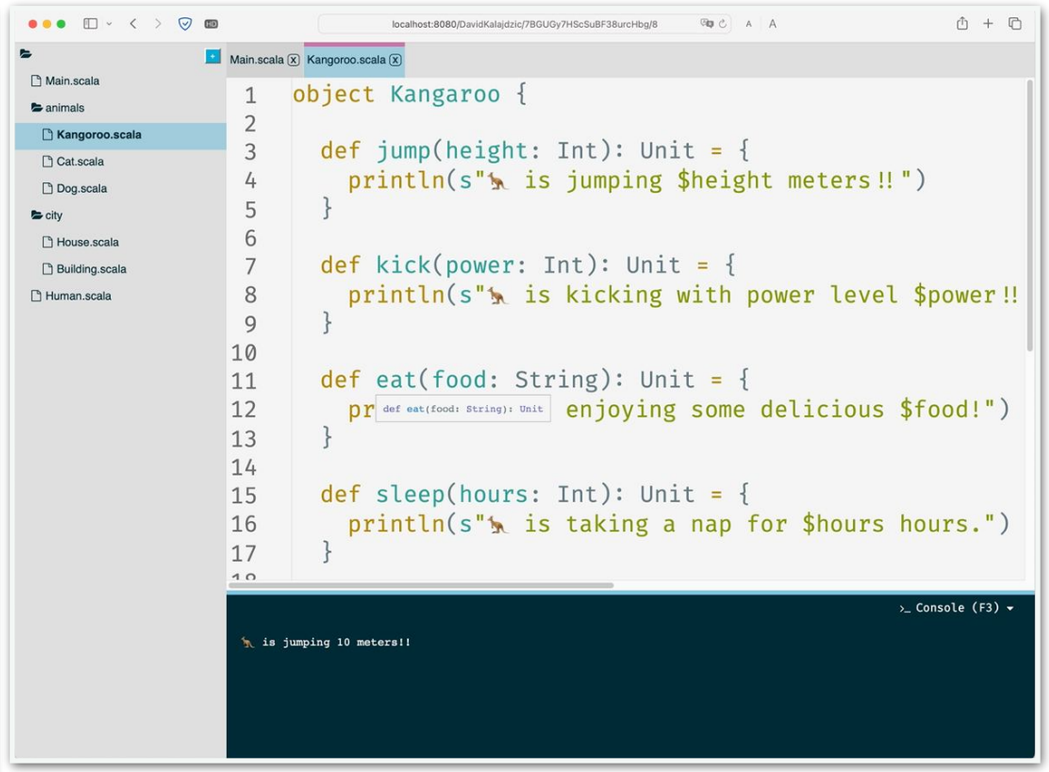
Demo

Front end conclusions

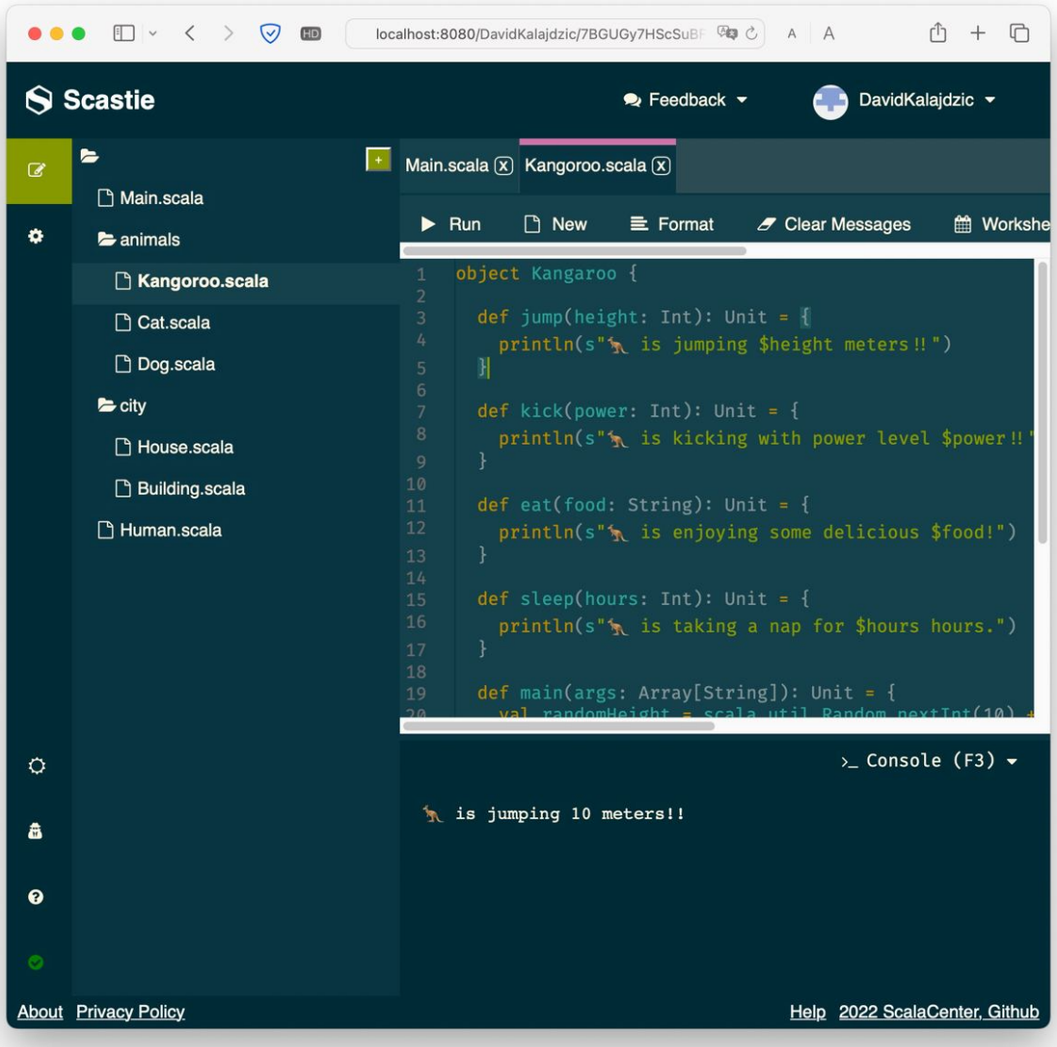
- Refactor UI to incorporate FileHierarchy & TabStrip
- Mobile & Tablet support
- Thin side bar (fast switching)
- Footer bar
- Embed mode (file hierarchy & tabstrip)
- Presentation mode (tab strip hidden)

