Spring 5 +Reactive programming Features : https://springframework.guru/spring-web-reactive/

https://dzone.com/articles/spring-data-mongodb-with-reactive-mongodb

# Spring Data MongoDB With Reactive MongoDB

### With the rise in popularity of NoSQL databases, MongoDB has rapidly gained popularity. See reactive programming features in Spring Framework 5 and Spring Data MongoDB.

**SPRING WEB REACTIVE**

An exciting feature in [Spring Framework 5](https://springframework.guru/what-is-new-with-spring-framework-5/) is the new Web Reactive framework for allows reactive web applications. Reactive programming is about developing systems that are fully reactive and non-blocking. Such systems are suitable for event-loop style processing that can scale with a small number of threads.

Spring Framework 5 embraces Reactive Streams to enable developing systems based on the [Reactive Manifesto](http://www.reactivemanifesto.org/) published in 2014.

The Spring Web Reactive framework stands separately from Spring MVC. This is because Spring MVC is developed around the Java Servlet API, which uses blocking code inside of Java. While popular Java application servers such as Tomcat and Jetty, have evolved to offer non-blocking operations, the Java Servlet API has not.

From a programming perspective, reactive programming involves a major shift from imperative style logic to a declarative composition of asynchronous logic.

**ACID** :

**Atomic** – In a transaction with two or move pieces of information, either all the information is committed to save, or none is saved.  Essentially, an “all or nothing” rule is followed.

**Consistent** – The data saved can’t violate any of the database’s integrity.  Interrupted changes are rolled back to ensure the database is placed in a state prior to the change.

**Isolation** – The transaction in question is not affected by any other transactions taking place.  This avoids “mid-air collisions.”

**Durable** – Once the transaction is committed, any failure or system restart, returns the data in a correct state.  Stated another way, once a transaction is committed, it will remain so, regardless of a subsequent system failure.

**MongoDB :**

MongoDB 4.0 will add support for multi-document transactions, making it the only database to combine the speed, flexibility, and power of the document model with ACID data integrity guarantees. Through snapshot isolation, transactions provide a globally consistent view of data, and enforce all-or-nothing execution to maintain data integrity.

**MongoDB** (de l'anglais *[humongous](https://fr.wiktionary.org/wiki/humongous" \o "wikt:humongous)* qui peut être traduit par « énorme ») est un [système de gestion de base de données](https://fr.wikipedia.org/wiki/Syst%C3%A8me_de_gestion_de_base_de_donn%C3%A9es) [orientée documents](https://fr.wikipedia.org/wiki/Base_de_donn%C3%A9es_orient%C3%A9e_documents), [répartissable sur un nombre quelconque d'ordinateurs](https://fr.wikipedia.org/wiki/Scalability) et ne nécessitant pas de schéma prédéfini des données. Il est écrit en [C++](https://fr.wikipedia.org/wiki/C%2B%2B). Le serveur et les outils sont distribués sous [licence AGPL](https://fr.wikipedia.org/wiki/GNU_Affero_General_Public_License), les pilotes sous [licence Apache](https://fr.wikipedia.org/wiki/Licence_Apache) et la documentation sous [licence Creative Commons](https://fr.wikipedia.org/wiki/Licence_Creative_Commons)[2](https://fr.wikipedia.org/wiki/MongoDB#cite_note-licensing-2). Il fait partie de la mouvance [NoSQL](https://fr.wikipedia.org/wiki/NoSQL" \o "NoSQL).