EP 12 Serverless Architecture with Spring Cloud Function: An Introduction and Practical Implementation

JAVINPAUL AND SOMA

AUG 05, 2023 • PAID

Hello friends,

How are you doing? This week we will cover a few interesting Java and Spring Boot Interview questions, but before that, I have an exciting gift for you.

30% discount on my Java and Spring Interview prep books:

1. Grokking the Java Interview
2. Grokking the Spring Boot Interview
3. Spring Framework Practice Questions

You will love these resources if you are preparing for Java developer interviews.

Now, in today's free issue, we will see a couple of important topics like

1. **Serverless architecture in Java using Spring Cloud Function**
2. **Demystifying the Volatile Keyword in Java**
3. **Demystifying HashMap in Java: How it Works Under the Hood**
4. **10 React Features That Make Frontend Development Easier**
5. **How to create System Diagrams using ChatGPT - Part 2 (video)**

1. Serverless architecture in Java using Spring Cloud Function

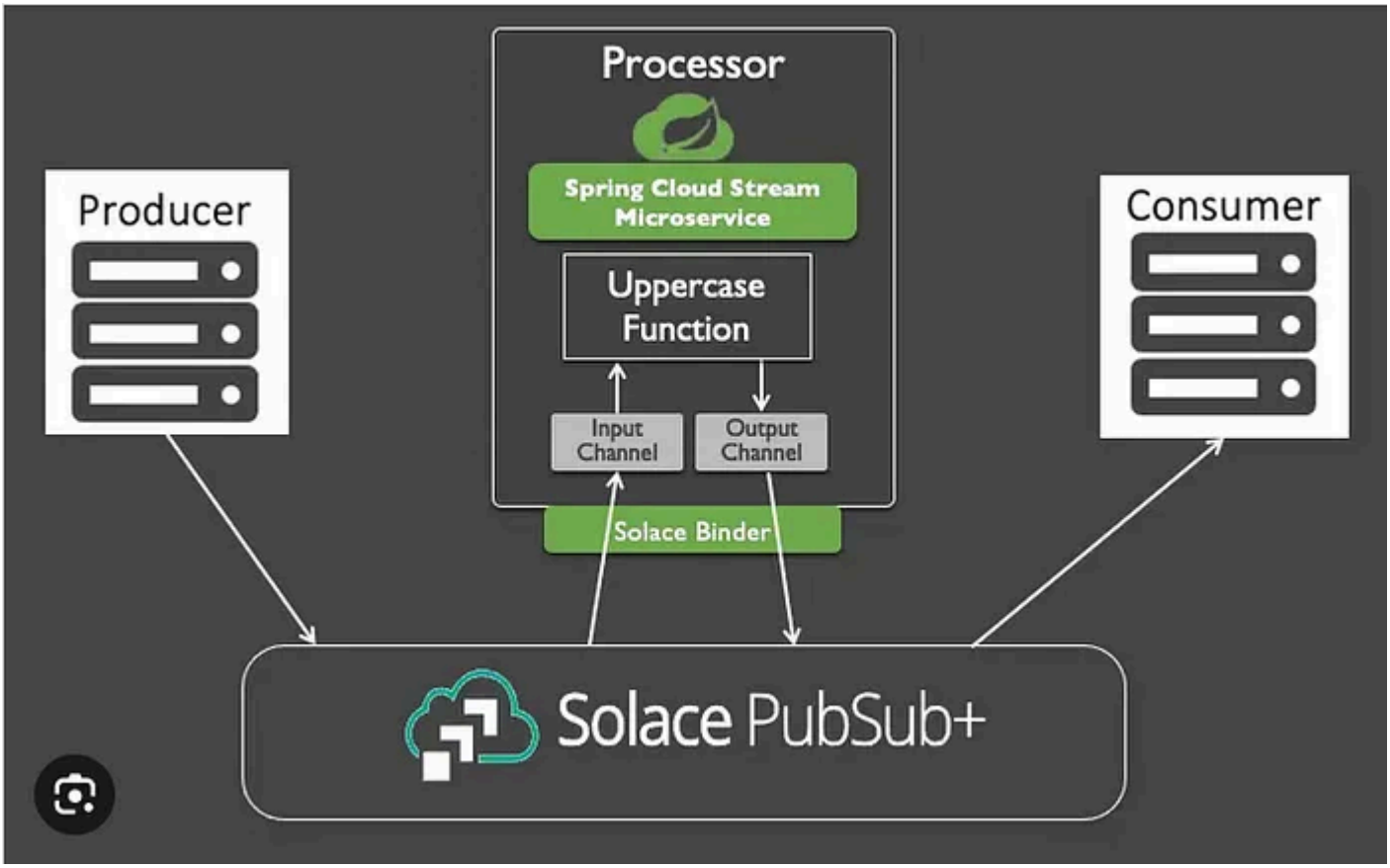
Serverless architecture is a cloud computing paradigm where developers can focus solely on writing application code without worrying about managing servers. It allows for auto-scaling and reduces operational overhead as cloud providers manage the infrastructure.