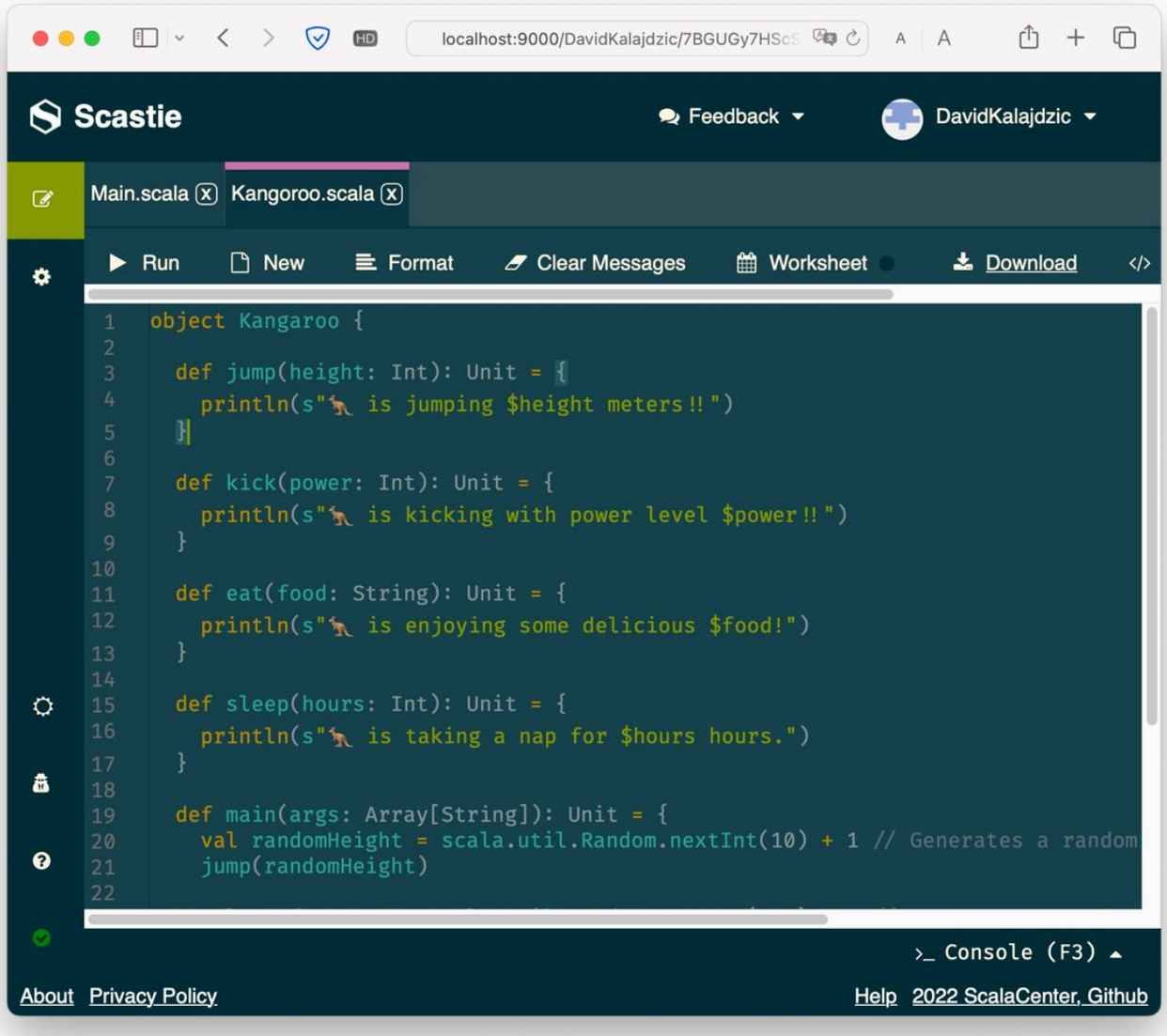
Compose as needed





Scroll bars width



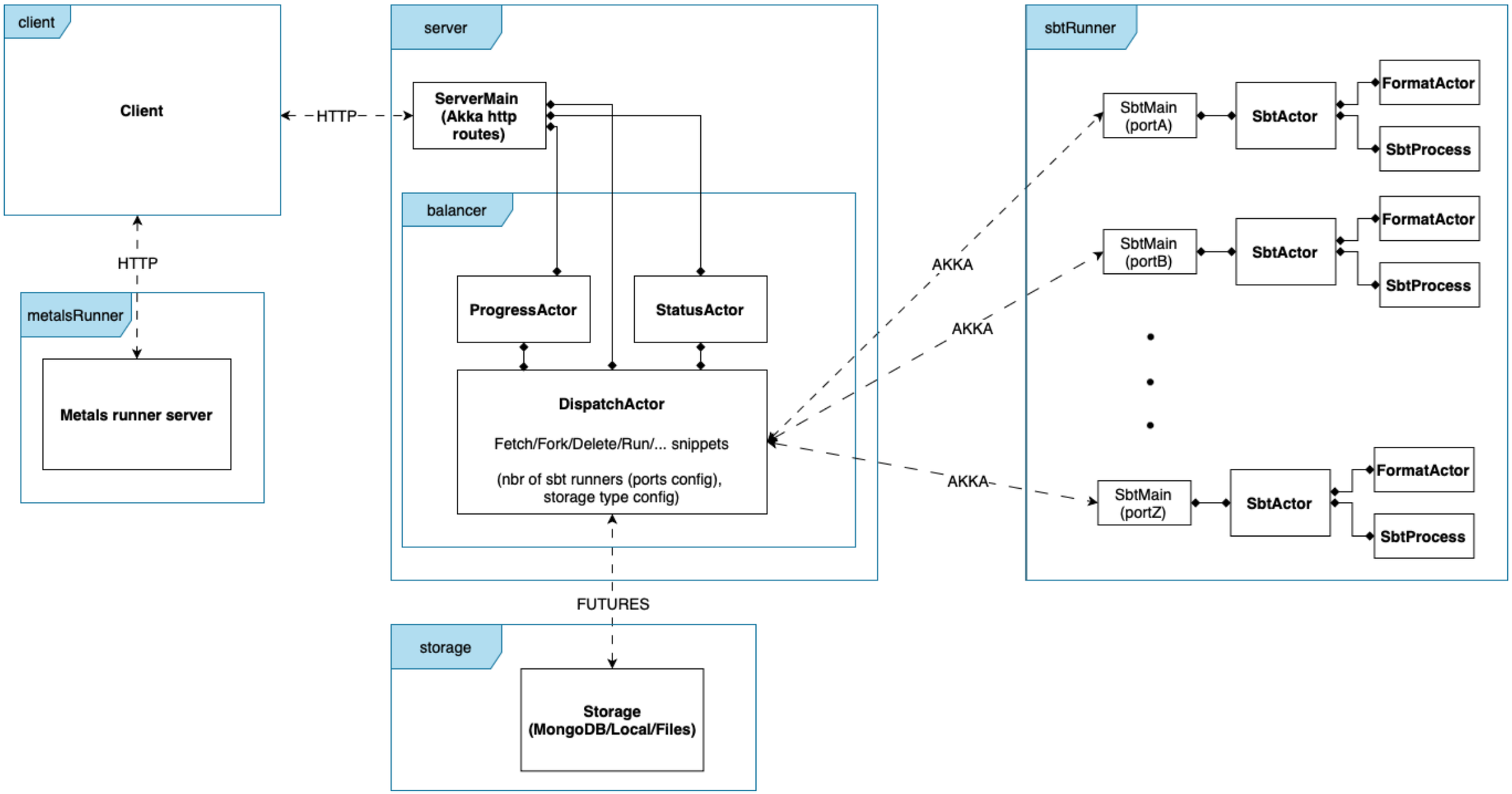


Backend

Main challenge

- Support old snippets and features
- Define a data structure for Multiple files

Backend architecture



Support old snippets

- Change schema & write Migration script
  - + clean and simple solution
