HOW TO LEVERAGE REACTIVE PROGRAMMING WITH R2DBC ASYNC JAVA CONNECTOR

DESCRIPTION

This developer guide provides an introduction to the MariaDB R2DBC connector and demonstrates how it integrates into a Java Spring application.

Not too long ago, a reactive variant of the JDBC API was released, known as Reactive Relational Database Connectivity (R2DBC). While R2DBC started as an experiment to integrate SQL databases into systems that use reactive programming models, it now has robust specification to manage data in a fully reactive and completely non-blocking fashion.

Using an Apache Maven-based Java solution, this guide provides information on how to take advantage of the MariaDB R2DBC connector within a simple TODO application. From there, you'll be able to examine how to take advantage of crucial concepts, like event-driven behavior and backpressure, that enable fully reactive, non-blocking interactions with a relational database.

GETTING STARTED

The code for the application lives in a GitHub repository called "dev-example-todo."

https://github.com/mariadb-corporation/dev-example-todo

Once you've accessed the repository, follow the steps to build and run the TODO application. Ultimately, the application consists of two parts:

1. A front-end web project using React.js: Instructions for building and running the front-end.
2. A back-end API Spring Boot project, written with Java, utilizing the MariaDB R2DBC connector and the R2DBC Spring Data library: Instructions for building and running the back-end.

KEY MARIADB RESOURCES

MariaDB SkySQL

MariaDB transactional storage

MariaDB Enterprise Server 10.5

MariaDB R2DBC connector