Serverless architecture is based on the "`Function-as-a-Service`" (FaaS) model, where developers write discrete functions to perform specific tasks. These functions are triggered by events and executed in response, scaling automatically based on demand.

The main advantages of serverless architecture include cost-efficiency, scalability, and reduced maintenance efforts.

Here is an example of Serverless architecture, where we have used a function to convert the word into Uppercase:

Read the full article - Demystifying Serverless: Benefits and Challenges Explored



2. Demystifying the Volatile Keyword in Java: Managing Thread Visibility and Ordering

Managing shared data between multiple threads can be complex in concurrent programming. Java provides various mechanisms to ensure thread safety and prevent race conditions. One such tool is the `volatile` keyword **plays a critical role in controlling how threads access shared variables.**

In Java, the `volatile` keyword is used to declare variables as "volatile," indicating that multiple threads may modify their value and should **always be read directly from the main memory rather than from the thread's cache.**

This *ensures that changes to the variable made by one thread are immediately visible to other threads,* preventing potential visibility issues in multi-threaded environments.

The Need for Volatility

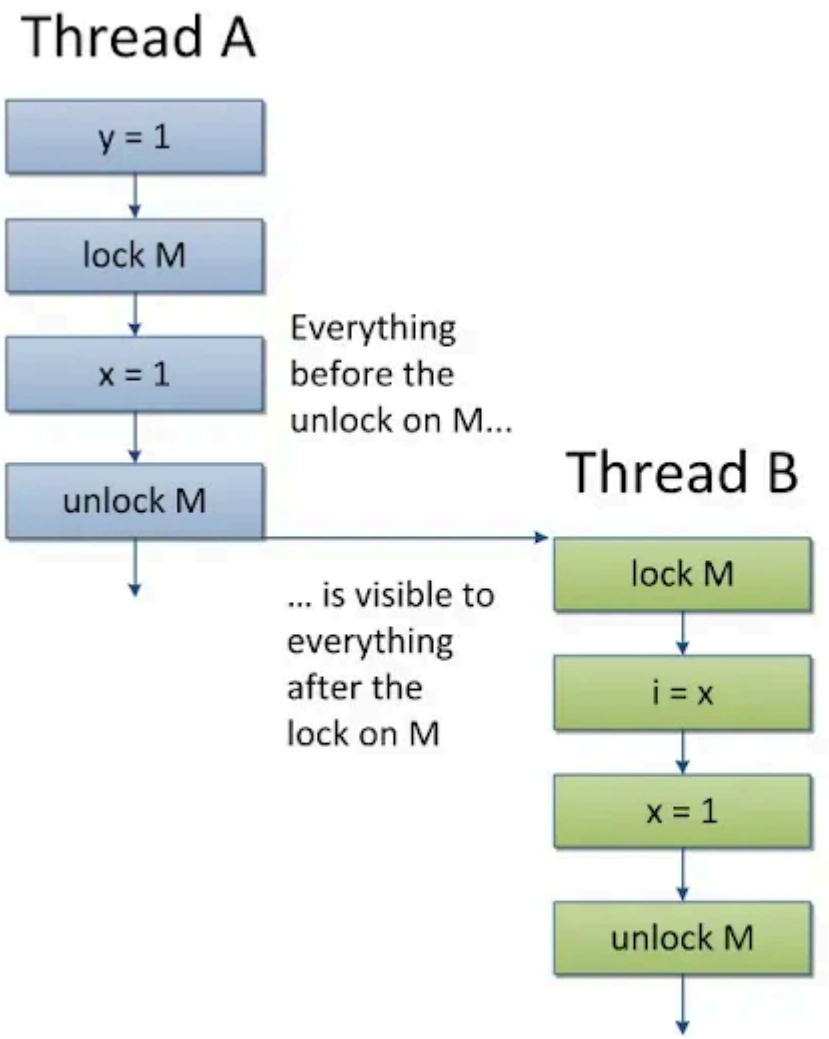
When multiple threads operate on shared variables concurrently, each **thread maintains its copy of the variable in its local cache**. This optimization, known as "thread caching," can improve performance. However, it can lead to problems when multiple threads try to access the same shared variable.