  - + major feature hardcoded
  - – migration scripts to write
- Handling versions on the fly
  - + easy to implement for highly demanded services
  - – annoying to maintain
  - – increased latency

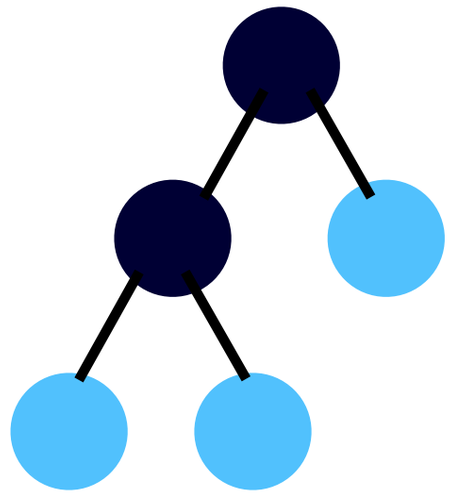
Migration script

- String → Folder

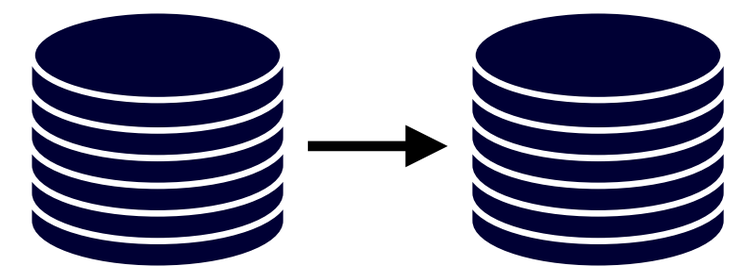
```
