You Are a
Scala Contributor

Seth Tisue @SethTisue
Scala team, Lightbend

or you can be,
if you want to.

here's how.

you are
a Scala contributor

If you open an issue
If you comment on an issue
or a pull request

If you make a pull request.
Even for a typo.

modest beginnings

Scala Programming Language / SI-725

typo in error message ("incompatibe")

Reporter:

Seth Tisue

Votes:

0

Martin Odersky added a comment - 05/Apr/08 7:24

will be fixed in next release.

use your
GitHub identity

issues
pull requests
chat
forums

help is available

chat (Gitter)
https://gitter.im/scala/contributors
(web or app)

forum (Discourse)
https://contributors.scala-lang.org
(web or email)

where to ask?

1. chat
2. forum
3. open a pull request!

half-baked pull requests are okay

You'll get help getting it across the finish line.

what's out there?
where is everything?

our GitHub organizations

scala/*

scalacenter/*

lampepfl/*

main scala/* repos

scala/scala-lang = scala-lang.org
scala/docs.scala-lang = docs.scala-lang.org

scala/scala = compiler, stdlib, spec
scala/bug = no code just bugs
scala/scala-dev = work tracking

scala module repos

scala/scala-**java8-compat**
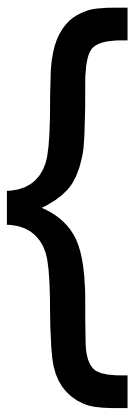scala/scala-**parallel-collections**
scala/scala-**collection-compat**
scala/scala-**async**

more soon?
scala/scala-dev#323

community
maintained
{
scala/scala-**xml**
scala/scala-**parser-combinators**
scala/scala-**swing**
scala/scala-**continuations**

the Scala websites

Scala. Do you know it?

Kind of? Still learning?

new contributors welcome at:
https://github.com/scala/scala-lang
https://github.com/scala/docs.scala-lang

the Scala websites

Do you have web skills?
Some issues that need those.

Can you put words together?
That's needed, too.

Also: **translators** wanted!

zero to website PR:

easiest: GitHub web UI
a little harder:

or `docs.scala-lang`

```
hub clone scala/scala-lang
cd scala-lang
bundle install
bundle exec jekyll serve --watch
edit foo/bar.md
```

website issues

look for the "good first issue" label

Scala modules

modules: contributing

modules are ordinary libraries

independent versioning
independent release schedule
(can be rapid)

standard tools: sbt, Travis-CI
typically: ScalaTest and/or ScalaCheck

scala/scala

scala/scala

language specification
standard library (Scaladoc)
standard library (code)
compiler

scala/scala sample month

majority of commits are from outside contributors

May 19, 2018 – June 19, 2018

Period: **1 month** ⌄

Overview

210 Active Pull Requests

0 Active Issues

⑂ **170**	⑂ **40**	⊘ **0**	ⓘ **0**
Merged Pull Requests	Proposed Pull Requests	Closed Issues	New Issues

Excluding merges, **33 authors** have pushed **224 commits** to 2.13.x and **228 commits** to all branches. On 2.13.x, **861 files** have changed and there have been **10,387** additions and **5,193** deletions.

scala/scala: language specification

Issues: "spec" label in scala/bug

"most wanted": merge in accepted SIPs

separate contributor doc: spec/README.md
separate tools: Markdown, Jekyll, MathJax

scala/scala: contributor docs

README.md
CONTRIBUTING.md

scala/scala: Scaladoc (content)

tools: Scaladoc, sbt (light)

zero to Scaladoc PR:

```
hub clone scala/scala
cd scala
edit src/library/scala/Foo.scala
sbt
> library/doc
(view in browser)
```

scala/scala: standard library

scala/scala: standard library

run tests using sbt

new tests should use
JUnit or ScalaCheck
existing tests may be partest-based.

to test performance, see
test/benchmarks/README.md

scala/scala: compiler

scala/scala: compiler

most tests are partest-based.
partest shines here.

test/files/pos, test/files/neg, test/files/run
test/files/*/*.check

Jenkins and Travis-CI run the full test suite.

compiler hacking

Jason Zaugg
(Scala team)
"Scalac Survival Guide"

http://www.youtube.com/watch?v=06oP5IXWveM

compiler hacking

Eugene Burmako's "Scala Hacker Guide"

https://scala-lang.org/contribute/hacker-guide.html

compiler hacking: ramping up

learn phases, get the big picture:
```
scalac -Xshow-phases
scalac -Xprint:all Foo.scala
```

compiler hacking: ramping up

set up IntelliJ (src/intellij/README.md)

IntelliJ debugger can attach to a running JVM

hardcore compiler hackers swear by IntelliJ

compiler hacking: ramping up

fix/improve errors & warnings

use the Scala reflection API

write a few macros

write a compiler plugin or two
(see "Compiler Plugins 101")

what about Scala 3?

what about Scala 3?