Consider a scenario where one thread updates a shared variable, but the update is not immediately visible to other lines due to caching.

If another thread relies on the updated value without proper synchronization, it may work with a stale value, leading to incorrect behavior and unpredictable results.

The `volatile` keyword helps solve this problem by immediately ensuring that any change to a volatile variable is visible to all threads.

Read the full article - Demystifying the Volatile Keyword in Java: Managing Thread Visibility and Ordering.



3. Demystifying HashMap in Java: How it Works Under the Hood

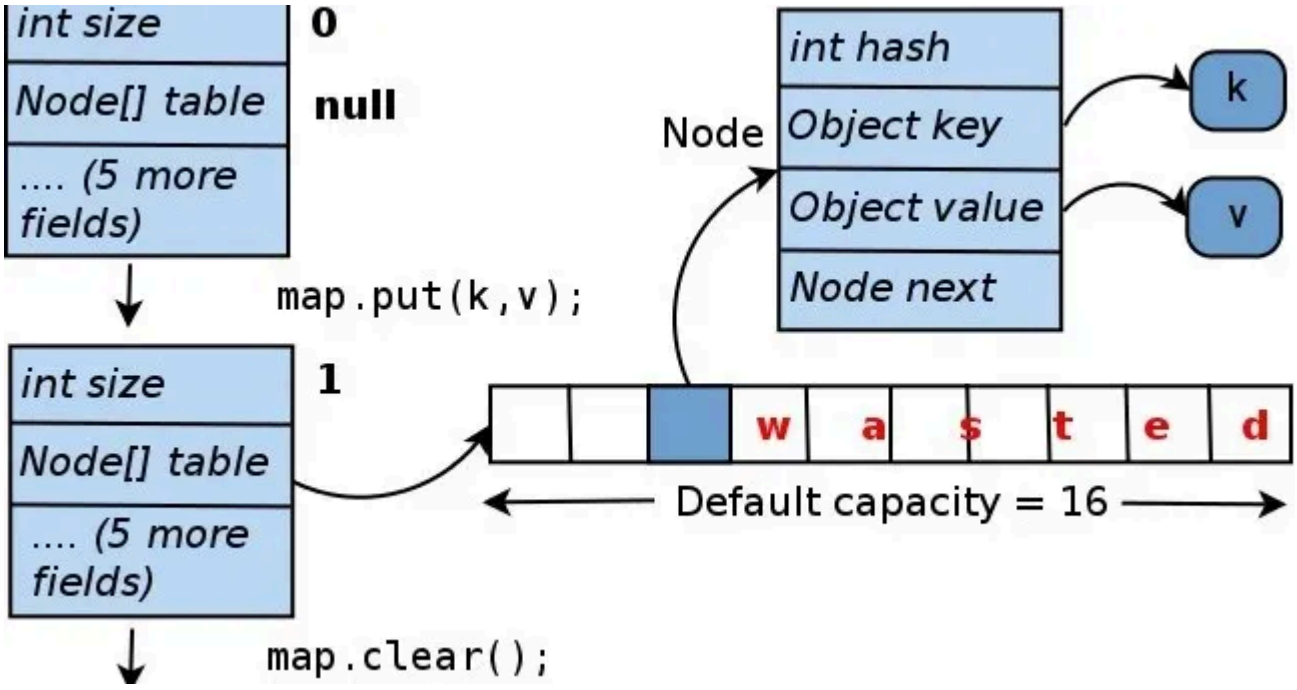
One of the popular topics of Java interviews is the Java collection framework, and on that framework, HashMap is probably the most asked class. I have personally seen questions like How HashMap works. How the get method of HashMap works, how the put process of HashMap works, HashMap is thread-safe, etc, in my career many times.