sealed trait FileOrFolder(name, path)

case class Folder(name, path, children: List[FileOrFolder])

case class File(name, path, content: String)
```

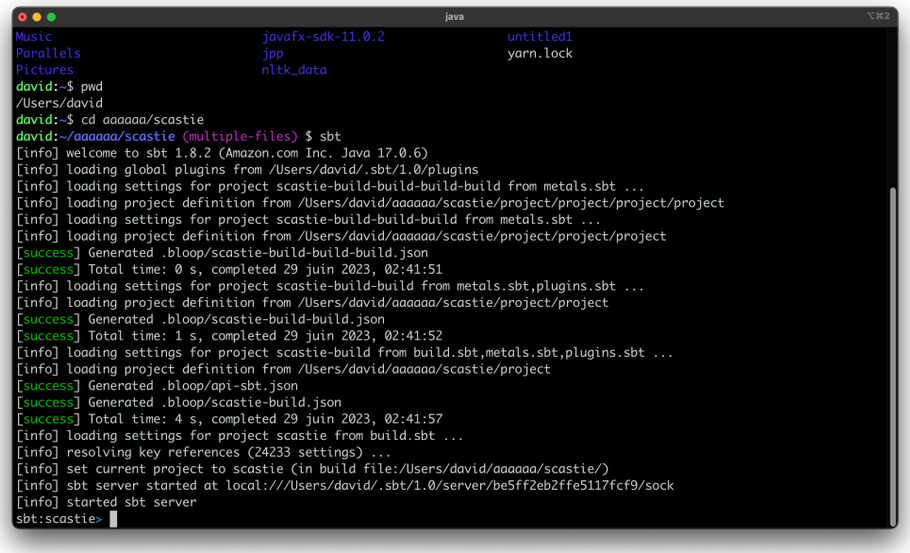


- Migration script
  - create new collection (with same indexes)
  - fetch all snippet ids
  - transform each snippet and store in that new collection

