The QuoteServices App Case Study: Test Driver Implementation & Performance

Douglas C. Schmidt
d.schmidt@vanderbilt.edu
www.dre.vanderbilt.edu/~schmidt

Professor of Computer Science
Institute for Software Integrated Systems
Vanderbilt University
Nashville, Tennessee, USA
Learning Objectives in this Part of the Lesson

• Understand the implementation of a test driver program that calls the methods in the QuoteServices Client-related classes & prints the results & performance

QuoteDriver

Microservice-based Quotes App

Gateway

Handey Application

Eureka

Zippy Application

HTTP GET/POST requests/responses

The performance of both sequential & parallel computations is analyzed
Implementing the QuoteDriver Class
Implementing the QuoteServicesTest Driver

```java
@SpringBootApplication
public class QuoteDriver {
    /**
     * Number of quotes requested.
     */
    private final static int sNUMBER_OF_QUOTES_REQUESTED = 10;

    /**
     * This object connects [QuoteDriver] to the [QuoteClient]. The [Autowired] annotation ensures this
     * field is initialized via Spring dependency injection, where an
     * object receives another object it depends on (e.g., by creating
     * a [QuoteClient]).
     */
    @Autowired
    private QuoteClient quoteClient;

    /**
    */

    public static void main(String[] args) {
        new CommandLineRunner() {
            @Override
            public void run() {
                runQuotes();
            }
        }.run();
    }

    private void runQuotes() {
        for (int i = 0; i < sNUMBER_OF_QUOTES_REQUESTED; i++) {
            quoteClient.getQuote();
        }
    }
```
End of the QuoteServices App Case Study: Test Driver