In this article, I will try to share everything I know about HashMap so that you can answer such questions with confidence.

To start with, HashMap is one of Java's most widely used data structures, providing an efficient way to store and retrieve key-value pairs. It belongs to the Java Collections Framework and implements the Map interface.

It represents a Hash table data structure, also known as a dictionary in Python or an associative array in Perl. It allows you to map one value to another to get an employee object by storing it against employeeId. The beauty of this is that it will enable operation in constant time, which means it doesn't matter how many records you need to store; you will also get the data continuously. There are some edge cases also, which we will discuss in this article.

Read the full article - Demystifying HashMap in Java: How it Works Under the Hood



4. 10 React Features That Make Frontend Development Easier

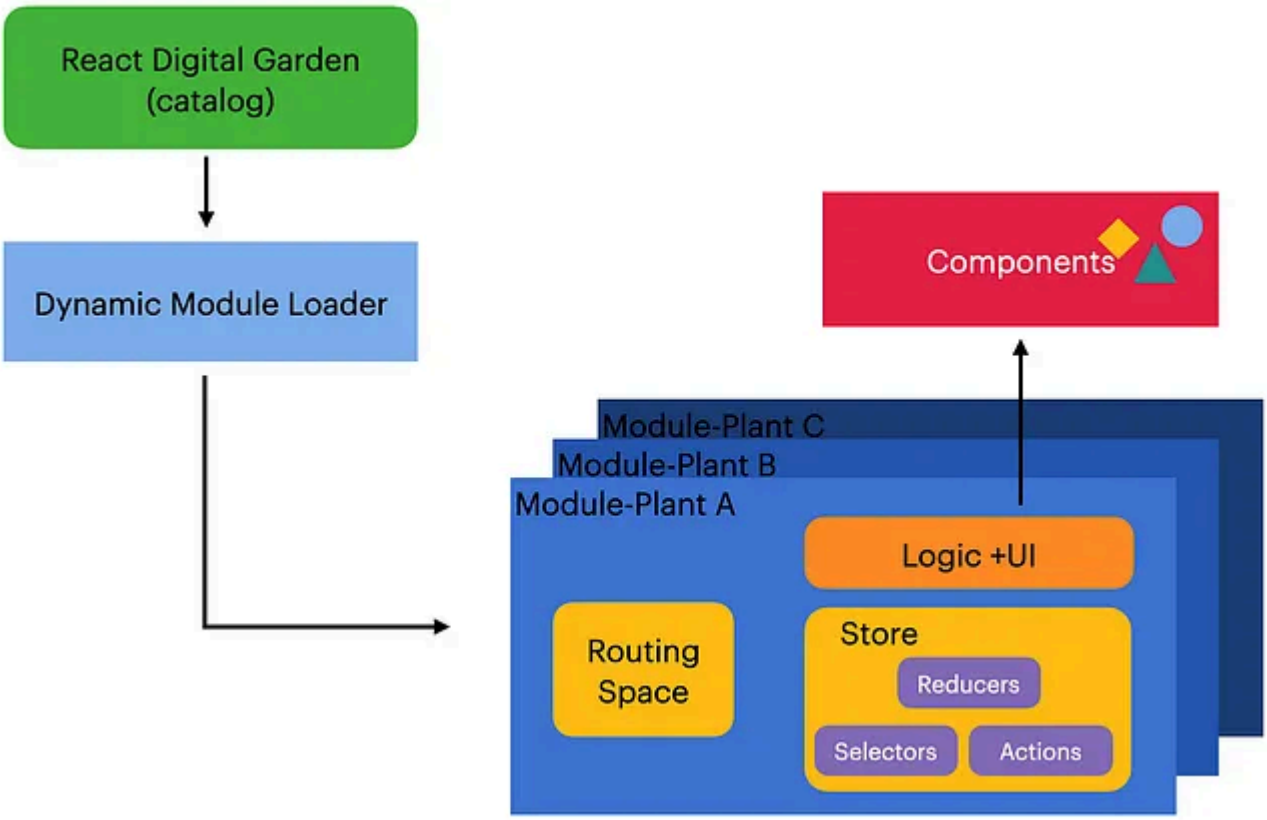
With its component-based architecture and extensive feature set, React has become the go-to framework for front-end developers. In this article, we will explore 10 React features that make frontend development more manageable and efficient.