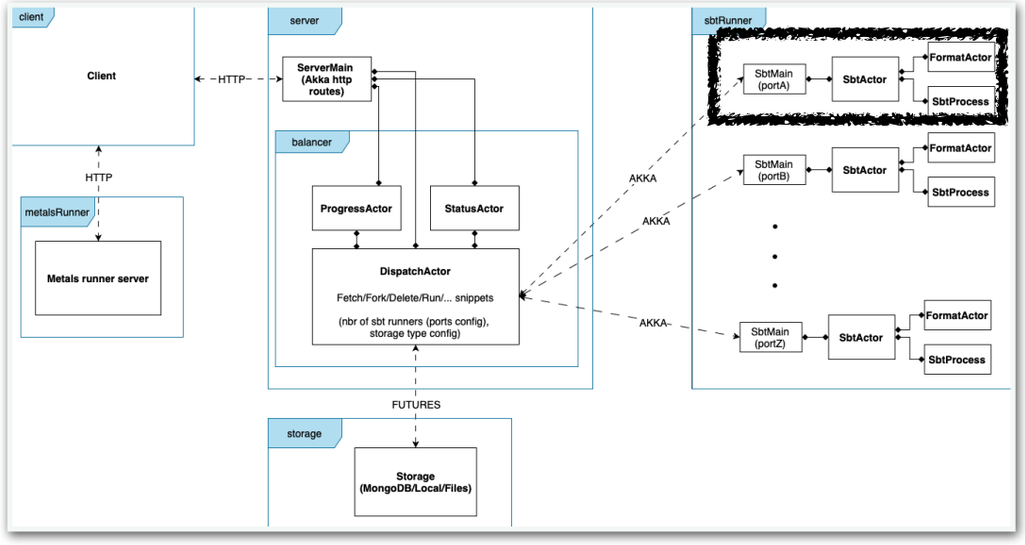


SbtProcess

- SbtProcess actor allows interaction in command line with a sbt instance
- A sbt project is created on disk from a Scastie snippet
- It was overwriting a main.scala
- *reload; compile/compileInputs*
- *fgRun*



sbt illustration



Recall the architecture

Compilation & Runtime infos

```
1  object Main {
2    def main(args: Array[String]) = {
3      println("Hello World)
4  unclosed string literal
5
6  }
```

Compile time error

```
1  object Main {
2    def main(args: Array[String]) = {
3      println(10/0)
4  java.lang.ArithmeticException: / by zero
5          at Main$.main(main.scala:3)
6          at Main.main(main.scala)
           at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
           at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:77)
           at java.base/jdk.internal.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
           at java.base/java.lang.reflect.Method.invoke(Method.java:568)
           at sbt.Run.invokeMain(Run.scala:143)
           at sbt.Run.execute$1(Run.scala:93)
           at sbt.Run.$anonfun$runWithLoader$5(Run.scala:120)
           at sbt.Run$.executeSuccess(Run.scala:186)
           at sbt.Run.runWithLoader(Run.scala:120)
           at sbt.Run.run(Run.scala:127)
           at com.olegych.scastie.sbtscastie.SbtScastiePlugin$$anon$1.$anonfun$run$1(SbtScastiePlugin.scala:38)
           at scala.runtime.java8.JFunction0$mcV$sp.apply(JFunction0$mcV$sp.java:23)
           at sbt.util.InterfaceUtil$$anon$1.get(InterfaceUtil.scala:21)
           at sbt.ScastieTrapExit$App.run(ScastieTrapExit.scala:258)
           at java.base/java.lang.Thread.run(Thread.java:833)
```

