Scala Center updates
Q1 2020 Advisory Board meeting

Scala Center team: Julien Richard-Foy, 60%; Jamie Thompson, 100%; Ergys Dona, 100%, since January 15 2020; Adrien Piquerez, 100%, since March 15 2020; Meriam Lachkar, 100%, since March 15 2020; Darja Jovanovic, 100%; Sébastien Doeraene, 100%; VirtusLab team; Maxime Kjaer, student project

At a glance

Developer Survey

@julienrf

We have published a developer survey to collect the usage trends in 2019 and possibly identify pain points in developer workflows.

We got about 4,000 respondents. The raw results are available here:
https://scalacenter.github.io/scala-developer-survey-2019/

We are looking into analysing the results and comparing with the last 3 years in order to help us better understand the trends and needs of the Scala user community and help us choose our next projects.

MOOCs

@julienrf @mlachkar

Functional Program Design

We have published a new version of the course Functional Program Design. It includes a completely new week of material explaining how to use implicits.

You can read more in the announcement.

Effective Programming in Scala

We are actively working on a new course targeting developers who are new to Scala. This single course will be an alternative to the current Scala specialization that spans 5 courses. It will go less deep in the theoretical parts and will focus on the practical parts.

Documentation around the Scala Book

@mlachkar

Recurrent points around documentation were brought by this contributor discussion “How to make scala more popular”:

- The getting started experience is confusing
- Is the tour of scala the actual documentation of scala?
- the documentation provided in the website is not enough

Alvin Alexander proposed his book “hello-scala.com” as an additional resource to the Scala documentation, which we had reviewed and merged earlier. We now identified additional possible points to improve the user experience with the Scala documentation:

- Improve the integration of the Scala Book, by fixing the content table in order to show the book structure (done)
- Rewrite the getting started guide (in progress)
- Rewrite the Download page (to do)
- Change the order of “Scala Tour” with “Scala Book” (to do)
- Change the order of Download and Getting Started links in scala-lang.org (to do)

coursier

@alexarchambault

coursier is a library to manage dependencies from Maven and Ivy repositories. It comes along with a CLI to conveniently interact with dependencies. It is used by sbt, pants, bazel, mill, etc.

Native binaries for Linux / macOS / Windows

We now build and publish native binaries for the coursier CLI on all 3 major operating systems, thanks to GraalVM. That allows to run the coursier CLI without requiring a JVM to be installed. The setup command (see below) relies on that, and offers to install a JVM itself thanks to that.

Setup command (one-click install)

We stabilized and polished the install command of coursier, which allows to install applications. We added new commands, java and java-home to install JVMs. Then, on top of these commands, we added a setup command, that allows to setup an environment for Scala, installing a JVM and CLI applications.

We are now updating the main download page of Scala to use that installer (PR in progress).

We hope that this will make it much easier to get started with Scala, by removing all the hurdles involved in installing the relevant software.

JSR-45 (SCP-022)

@errikos

JSR-45 is a Java Specification Request for adding debugging support for other languages that ultimately compile to JVM bytecode (such as Scala, Kotlin, Groovy, etc.). By implementing JSR-45 for Scala, we aim to improve the developer debugging experience.

We started with the scalac inliner and looked into what ASM provides in order to embed debug information to the .class files. We are currently working on constructing the JSR-45-specific debug information.

