BBD VIEW | SPRING 5

By Rory Preddy, BBD Software Developer

SPRING HAS SPRUNG IN DEVELOPMENT COMMUNITY

20 Mar 2018

Functional reactive programming (FRP) is not a new idea, but with more frameworks coming into the mainstay and reactive programming more widely adopted, developers need to stay on top of trends.

BBD's software developer whizz, Rory Preddy, explores FRP with Spring 5.

Why is this something we should be talking about?

Reactive programming revolves around models that are built to react to change. Recently Spring has made itself the de facto standard for Enterprise Java, predominantly as Oracle has 'abandoned' JavaEE i.e. Oracle is still heavily involved but in flux as they're migrating to the Eclipse foundation – with Eclipse attempting to mimic the Spring frameworks. This goes against intuition as Spring already exists and for developers looking for a stable enterprise platform, Spring 5 is it.

Has Spring become the Enterprise platform of choice?

Yes. With certain libraries only moving to Eclipse – Spring is the safer bet for developers.

Would you say that Spring has the longevity to maintain this position as the de facto choice?
Pivotal, the provider, is managing it well

as they've invested in providing training, maintenance and support. This means that not only is there someone to call to resolve any issues, they believe in the longevity of the framework.

What's new in Spring 5?

Spring 5 has three main changes:

- Full end-to-end Java 8 compatibility. You must use Java 8 and yes, we know Java 10 is releasing later in March, but that's a story for another day
- Netty: a full non-blocking web server to use in conjunction with Spring 5
- WebFlux: the standard front-end model view controller

These changes are exciting, as WebFlux is providing more control over the web request handling. This means developers can write a good non-blocking protocol, using what WebFlux is describing as request mapping. I haven't seen this before, and to be honest it's very complicated because a developer needs to write their own protocol handler but in a non-blocking fashion.

Luckily, the old-school Java developers are still catered for and you can use standard controller syntax for non-blocking handling.

What features of FRP do developers need to be aware of?

These three features are all those of reactive programming, with a functional flair added to them:

- **Composionality:** tough to say, but it basically means that services need to be composed of multiple small functions. It's bit-by-bit, high level programming with Java 8.
- **Immutability:** certain classes and objects in Java shouldn't change. This gives a developer the ability to control the state of the underlying system because you know it won't change. Java has traditionally been terrible at immutability, as there aren't whole collections that are immutable. To get around this in Spring 5, Java has provided base objects that remain immutable. The jury is still out as to whether this will work well for Java developers.
- **Inherent parallelisability:** the ability to scale in parallel, is guaranteed with the Spring 5 framework when using WebFlux. This is an excellent thing because with the cloud the ability to scale is very important, but scaling gracefully is even more so.

How could a Java developer scale gracefully using Spring 5?

One of the features that allows these reactive services to scale gracefully is the ability to use back pressure, and WebFlux now has this built in. Back pressure has actually been around for a

long time, despite many developers not knowing exactly it is.

What this means for the developer is that if you are not coping, you can tell the system to slow down. If you want to throw the data away, you do so at your own discretion. It's about consuming what you want and how you want, with the system not mandating that you consume at a speed that you are uncomfortable with. A reactive stream's main purpose is to allow the developer to control how fast or how slow the publisher will produce data.

Back pressure works both ways – if you need to speed up certain containers or servers, you can do so, and gracefully. This all results in developers now being able to send big data to a consumer across a standard rest endpoint, that gracefully expands to allow the consumption of that data. This brings a

fascinating 'near' data science element to Spring 5 and reactive programming.

At BBD, we have a heavy Java investment, so as JavaEE becomes less of a choice, Spring 5 is on track to become our go-to platform. This could be the route many software development organisations choose and Spring 5 developers need to be familiar enough with this to decide whether to go reactive with it or not

About Rory Preddy

Part of BBD's R&D team, Rory's 20 years' experience in the ICT sector gives him an unparalleled insight into industry movements. His mother tongue is Java, but he has a deep love affair with AWS. Rory is also the organiser for the Jozi-JUG, Kotlin and AWS user groups.

About BBD

A provider of custom software development and application design solutions, BBD's 34 years of technical and developer expertise spans the banking, insurance, telecommunications, education and public sectors. Employing over 700 highly skilled, motivated and experienced IT professionals - BBD is South Africa's largest independent custom software development company.