The Essence of Reactive Programming in Java

Vladimir Sinkevich
Head of Java Development, ScienceSoft

SOFTWARE DEVELOPMENT

Published: Jul 8, 2018

Table of contents

- A brief explanation of RP
- A really simple explanation of RP
- Reactive programming vs reactive systems
- The value of reactivity
- Reactivity in Java
- Business benefits of RP
- How to introduce RP into your system
- RP use cases
- How to start writing reactive code
- Conclusion

Reactive programming (RP) is not something new and cutting-edge when it comes to application development. Chances are, you've already heard about it. The term was initially introduced to the IT world in the 1960-s and ever since much has been said and written in its regard. Unfortunately, as it often happens, the new concept quickly fueled a set of misinterpretations around itself, and still continues doing so today. The Reactive Manifesto of 2014 that introduced **'reactive systems'** and their four 'sacred principles' messed everything up even more. So, let's try to clarify what is what, and understand why and where we need reactive programming in Java application development (if we really do).

#1 What is reactive programming in a few words?

Reactive programming is a programming paradigm that deals with asynchronous data streams (sequences of events) and the specific propagation of change, which means it implements modifications to the execution environment (context) in a certain order.

#2 What is this 'specific propagation of change'?

Here's a real-life example. Say, it's Friday and John wants to spend this evening with his friend Bob, scarfing pizza and watching one of the Star Wars episodes. Let's outline the options he has.

1. John finishes his work. Then goes and orders the pizza, waits till it's done. Then picks up his friend. And finally (with Bob and pizza) makes it home and gets down to the movie. It will be the **sync** approach and it will be way too long, so that probably John will have wanted to call the thing off by that time.
2. John orders his pizza online, phones Bob, invites him to come. He heads home, has his pizza delivered and starts watching the movie (and eating the pizza) without waiting for Bob to show up. That is what can happen with the **async** approach.
3. John orders pizza, phones Bob, invites him to come, heads home, and gets his pizza delivered. But this time, he waits until Bob comes and only after that he turns the movie on. This is what the **reactive approach** is about. You wait till all async actions (changes) are completed and then proceed with further actions.

#3 Are reactive programming and reactive systems the same thing?

No, they are not. Though often used interchangeably, the terms are not exactly synonymous and reflect different things.

Reactive systems represent the next level of 'reactivity'. This level implies specific **design** and **architectural** decisions that allow building resilient, flexible, and responsive applications.

You don't have to use reactive programming in reactive systems, but it's a good idea to do so, as the combination brings even more benefits to your application, as make them even more loosely coupled, allow more efficient use of resources, make them more responsive and ensure lower latency.

#4 Why do we need 'reactivity' in Java?

When it comes to **huge volumes of data** or **multi-userness**, we often need asynchronous processing to make our systems fast and responsive. In Java, a representative of old object-oriented programming, asynchronicity can become really troublesome and make the code hard to understand and maintain. So, reactive programming is especially beneficial for this 'purely' object-oriented environment as it simplifies dealing with asynchronous flows.

#5 How do I go reactive in Java?

With its latest releases (starting with Java 8), Java itself has made some attempts to introduce *built-in reactivity*, yet these attempts are not very popular with developers to date. But there're some live and regularly updated *third-party implementations* for reactive programming in Java that help to save the day and thus are particularly loved and cherished by Java developers.

RxJava was the first Reactive Extension API specific for the Java platform. It works with Java 6 and provides an opportunity to write asynchronous, event-based programs for both Java and Android Java, which is very convenient.

Spring Reactor is another framework for Java from Spring developers. It is quite similar to RxJava but has simpler abstraction. The framework has managed to win popularity due to the possibility to leverage benefits of Java 8.

#6 What do I get with RP in real life?

Increased performance – due to the possibility to handle huge volumes of data in a quick and stable way.

Improved UX – due to the possibility to keep the application more responsive to its user.

Simplified modifications and updates – due to more readable and easier to predict code.

#7 Do I have to make it all reactive?

'Reactive' components can be smoothly introduced to an application just as its part, so it's unnecessary to change the whole project programming model, sabotage other trusted programming styles, cling to 'reactivity' completely and introduce unnecessary complexity. Measure is treasure. For example, if it's just a simple web site – there's hardly any need to write it reactive. But as soon as you want to upgrade it and introduce recommendation system – reactive code will be a good idea here to deal with the high load of big data.

#8 When to use RP?