Runtime error

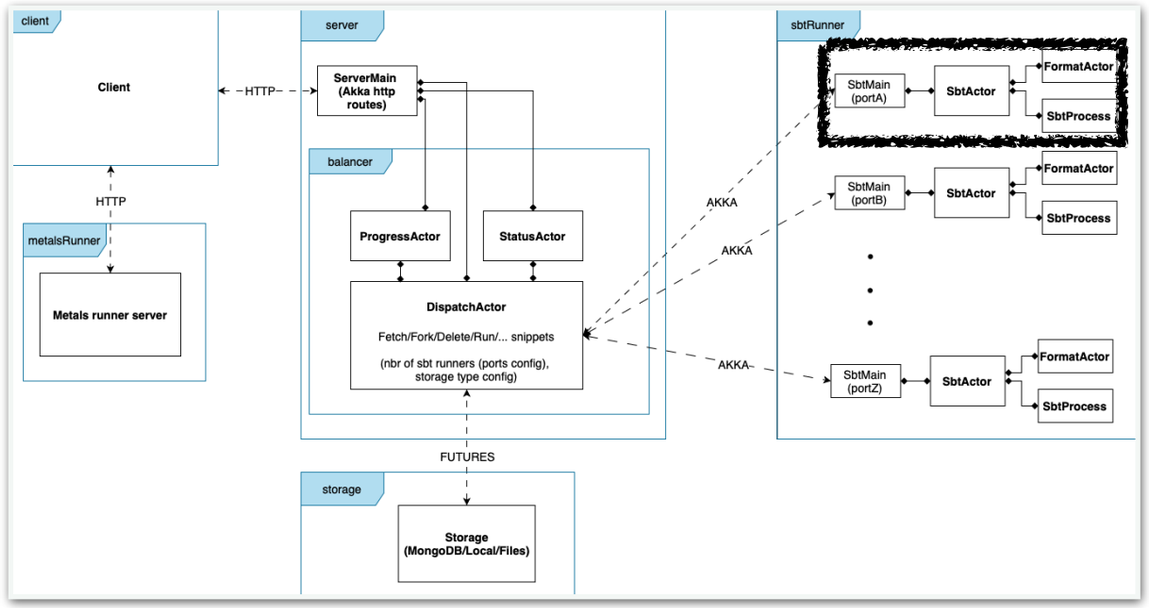
Sbt-scastie plugin

```scala
private def sbtPluginsConfig0(withSbtScastie: Boolean): String = {
  val targetConfig = target.sbtPluginsConfig

  val sbtScastie =
    if (withSbtScastie)
      s"""addSbtPlugin("org.scastie" % "sbt-scastie" % "${BuildInfo.versionRuntime}")"""
    else ""

  mapToConfig(targetConfig, sbtScastie)
}
```

sbt-scastie Plugin injected to each sbtRunner



sbtRunner

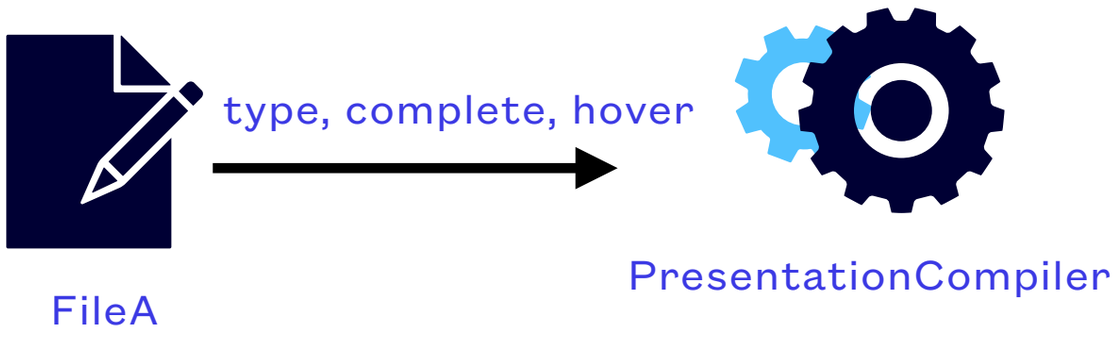
```
sbtRunner < 182: Input(fgRun)
sbtRunner -- Running --
sbtRunner > 183 30767ms:  {"output":{"line":"[info] compiling 1 Scala source to /private/var/folders/jb/f9h5g4rs3xb_sdsm_dbbr9jw0000gn/T/scastie74554
sbtRunner > 184 102ms: [{"severity":"Error","line":3,"filePath":"${BASE}/src/main/scala/code.scala","message":"unclosed string literal"}]
sbtRunner > 185 10ms: {"output":{"line":"[error] (Compile / compileIncremental) Compilation failed","tpe":"StdOut"},"tpe":"SbtOutput"}
sbtRunner > 186 9ms:
sbtRunner > 187 11ms: OtiIyNIWuc
sbtRunner -- Ready --
```

