Microsoft

How to Keep your Spring Boot Projects Up-To-Date

Martin Lippert, Spring Tools Lead, Broadcom

Where are we?

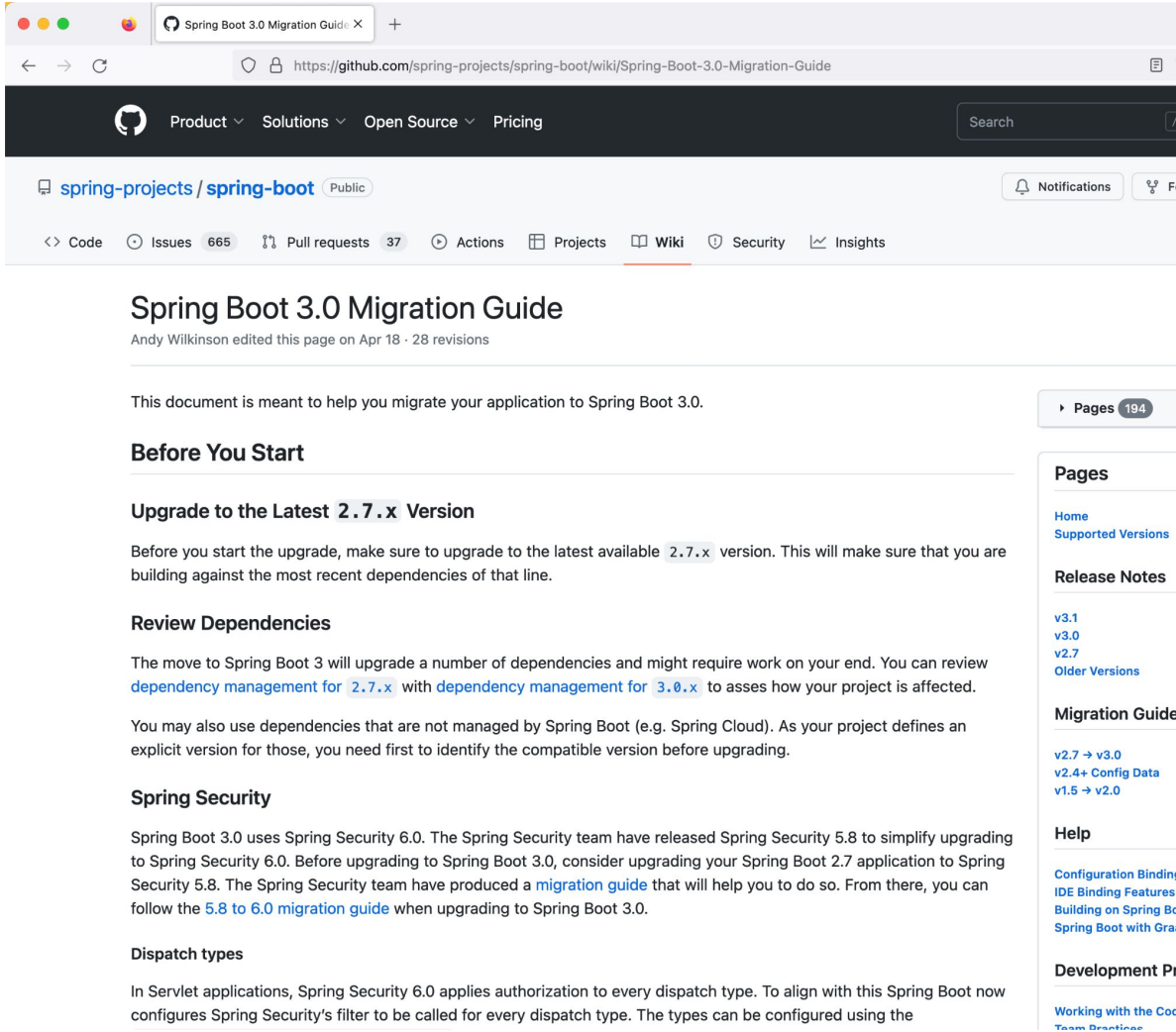
New Spring Boot releases all the time

- Many patch releases all the time
- New minor release every 6 month, sometimes new major releases
- It is super important to stay up-to-date
- But it is sometimes hard to always stay up-to-date

How to upgrade?

Release Notes + Migration Guides

- You have to read everything carefully
- You need to find out what needs to be changed for your project
- You need to apply all those changes manually



Let’s do something about this

(Spring Tools to the Rescue)

What is new to Spring Tools?

Let the user know

- Automatically check the versions that you use
- Show information about new versions and support ranges

Help the user to upgrade

- Migration guides written in “code”
- Looks at your project and applies necessary changes - AUTOMATICALLY
- Some limitations apply

Limitations

No silver bullet

- The tools apply many changes, but not all of them
- The goal is to automate as much as possible
- There is no guarantee that you are done with the upgrade afterwards - probably additional manual steps needed
    - But this will improve with every tools release - of course... 😉

Looking for feedback

Reminder: Everything that you will see is early days

- We are looking for feedback and suggestions
- If you want to get involved here, let us know

Live Demo

(Spring Version Validation & Upgrade Support)

Under the hood

What is OpenRewrite?

- *“Open-source, semantic type aware search and transformation framework.”*
- *“OpenRewrite enables large-scale distributed source code refactoring for framework migrations, vulnerability patches, and API migrations”*
- Automatically transform source code (for various purposes)
- https://docs.openrewrite.org/
- https://github.com/openrewrite

Based on initial work at Netflix to keep source code up-to-date. Sponsored now by Moderne.io. The Moderne SaaS allows organizations to run search and transformations across hundreds of repositories (millions of lines of code) simultaneously and offer a free service for the OSS community at https://public.moderne.io/

Purpose

What can OpenRewrite be used for?

- Patching CVEs
- Migrate from Java 8 to Java 11 to Java 17 to Java 21 ...
- Migrate between framework versions
- Automatically adapt code to changed APIs
- ...

- Works across various source file types (like Java Source Code, property files, YAML, other languages, etc.)

What we do inside the Spring Tools

List and run recipes from the UI

- Show the recipes that are available
- Let the user select the recipes
- Execute the recipes within the IDE

Another use case

Validations and Quick Fixes

- Let’s now push this beyond running recipes on projects
- Let’s combine this with validations/markers and code actions/quick fixes

- *This goes beyond what OpenRewrite supports out-of-the-box, but it can be added on top*

Validations and Quick Fixes

Something that looks like this

validation

code action / quick fix
(implemented as a recipe)

spring

Resources

OpenRewrite

- https://docs.openrewrite.org/
- https://github.com/openrewrite

IDE Integration

- Started as part of the Spring Tools: https://github.com/spring-projects/sts4/
- Independent of Spring Tooling in the future?
- Contact us on Twitter: http://twitter.com/springtools4/

Thank you

@martinlippert

(special thanks to Tyler van Gorder and Alex Boyko for their support and work on this)