# Scala Center updates
# Q3 2021 Advisory Board meeting

Scala Center team: Julien Richard-Foy, 60%; Jamie Thompson, 100%; Vincenzo Bazzucchi, 100%; Adrien Piquerez, 100%; Meriam Lachkar, 100%; Katja Goltsova, part-time intern, until September; Maxime Kjaer, part-time intern, until July; Sébastien Doeraene, 100%; Darja Jovanovic, 100%, Vincent Derouand, 40%; Valérie Pedroni, 80%; VirtusLab team: Tomasz Godzik, 100%; Wojciech Mazur, 100%.

## At a glance

# Inclusive language guide (SCP-025)

@sjrd @ValeriePe

We performed the necessary research and discussions around inclusive language for the Scala community. We wrote a version of the guide, already reviewed by Lightbend, VirtusLab, and the LAMP team at EPFL.

We intend to publish that guide on the scala-lang.org website after the current advisory board meeting. Advisory Board members received the current draft together with this report, and are encouraged to provide feedback before we publish its final version.

In the meantime, we have already been applying the guidelines in a number of our repositories. For some others, co-maintainers outside our organizations are waiting for the guide to be published to apply the discussed changes, which means they will take effect soon after the publication.

# Solidify getting started with coursier (SCP-026)

@sjrd @adpi2

We published the roadmap for addressing this Recommendation on Contributors. We implemented the necessary foundational features in coursier:

- Create App Descriptors that are always installed for a third-party .zip/.tgz archive (required to install sbt's official distribution)
- Specify different artifacts, main methods and other information for different versions of a given application (required to merge the Scala 2 and Scala 3 distributions under a unified "Scala" distribution)

We also fixed some issues regarding the introduction with Scala 3 and with Windows.

# MOOCs

@julienrf @vincenzobaz

## Effective Programming in Scala

This new course was published during the previous quarter. We have been monitoring the discussion forums, and improving the assignments based on the feedback of learners. We have also upgraded the course to Scala 3.0.2.

We observed a higher completion rate after we improved the course content (40% vs 28% in the first version). Overall, almost 3,000 new learners took that course in Q3.

# Scala documentation website

@julienrf @vincenzobaz

We have been working on making the Scala documentation simpler and better, as presented in the roadmap on Contributors. In particular, during this quarter, we reviewed and improved the Scala Book, and moved content from https://dotty.epfl.ch/ to https://docs.scala-lang.org/, in order to unify the documentation websites.

We have also reviewed and merged various pull requests from the community on the documentation website.

# Scala in universities

@vincenzobaz @julienrf

We are looking into widening the Scala community by using the language for introductory courses in universities.

We interviewed three lecturers and we collected their opinions, difficulties and advantages about using Scala for teaching.

# Scala Debug Adapter

@adpi2

The main highlight for this project this quarter is the expression evaluation for Scala 2.12 and Scala 2.13.

To evaluate an expression at some breakpoint, the debugger needs to find the original source file of the top class in the stack frame. We implemented that by using the source file name that is stored in the class files. Because Scala source paths do not always reflect the name of the package, this method does not work 100% of the time. However we found a good heuristic that handles a large majority of cases. Some metrics can be found here.

Tomasz Dudzik (@tdudzik) from VirtusLab implemented the expression compiler and evaluator for Scala 2.12 and Scala 2.13. Some cases are not yet supported (like nested methods or lambdas) and he is working on it.

We released scala-debug-adapter v2.0, featuring expression evaluation for Scala 2.12 and 2.13. We adapted Bloop, sbt-debug-adapter and Metals to support it. It will be available to the user in the next version of Metals.