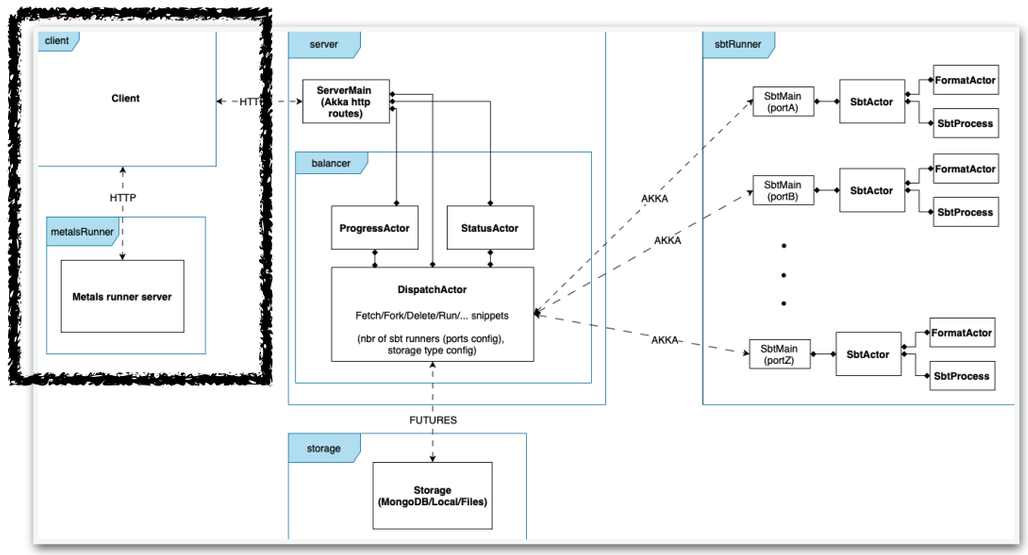
```
sbtRunner Running: (login: Some(DavidKalajdzic), ip: unknown)
sbtRunner  Folder(,List(File(Main.scala,o
sbtRunner writing in /var/folders/jb/f9h5g4rs3xb_sdsm_dbbr9jw0000gn/T/scastie7455467925025399069/src/main/scala
sbtRunner -- Running --
sbtRunner < 158: Input(fgRun)
sbtRunner > 159 21809ms:  {"output":{"line":"[info] compiling 1 Scala source to /private/var/folders/jb/f9h5g4rs3xb_sdsm_dbbr9jw0000gn/T/scastie7455467925
sbtRunner > 160 239ms: {"output":{"line":"[info] done compiling","tpe":"StdOut"},"tpe":"SbtOutput"}
sbtRunner > 161 16ms: {"output":{"line":"[info] running Main","tpe":"StdOut"},"tpe":"SbtOutput"}
sbtRunner > 162 4ms: Hello World
sbtRunner > 163 1ms: {"output":{"line":"[error] (run-main-0) java.lang.ArithmeticException: / by zero","tpe":"StdOut"},"tpe":"SbtOutput"} <1 internal line>
sbtRunner > 165 1ms: {"output":{"line":"[error] java.lang.ArithmeticException: / by zero","tpe":"StdOut"},"tpe":"SbtOutput"}
sbtRunner > 166 0ms: {"output":{"line":"[error] \tat Main$.main(Main.scala:4)","tpe":"StdOut"},"tpe":"SbtOutput"}
sbtRunner > 167 0ms: {"output":{"line":"[error] \tat Main.main(Main.scala)","tpe":"StdOut"},"tpe":"SbtOutput"} <4 internal lines>
sbtRunner > 172 28ms: {"output":{"line":"[error] Nonzero exit code: 1","tpe":"StdOut"},"tpe":"SbtOutput"}
sbtRunner > 173 0ms: {"output":{"line":"[error] (Compile / fgRun) Nonzero exit code: 1","tpe":"StdOut"},"tpe":"SbtOutput"}
sbtRunner > 174 10ms:
sbtRunner > 175 9ms: OtiIyNIWuc
sbtRunner -- Ready --
server c.o.scastie.balancer.DispatchActor:98 | pinged ActorSelection[Anchor(akka://SbtRunner@127.0.0.1:5150/), Path(/user/SbtActor)] server
```

Prints in JSON

Autocompletion (ScastieMetals)

- Standalone running server (no knowledge on files, just ongoing edits)
- PresentationCompiler (a faster asynch version of the Scala Compiler)
- After each keystroke, reanalyse