1. Component-Based Architecture

This is probably the biggest reason why I love React and why React.js is so much popular. React's component-based architecture promotes **modularity** and **reusability**, allowing developers to break down complex user interfaces into smaller, manageable components.

Components encapsulate their own logic and rendering, making reasoning and maintaining code easier. Reusable components save development time and effort by allowing developers to leverage existing features across projects.

Read the full article - 10 React Features That Make Frontend Development Easier



5. How to create System Diagrams using ChatGPT - Part 2 (video)

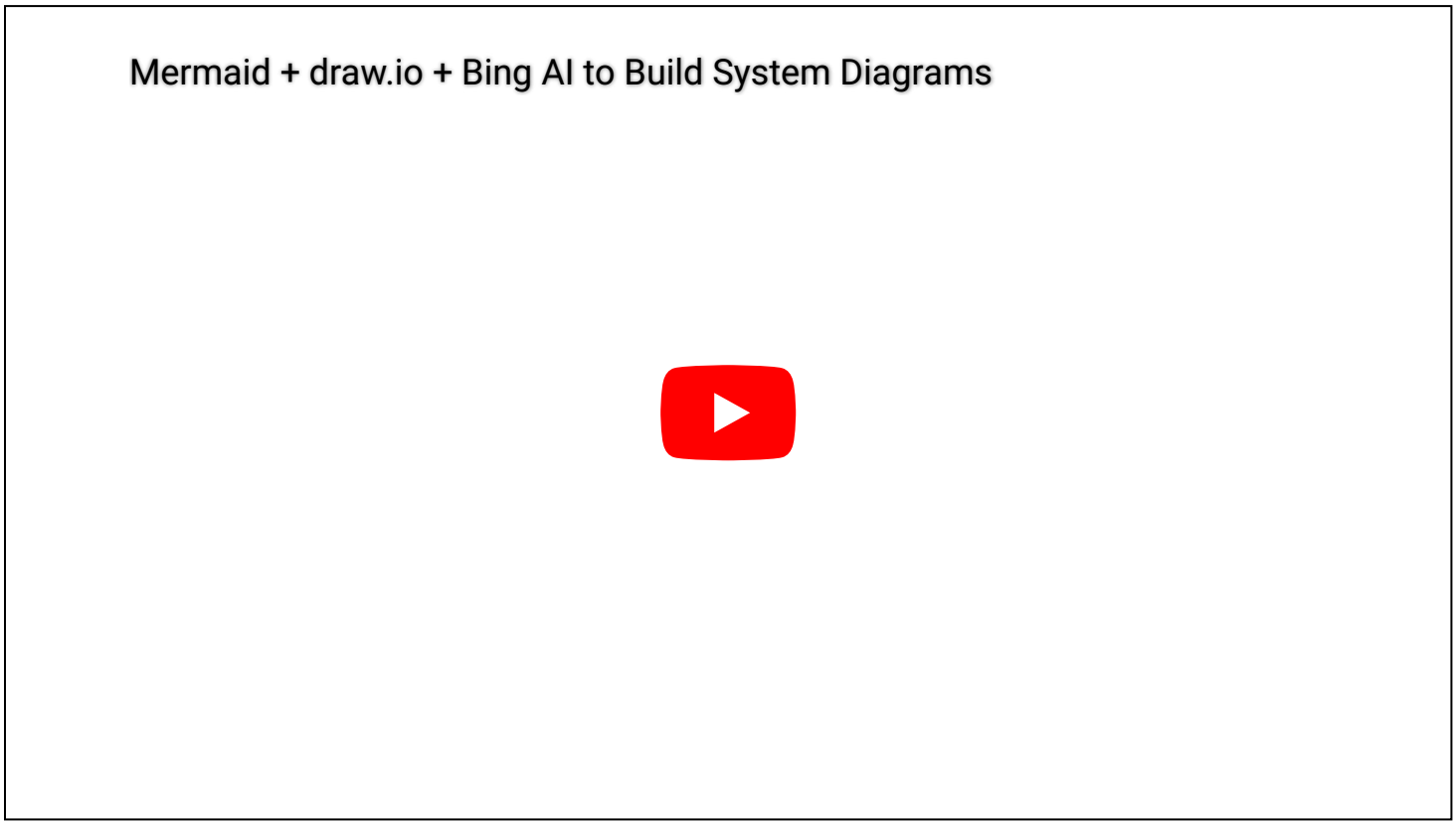
Creating software diagrams like high-level designs, flow charts, and workflows is an essential skill for senior developers and software architects but at the same time, its hard.