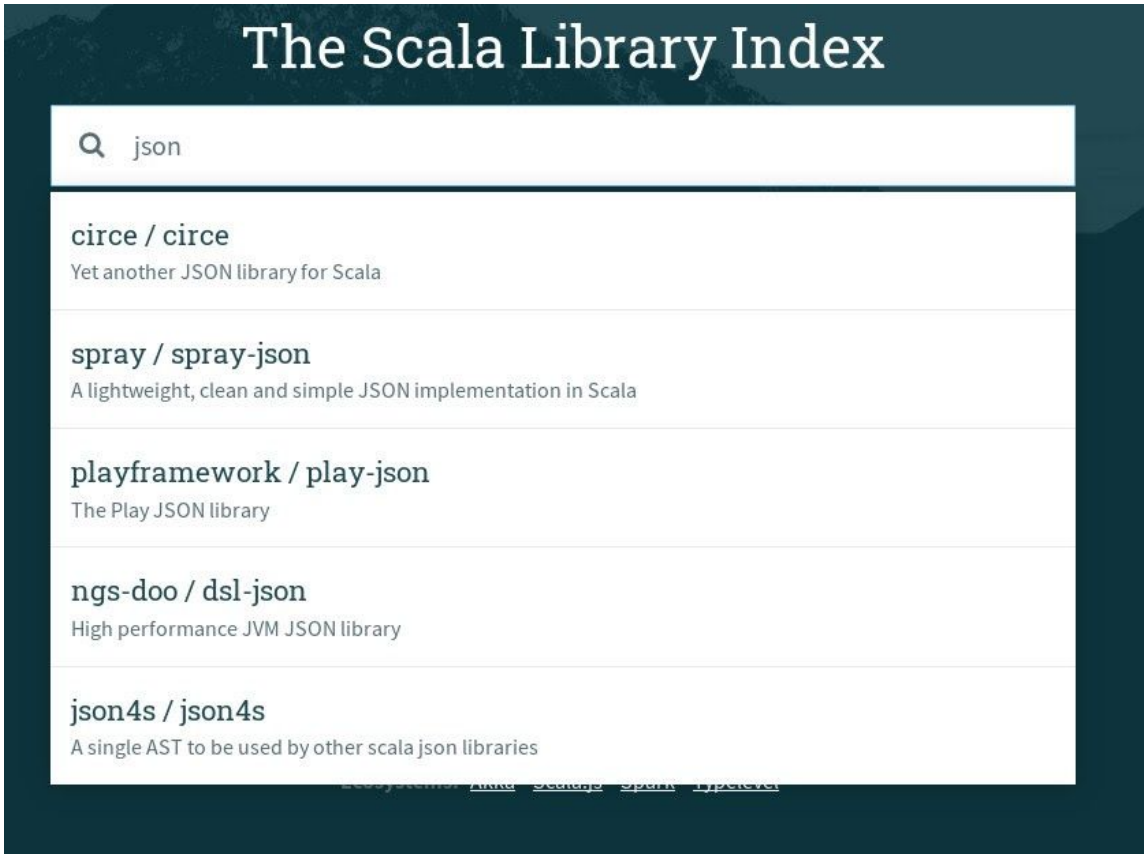
Scaladex

@adpi2

The search engine is the cornerstone feature of Scaladex. We have improved the search relevance, which had been lacking for a long time, by reworking the scoring function and by adding support for language analysis.

Scoring function

The scoring function is a combination of term matching and popularity (number of github stars). A project needs a good score on both factors to appear on top of the list.



Language analysis

Tokenization and stemming are used to analyse both the search query and the project information. Searching for “unit-testing” will now match projects that mention “tests”, “testing”, ”unit-tests”...

Metals

VirtusLab team, @alexarchambault

We released versions 0.8.0, 0.8.1 and 0.8.3 of Metals. The main highlight was the debugging support in VS Code and Emacs. Other new features include:

- Go to implementations
- Refactoring: rename symbol and import missing symbol
- Support for worksheets

- Completion to add all abstract members and insert missing abstract members

Support for Scala 3

We have been working on basic support for Scala 3 in Metals. We opened a PR with basic support a few days ago.

Support for Ammonite

We've been working on adding support for Ammonite scripts to metals. This is an ongoing work, pending review in both Ammonite and metals. In order to do that, we mainly

- added BSP support to Ammonite,
- created a new library, ammonite-runner, that can start Ammonite for compatible Ammonite and Scala versions, and
- added new commands to metals, to start Ammonite as a BSP server via ammonite-runner, and get metals IDE-capabilities in Ammonite scripts.

TASTy Reader For Scala 2

@bishabosha

There has been significant progress from the previous quarter. We are most of the way to supporting all of the shared subset of types between Scala 2 and 3. Additional work is still required for full annotation support, and more detailed below.