## Metals

@tgodzik

We published the new Metals releases v0.10.5, v0.10.6 and v0.10.7 with improved support for Scala 3, among others. The main new features are:

- Reworked the tests on the integration with sbt server to improve stability
- Preparing expression evaluation during debug sessions
- Inferred type Scala 3 action
- Allow organize imports on unsaved code
- Support for Scala 3.0.2 and 3.1.0-RC1
- JDK 17 support
- Code action to switch to braces/parentheses
- Fixes for organize imports scalafix rule in Scala 3

## bloop

@tgodzik

We improved the support for Maven multimodule projects.

## Gradle

@tgodzik

We added support for Scala 3 in Gradle.

## Scalafmt

@tgodzik

We worked on general bug fixing, notably for the support of Scala 3.

## Scalafix

@mlachkar

We published Scalafix v0.9.30, with the following highlights.

We fixed a bug in scalameta that was impacting scalafix users. In fact, Some rules rely on the warning messages to apply some fixes. Since 2.12.13 and 2.13.x, the number of warnings stored by scalameta depended on a compiler option -Xmaxwarns, which limits the number of fixes applied during each run, and the user had to configure this scalacOption to catch more warnings. Now Scalameta reports all warning independently of this scalacOption.

The support for sbt 0.13 is now dropped. Starting with this version, the plugin is only available for sbt 1.x.

We added a new feature to the RemoveUnused rule, which allows removing unused function parameters (#1444, #1448).

# sbt

@adpi2

We worked on making sbt server and the BSP implementation more reliable. In particular, the command that starts the BSP server now loads the sbt and java options from the `.sbtopts`-like files, and from the environment variables by using the official sbt shell script. We also improved the error reporting of the `workspace/reload` request and we made many of the BSP requests robust to individual build target failures.

# Scaladex

@adpi2 @mlachkar

Scaladex is a showcase of the Scala ecosystem and we can make it more attractive in terms of content, design but also technical stack. This is why we decided to redesign Scaladex to make the maintenance easier, the design more scalable, and the data more accurate.

For the last two months, we performed the following main changes:

- Replacing the usage of ElasticSearch by PostgreSQL for storing the data. This new design brings better data accuracy and fixes some bugs that were hard to fix in the previous design, including the dependencies and reverse dependencies for projects on the project page.
- Fix the "most dependent upon projects" feature
- Working on improving the badges that projects add to their GitHub readmes.

# Scastie

@vincenzobaz

We improved the support of Scala 3 in Scastie, notably with the following improvements:

- Allow users to choose among different Scala 3 versions
- Allow users to choose among different Scala versions when the target is Scala.js
- Add worksheet support for Scala 3 snippets

- Fix scalafmt integration for Scala 3 snippets
- Add support for Scala 3 dependencies

# Compiler

@bishabosha

We are continuing to provide maintenance efforts to the Scala 3 compiler, as well as the TASTy Reader in the Scala 3 compiler.

Some of the highlights for this quarter are various bug fixes around union types, ClassTags, and the TASTy reader. In addition, we implemented synthesizing old-style Manifests when requested by implicit search, which improves compatibility with Scala 2.

# TASTy manipulation library

@cache-nez

We can now unpickle simple types and resolve symbols, including across different TASTy files. The next difficulty is to read Java class files, as we discovered that we cannot build a meaningful type system, even a cripled one, without them.

# Scala.js

@sjrd

We published Scala.js 1.7.0 and later v1.7.1. The highlight of Scala.js 1.7.0 is that we completely rewrote the implementation of `java.util.regex.*` to be compatible with Scala/JVM. This fixes several compatibility issues with regexes that have been reported over the past 7 years. While these fixes had the potential to cause compatibility concerns (if users depended on the JS-specific behavior), none has been reported since the release, two months ago.

With these changes, Scala.js 1.7.0 became the first version of Scala.js to have zero known issues at the time of its release. New small bugs were reported since then and were fixed in Scala.js 1.7.1.

# Scala Native

@wojciechmazur

The big highlight is that we finished the support for Windows. It was released in Scala Native v0.4.1.

We have also been working on support for multithreading, notably in the areas of ForkJoinPool and of the garbage collectors.