This exciting video delves into the powerful combination of Mermaid, draw.io, and Bing AI to revolutionize how you create system diagrams. Mermaid, a simple markdown-like syntax, allows you to generate charts using text-based inputs effortlessly.

Draw.io, a popular diagramming tool, provides a user-friendly interface to bring your Mermaid diagrams to life with customizable shapes, colors, and connections. But what sets this tutorial apart is the integration of Bing AI, which enhances your diagramming experience with intelligent suggestions, auto-layout, and object recognition.

You can watch it on YouTube, and if you haven't subscribed to our channel, then you can also subscribe here - https://bit.ly/2H20FjD

And please like and share, and if you have any requests feel free to comment.



My Books and Courses

If you are new here and want to support me, you can check out my books and courses; you can use the unique discount code "friends20" to get a 20% discount on my books.

1. **Grokking the Java Interview Questions**
   It covers:
   ➡️ OOP
   ➡️ Thraed
   ➡️ Collection
   ➡️ Stream
   ➡️ Lambda
   ➡️ JVM
   ➡️ Design Patterns
   ➡️ Generics
   Download the FREE PDF Sample here - https://bit.ly/3PywdMc
2. **Grokking the Spring Boot Interview**
   It covers:
   ➡️ Core Spring
   ➡️ Spring Boot

➡️ Spring MVC
➡️ Spring Data JPA
➡️ Spring Cloud
➡️ Security
Download the FREE PDF Sample here - https://bit.ly/3PywdMc

Courses:

I have also created multiple courses for IT certification on Udemy, like Java, Spring, Azure, and AWS Cloud certifications; you can use them to prepare better for your IT certifications.

1. Java SE 17 - 1Z0-829 Certification
2. Java SE 11 - 1Z0-819 Certification
3. Spring Professional Certification
4. Java Fundamentals 1Z0-811 Certification
5. Azure Fundamentals Certification
6. AWS Cloud Practitioner certification
7. Java EE Application Developer Certification

4 Likes · 2 Restacks

← Previous

Next →

A guest post by

Soma

Java and React Developer

Subscribe to Soma

Comments

Write a comment...