Highlights

We now have full support for Higher Kinded types, including detecting incompatible signatures, such as higher kinded bounds with type parameter lists of different lengths, which are supported in Dotty.

We also added support for refinement types, and from adding basic annotation support we gained analysis of pattern match exhaustivity and reachability for sealed types.

Looking forwards, support needs to be added for arbitrary trees in annotations, but these could be added after an initial public preview. Additionally there is the question of error handling for unsupported signatures from Scala 3, which is at present intended to be handled in the future by Dotty not publishing incompatible signatures if a flag is set.

Miscellaneous

The full tested feature set can be examined within the testing directory, on the various branches (within each sub-directory `src-2` (compiled with Scala 2) contains test cases of sources from

`src-3` (compiled with Dotty 0.22.0-RC1). We support reading TASTy “tags” in Scala 2 for the following at present:

- package/class/trait/object/def/val/var/type
- Higher-Kinded Type Bounds
- Sealed Types (with full pattern match analysis)
- Operator names
- Literal Types
- Path Dependent Types
- SIP-15 Value Classes (with extension methods)
- Refinement Types (type members, vals and methods, structural refinements)
- Super in paths
- Annotated types
- Annotations with simple trees (constants and paths)
- ByName and Vararg parameters
- Compound Types (intersection type in Dotty)
- `@compileTimeOnly` attachments for unsupported `inline` methods and union types

Enabling a new “tag” in TASTy is accompanied with addition of test cases that assert how it interacts with other features, but more can be done by expanding testing to include more community libraries which are likely to have novel combinations of features.

Zinc Improvements (SCP-021)

@bishabosha

We made a primary implementation of a SAX style callback API in Dotty. The new API will be used to fuse the Zinc analysis phases and transform the compiler’s traversals of symbols from constructing data structures into emitting a stream of events to be interpreted outside of the compiler, but still controlled by it.

The aim of using the new API is to remove the dependency on intermediate data structures where possible, and to reduce under compilation by enabling dependency analysis to be more reactive to events such as inline expansion.

The planned milestones are to first keep the implementation of the SAX API private to the compiler, and delegate to Zinc’s AnalysisCallback, and then move the implementation to a shared codebase between Zinc and Dotty once both use the new API.

The full report of experiments and discussions can be read here.

Scala Native

@errikos

The Scala Center is taking over development and maintenance of Scala Native. We worked on reflective instantiation of objects and classes (WIP - 80% complete).

Scala.js

@sjrd

After 7 years of development, including 5 years of stability within the 0.6.x series, we finally released and announced Scala.js 1.0.0 last month. Since then, we also released version 1.0.1 with a few bug fixes.

The 0.6.x series are not quite EOL yet, but we will only maintain it with bug fixes and additions to the JDK implementation.

We contributed support for Scala.js 1.x in Mill as well, so that Mill-based projects can be built and published for the 1.x ecosystem. The support was published in Mill 0.6.1.

We also performed the groundwork to support Scala.js 1.x in https://scalafiddle.io/, but that has not been deployed yet.

scalajs-bundler

@sjrd

We updated scalajs-bundler to support Scala.js 1.x out of the box. Improvements in Scala.js itself allowed scalajs-bundler to have a better integration with Scala.js 1.x, including better performance.

Webpack plugin for Scala.js sources (student project)

@MaximeKjaer

While scalajs-bundler exposes Webpack features to sbt, for use in a build setup mainly based on sbt, the new project for a Webpack plugin for Scala.js sources aims at providing sbt-scalajs features in Webpack, for use in a build setup mainly driven by Webpack. This is worked on by Maxime Kjaer as a student project, within his studies at EPFL.

The plugin is written in Scala.js, and will be published on NPM. The goal is to be able to import Scala.js projects in JavaScript, or simply to be able to build a JavaScript bundle only using familiar web development tools. To use the plugin, you only need to have two things installed on your machine: npm and the JVM. The plugin takes care of downloading the rest.