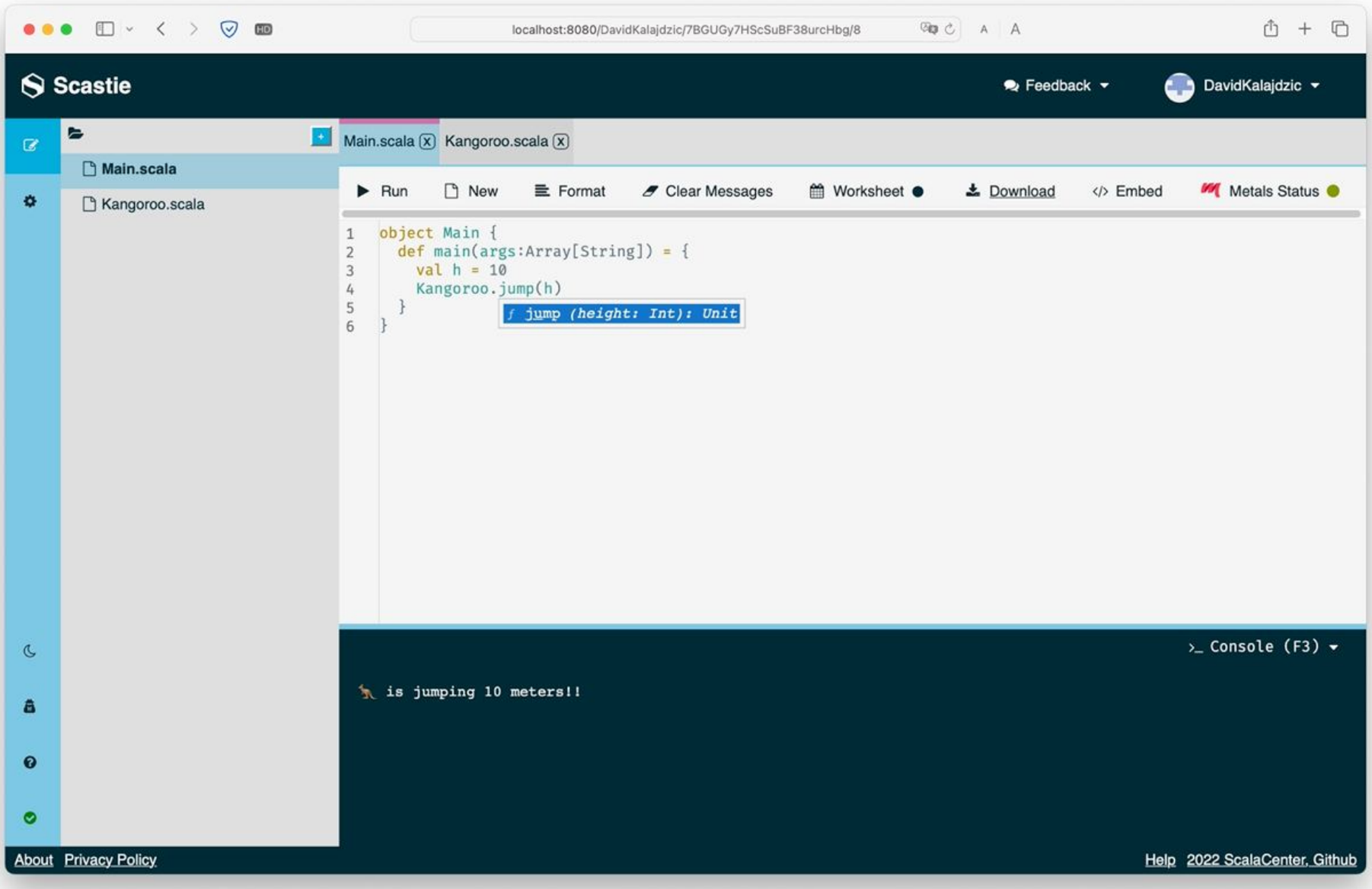




Recall the architecture

- Previously :
  - Virtual file with an URI of NoSourceFile.path
  - /complete, /hover

Autocompletion in action



Keeping the extras

- Export snippet as Zip
- Code formatting the current file

Conclusion

Conclusion

- Evaluation & testing to protect those new features
- Compatibility with browsers
- Scastie is a real world application (attention to details for prod)
- Backward compatibility of snippets
- Multi file snippets will enhance user experience
- Macros are now supported