Concrete next steps for Scala macros

Eugene Burmako

San Francisco
June 7, 2017

Video available at:
https://www.youtube.com/watch?v=MWqC3nMYdeo

Who am I?

- Former PhD student of Martin Odersky at EPFL (2011-2016)
- Devtools engineer at Twitter (2016-present)
- Author and maintainer of Scala macros

Who are you?

- You heard that macros are going away and are not sure about what’s going to happen next
- You heard about cool macro annotations that work in IntelliJ and are wondering when everyone will be able to use them
- You are using the cool macro annotations that work in IntelliJ and now want def macros that also work in IntelliJ
- You like Scala and/or programming languages in general

Goals of this talk

- Explain the decisions of the Scala language committee regarding macros
- Present latest developments in Scala macrology
- Provide clear understanding of the roadmap for Scala macros

Non-goals of this talk

- Introduce Scala macros
  - See "What Are Macros Good For?"
- Introduce Scalameta
  - See http://scalameta.org/
- Present latest developments in Scalameta
  - See "Semantic Tooling at Twitter"
- Gossip about Dotty
  - See "What's Different In Dotty"
  - Also see "Reconstructing Scala"

Old-style Scala macros

Old-style Scala macros

- Experimental language feature of Scala 2.10 and later
- You write functions against the reflection API (scala.reflect)
- Compiler invokes them during compilation

Why macros?

More power:

- Code generation
- Static checks
- Domain-specific languages

Problem #1: Hard to write

- Old-style macros require knowledge of Scala compiler internals
- Moreover, they are often unnecessarily verbose
- There is no easy way to troubleshoot errors in macro expansions

Problem #2: Don’t work in IntelliJ

- Old-style macros require Scala compiler internals to run
- As a result, IntelliJ cannot expand old-style macros
- Because of the same reason, Dotty cannot expand old-style macros

The future of old-style macros

- Will remain experimental till the last release in the Scala 2.x series
    - That's going to be Scala 2.15
- Will not be supported in Dotty (aka the Scala 3.x series)

Research into better Scala macros

Scalameta ca. 2015

- ScalaDays San Francisco 2015 / Amsterdam 2015
- Prototype based on the 1st generation of Scalameta (v0.0.2)
- Failed to scale to real-world macros

Scalameta ca. 2016

- ScalaDays New York 2016 / Berlin 2016
- Technology preview based on the 2nd generation of Scalameta (v1.0.0)
- Macro annotations that support Scala 2.x, IntelliJ and snapshots of Dotty
- Available at https://github.com/scalameta/paradise

liufengyun/gestalt

- Independent research at EPFL (2016-2017)
- Technology preview based on an API inspired by Scalameta
- Macro annotations and def macros that support Dotty only
- Available at https://github.com/liufengyun/gestalt

New-style Scala macros

New-style Scala macros

- Developed in May 2017
- Prototype based on a Scalameta-compatible API
- Macro annotations and def macros that support Scala 2.x
    - IntelliJ support is in the works
    - Dotty support is in the works
- Available at https://github.com/scalamacros/scalamacros

Live demo

- New-style macro annotation `@main`
- New-style materializer for `Serialize[T]`

Q: What happens next?

A: We productize scalamacros/scalamacros by the end of 2017.

Q: Who is involved?

- Eugene Burmako (Scala 2.x implementation)
- Mikhail Mutcianko (IntelliJ implementation)
- Ólafur Páll Geirsson (Dotty implementation)
- Long-term maintenance is being discussed with Lightbend, JetBrains and EPFL compiler teams

Q: What platforms are supported?

A: Scala 2.12+, IntelliJ, Dotty.

Q: Scala 2.11?

A: Probably not, would appreciate community contributions.

Q: How to migrate?

- Old-style macros that don’t use compiler internals
- scalameta/paradise macros
- Exact migration strategy is being discussed

Q: How to discuss?

A: https://gitter.im/scalamacros/scalamacros.

Q: What about Scalameta?

A: Scalameta is for devtools, Scalamacros is for macros.

Conclusion

Credits

https://github.com/xeno-by/scalamacros/blob/master/README.md

Summary

- Research in Scala macros is over, we are now productizing what we have
- Check out the code at https://github.com/scalamacros/scalamacros
- Join the discussion at https://gitter.im/scalamacros/scalamacros
- Scalameta is for devtools, Scalamacros is for macros

We’re hiring

- Speaking of Scalameta
- At Twitter, we’re building useful and robust devtools on top of Scalameta
- Check out our ScalaDays talk to learn about our agenda
- Send me an email if you’d like to join us