We have created a working configuration for writing Webpack plugins in Scala.js, and for publishing them to NPM. There is a system of integration tests, where we can create full-fledged Webpack projects that use the plugin, and test the output bundle of these projects. The plugin is

currently able to download dependencies, including the Scala.js compiler, from online repositories using Coursier. It can save these dependencies to a local cache directory, just as other build tools. The next steps are to invoke the compiler on a Scala.js project in order to produce JavaScript output, and to pipe that output to Webpack.

Note that due to the measures against COVID-19, all student projects have now been officially cancelled at EPFL. Unless this decision is at least partially reverted, this project might disappear.

Compiling Java to the Scala.js IR (student project)

@arthanzel

We are working on compiling Java source code to the Scala.js IR. This will allow to easily cross-compile Java libraries so that they can be used in Scala.js projects. Ultimately, this will also lay the groundwork for compiling Java source code to the Scala Native IR, which will also benefit the Scala Native ecosystem.

The project is subject to the same reservations with respect to EPFL decisions about COVID-19.

6th F(by) Conference, Minsk Belarus

@darjutak

Darja performed the opening keynote (video here), titled: “Scala 2 to 3 Transition: Community and Communication Management Perspective”

F(by) is a 1-day FP conference that introduced a second track this year - the Scala track. Jamie and Darja were invited to the F(by) conference to present the Scala Center’s work and Scala 3 development in front of an FP community that is less familiar with the ins-and-outs of the current situation.

They both got excellent feedback and had the opportunity to connect with different communities and hear their stories and advice.

ScalaDays 2020

@darjutak

ScalaDays Program finalised for both editions:

- 195 proposals were submitted
- The PC reviewed the submissions and held 2 PC meetings
- We accepted 66 talks

ScalaDays Seattle update:

- We made the decision to postpone the Seattle edition to May 2020 due to the new situation related to the Coronavirus outbreak;
- We reassigned 14 speakers to join the Berlin edition (by opening the 4th track in Berlin)

ScalaDays Berlin update:

- As of now, the ScalaDays Berlin is being organised as planned beginning of July 2020. We are closely observing the situation and will inform timely about any change.

Reviewing

@sjrd

As a PC member for Scala Days, we reviewed all the submissions to Scala Days 2020.

We are also reviewing Li Haoyi's book for beginners in Scala.

Local Meetups & spree

@darjutak, and the entire team

In collaboration with the local Scala communities (Lausanne and Bern), we invited Nicolas Rinaudo to give his new talk " ADTs are a simple yet surprisingly powerful concept". We used the opportunity to organise a Scala spree during the Bern Meetup (Tweet). About 20 enthusiastic developers have contributed to various libraries of the Scala ecosystem such as scala-collection-compat and Scala.js.

Internal team 2-day training

@darjutak, and the entire team

On March 9-10 we organized an internal team training to:

- Help onboard new team members
- Get to know each other in the team, and to introduce the team to main actors & their work in the ecosystem (LAMP and Dotty teams, Lightbend team, SIP members)
- Familiarizing with the projects and sharing the vision for the mid and long term Scala Center development.

New employees were asked to, by the end of the training, create their own 5 min presentation about the Scala Center and its importance and role in the Scala ecosystem.

Check our training program here.

4th SIP Retreat

@darjutak, @sjrd, Sylvie Jankow

Due to the Coronavirus outbreak, our plans were shifting day-by-day. Even though Committee members could not join us in Switzerland as planned, we held the meetings online for 3 full working days with around 12 different contributors/members/collaborators.

The proposed agenda is here.

Minutes:

- 11 March, day 1 here.
- 12 March, day 2 here.
- 13 March, day 3 in progress.

Organizers Summit

@darjutak

Due to the Coronavirus outbreak, the Scala organizers (Meetups, Conferences, ScalaBridge, sprees…) were contacting us massively, asking for help and advice. We decided to:

- Organise the Organizers Summit to discuss the topics at hand (link)
- Open an Organizers specific Discourse to be able to capture all the important knowledge and discussions in one place (in progress, should be available by Thursday, March 26)
- Raise awareness by giving a talk at "ScalaLove" online conference organised by Oli Makhasoeva, 47 Deg AB representative