Finally, we started porting the compiler plugin to Scala 3, which is the central piece to add Scala 3 support for Scala Native.

# Communication and Management

@darjutak @ValeriePe @VincentDrnd

In this quarter, the communication team especially worked on **standardizing Scala Center identity**. It is crucial and necessary that the "face" of the center reflects our values and dedication thus making it more professional in both networking and fundraising settings.

To achieve that:

- In the process of getting @scala_lang twitter account verified. We noticed some twitter accounts had the same outward appearance as us (such as @scalafriends) or presented themselves as leading organisations. That is why it seems imperative to distinguish ourselves better and clarify our status by getting the blue mark.
- Organized, along with HR, a team photoshoot to standardize the team pictures on our website and other social platforms.
- Created a LinkedIn account for the Scala Center and defined our strategy and intentions. The posting schedule for Q4 2021 is defined here.

**Scala courses promotion campaign** *(in collaboration with Coursera and 47 degrees)*

- Filmed a promotional video (will be out soon!)
- Sent an email to one million ex learners and got 1000 new enrollments to the course Effective Programming in Scala
- Tweeted (link) which brought 100 new enrollments to the course Effective Programming in Scala

**Scala Center 5 Year Report (2016-2020)**

- Collected in a document data such as:
  - AB data (i.e. number of meetings, changes in representatives, number of members, projects and proposals).
  - Scala Center data (i.e. staff movements, projects and proposals)
  - SIP (i.e. number of completed, dormant, rejected and pending)
  - MOOCs (i.e. number of courses, learners and completers)
  - Scala Events (i.e. from 2010 to 2020, number of events/year, countries and cities)
  - Scala Meetups
  - Scala Conferences

- Interviewed seven people from the Scala community (i.e. AB, Scala Center, Scala Users and/or collaborators).

To ease future Scala Center staff onboarding, we finalized the existing onboarding document.

The 2nd Organizer Summit blogpost was written and published.

Our yearly team retreat took place between September 7th and September 9th at the Maison de Bretaye.

# Let's Talk About Scala 3 video series

@vincenzobaz @anatoliykmetyuk (LAMP) @MaximeKjaer @tgodzik @wojciechmazur @adpi2

In this quarter, we published the following videos:

- Anatolii Kmetiuk: Ammonite Ecosystem
- Maxime Kjaer: Metaprogramming in Scala 3: Inline
- Tomasz Godzik: Getting Started with Metals IDE
- Wojciech Mazur: Introduction to Scala Native

We also recorded one more video on sbt server, to be published later.

From when we started, at the beginning of March, 11 videos have been published in total.

Current stats are:

- A total of 16.5k+ views
- 1,200 hrs+ of watchtime
- 757 likes and 4 dislikes

# Google Summer of Code

@vincenzobaz @darjutak

The bulk of the Google Summer of Code happened during this quarter, and ended last month. All four of the students who worked with the Scala organization successfully completed their projects.

We collected feedback from students both using the answers to the questions that GSoC asked them but also directly via email. The experience was very positive for students and we hope that they will continue contributing to the Scala ecosystem.
We sent some Scala goodies to the students, including invitations to ScalaCon. Students

participated in the event and one talk acknowledged the participation of one student to the project.

We published a blog post about our experience, which is available here.

We are preparing for the next edition of GSoC which will include multiple changes that can be beneficial for the Scala community.

# Scala Symposium

@sjrd @julienrf

We were asked this year to organize the Scala Symposium, the main research symposium centered around the Scala programming language, in order to help the research community for this difficult year.

We formed the Program Committee, oversaw the paper submission process, and their reviews. We also filled in the administrative details asked by the SPLASH conference, which is responsible for all the logistics. Finally, we chaired the sessions during the actual event, which took place on Sunday October 17.

The Program Committee received six submissions, and four of them were accepted. You can find the program of the symposium as well as the published papers on this website.