Going reactive provides an elegant solution when it comes to specific types of **high-load** or **multi-user applications**:

- *Social networks, chats*
- *Games*
- *Audio and video apps*

And to the following components of **any application type**:

- *Server code that serves highly interactive UI elements*
- *Proxy servers, load balancers*
- *Artificial intelligence, machine learning*
- *Real-time data streaming*

Would like to see a real life example? Check our recent Java project where we applied reactive programming.

VIEW THE PROJECT

#9 When not to use RP?

Simply put, do not try to apply to RP where there is no need to, e.g. where there is no 'live' data, high load, or a large number of users who change data simultaneously.

#10 What does it take to start reactive programming in Java?

Unfortunately, simply studying the theory and downloading the frameworks will have little impact. It takes a good amount of time, effort and practical experience for a Java developer to make the imperative mind get used to a different level of abstraction. That's why, in case you decide to adopt 'reactivity', it is better to turn to a good consultant with true understanding of the approach who will be able to detect where reactive components would be a benefit for your system and how you can seamlessly introduce them in it.

The crux

'Reactive programming' is not a buzzword anymore but still not clearly defined. We hope that next time you come across this term it will not bewilder you as now you know that it's just another coding style that centers on the effective management of changes with async data streams. RP is of particular importance for object-oriented Java where asynchronicity often results in the code that is hard to understand and maintain. RP can be challenging as it requires a Java developer to wrap the mind around a completely new programming style. However, in case of success, no efforts will be wasted as it allows applications to be more resistant to high load and greatly improves UX making them more responsive. Nevertheless, it's also important not to *overreact*. The thoughtless use of the reactive approach with no real need will just ruin an application with unnecessary complexity.

Plan to build a Java-based project? Leverage the experience of our dedicated architects and developers to discover the full potential of Java environment.

I WANT TO LEARN MORE →

10 Comments Login ▾

Join the discussion…

LOG IN WITH OR SIGN UP WITH DISQUS

Name

12 Share Best Newest Oldest

Anna Zyatikova
4 years ago
Well, that's probably the best explanation of a complex technical thing with simple language that I've seen. Ever. Thank you so much!
I just loved the John-Bob-pizza-Star-Wars example: it all becomes so clear when you describe such difficult things using everyday situations.
And what is even more pleasant, the article doesn't take up too much time to read (sometimes such explanations tend to be rather wordy).
6 1 • Reply • Share ›

@@@ → Anna Zyatikova
2 years ago
haha same with me
0 0 • Reply • Share ›

Yehia Serw
4 years ago
Is reactive pardigam is eligable for search engine application based on elasticsearch?
1 1 • Reply • Share ›

Stav Alfi → Yehia Serw
3 years ago
sure why not. almost anything can be written in reactive code. the only thing I wouldn't it to use code with alot of mutations that has been written that way for proven optimizations reasons: manging bits in the java bits collection (for example).

RP is a different way (and ,personally, i think a better one) to write an async/sync software.
1 0 • Reply • Share ›

Sergey Kozlov
3 years ago
RP is not much useful in stateless server-side applications, isn't it? In such a case, good old horizontal scaling looks way easier and cheaper, than supporting complicated code.
0 0 • Reply • Share ›

omar rebollo
3 years ago
Simple and concise, thank you. Very good explanation.
0 0 • Reply • Share ›

Jaydeep Ramesh Deshmukh
3 years ago
@Uladzimir Sinkevich : This is great article. You have explained each concept/doubt with such an absolute clearity that one read is enough to clear our mind of all doubts.
0 0 • Reply • Share ›

white devil
4 years ago
Excellent Article !!!!
0 0 • Reply • Share ›

Elizabeth Barkaline
5 years ago
I've never heard about PR before I read this very informative article. However, while reading, I realized I've actually "met" PR. Moreover, I deal with it on a daily basis. It's AJAX (Asynchronous JavaScript and XML). AJAX allows web pages to update without reloading. This way, only a small party of content on the page change. The majority of pages and web applications using AJAX are data-intensive (for example, news websites or social media); so loading the entire page over and over again would hinder the UX. AJAX, on the contrary, offers swift reloading without unnecessary waiting. Hope this small insight will make PR even more understandable for everyone.
0 0 • Reply • Share ›

Nizar → Elizabeth Barkaline
4 years ago
This is completely wrong, Ajax is not and has nothing to do with Reactive Programming.
1 0 • Reply • Share ›

Subscribe Privacy Do Not Sell My Data

ScienceSoft Professional Software Development

5900 S. Lake Forest Drive Suite 300, McKinney, Dallas area, TX 75070

+1 214 306 6837
+1 972 454 4730

contact@scnsoft.com

Financial Times badge

35 REVIEWS

ScienceSoft. Computer