Scala 2.14 theme: align with Dotty

Scala 2.14 and Scala 3 will share:
- standard library
- compiler back end
- library ecosystem, via Tasty
- much else

work on Scala 2.14 *is* work on Scala 3

what about Scala 3?

code:
https://github.com/lampepfl/dotty
discussions:
https://github.com/lampepfl/dotty/issues
chat:
https://gitter.im/lampepfl/dotty

deciding what to work on

specific suggestions

help improve (& translate) "A Tour of Scala"

https://docs.scala-lang.org/tour/tour-of-scala.html

please help triage bugs in scala/bug

We have too many old tickets
and too many new ones.

Is it clear, actionable, still reproducible?
Is it well labeled?
Is it a "good first issue"?

Triage is needed and valued.

please help review PRs in scala/scala

Not for core maintainers only!

PR review includes asking questions.
It isn't just thumbs up/down.

PR review is usually about improving PR quality.

If you don't understand the code, consider:
testing; documentation; clarity; performance;
Is the change justified?

help the ecosystem move to sbt 1

many projects have upgraded
quite a few have not

help the ecosystem move to Scala 2.13.0-M5

every library needs help with this right now

help your favorite library
(& help their upstream dependencies)

please help
test and polish
2.13 collections

report issues you find when upgrading libraries and personal projects to M5

fix issues being reported, they are usually manageable in difficulty

deprecations, removals, modularization

adding code is human

deleting code is divine

get an early start
on Scala 2.14

- Features
  - Opaque types (#448)
  - Trait parameters (#439)
  - Type lambdas (#576)
  - Align name-based pattern matching with Dotty (#533)
  - Configurable error reporting / warning suppression (#333)

deciding what to work on

general thoughts

deciding what to work on

start small

scratch your own itch

get ideas from Gitter, Stack Overflow, ...

deciding what to work on

how long is the release cycle?

assess your personal capacity
for delayed gratification

deciding what to work on

seek especially welcoming and responsive
maintainers and communities

look for "good first issue" labels
look for a contributor's guide
ask a question in chat, try a modest sample PR
(request feedback)

staying motivated

staying motivated

Expect delays.
Expect some failures.

Pursue multiple pull requests in parallel.
Write everything down.

redefine failure as progress

Give up? Tell others what you saw.
Describe where you got stuck.

This is contributing, too.

Someone else may pick it up.
It happens all the time.

redefine failure as progress

In the end, your PR didn't make it?

You learned something.

And the PR stands as documentation of a path not taken.

This is contributing, too.
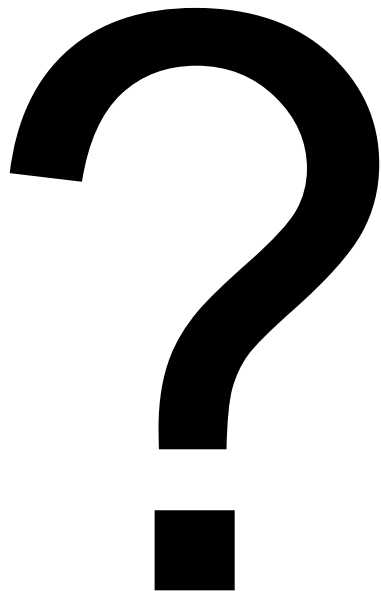
staying motivated

Seek online collaborators.
Help others as soon as you are able.

Seek local collaborators.

Scala Open Source Sprees

organized by the Scala Center / follow @scala_lang

no prerequisites

this is not a compiler internals session

pull request validation

most scala/* repos have PR validation

scala/scala: 28 minutes
(other repos are faster)

(half-baked issues are less okay)

communities vary, but:

In scala/bug and many other Scala repos, issues aren't for questions.
Use chat and forums for questions.

more scala/* repos

scala/collection-strawman

scala/scabot
scala/scala-jenkins-infra
scala/community-builds

scala/scala-dist
scala/scala-dist-smoketest
scala/make-release-notes

where to {open, find} issues

in the repo where the bug is!
ask if unsure (chat or forum)

GitHub has organization-wide search (demo)

why two repos?

Good reasons and bad.

scala/scala-lang:
why Scala, getting started,
downloads, news, community.
small enough to be high quality & fully up-to-date

docs.scala-lang.org:
language tour, feature guides, etc.

scala/scala: sbt build

since 2016

Scala is a normal
sbt project

(*) pretty normal

tips for exploring

Use tab completion in sbt

Tab completion is available in some surprising places, e.g. for test names, flags to "scala" and "partest" tasks, etc

Use tab completion in the REPL

scala/scala: Scaladoc (tool)

Scaladoc tool hacking

challenge: Scaladoc is part of the compiler

good news: you're consuming compiler data structures, not modifying

caveat: Dottydoc is coming.
Some changes should be done there instead.

deciding what to work on

watch Gitter, Stack Overflow, etc.

watch for common pitfalls and sticking points

ask yourself: could I fix that?
could I document that?

legal stuff

license: BSD
CLA: very nearly Apache 2

coming soon: Apache 2 for both

https://www.scala-lang.org/news/license-change.html

we're all in this together