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

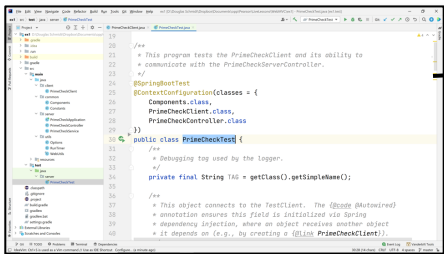**Institute for Software Integrated Systems**

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
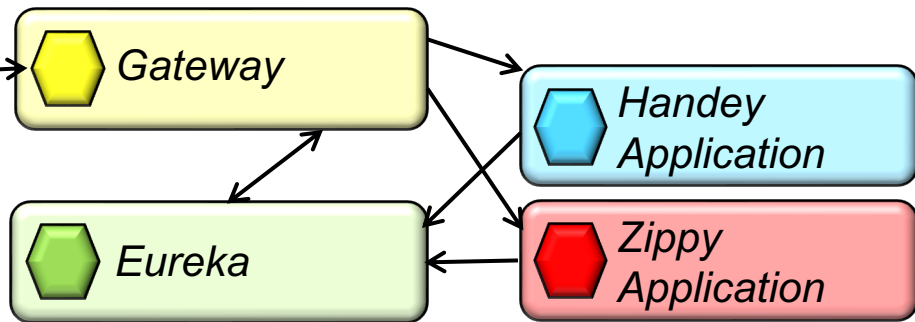

QuoteDriver

Microservice-based Quotes App

Gateway

Handey Application

Eureka
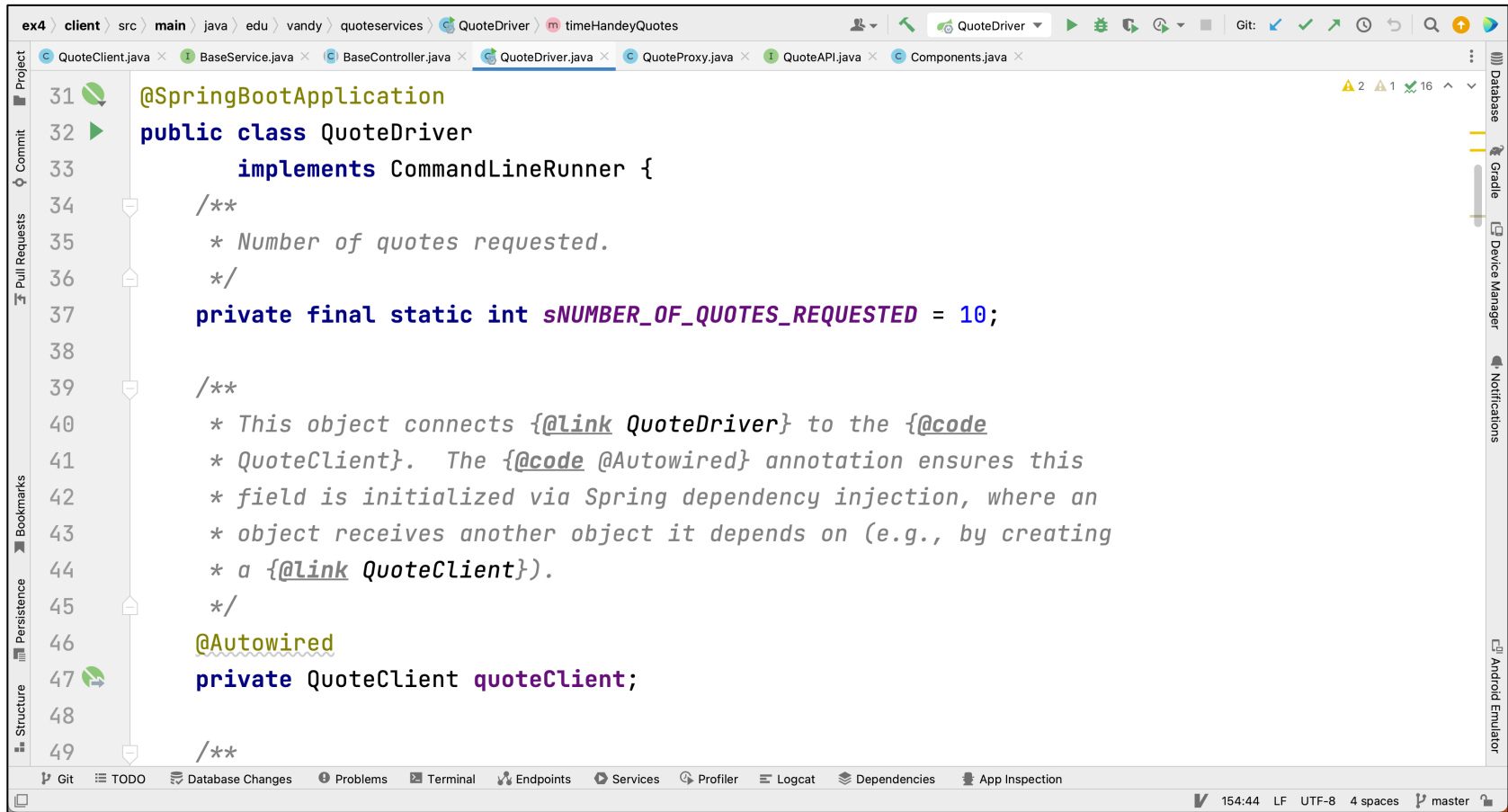
Zippy Application

HTTP GET/POST requests/ responses

The performance of both sequential & parallel computations is analyzed

Implementing the QuoteDriver Class

Implementing the QuoteServicesTest Driver

```
31   @SpringBootApplication
32   public class QuoteDriver
33           implements CommandLineRunner {
34       /**
35        * Number of quotes requested.
36        */
37       private final static int sNUMBER_OF_QUOTES_REQUESTED = 10;
38
39       /**
40        * This object connects {@link QuoteDriver} to the {@code
41        * QuoteClient}.  The {@code @Autowired} annotation ensures this
42        * field is initialized via Spring dependency injection, where an
43        * object receives another object it depends on (e.g., by creating
44        * a {@link QuoteClient}).
45        */
46       @Autowired
47       private QuoteClient quoteClient;
48
49       /**
```

See github.com/douglascraigschmidt/LiveLessons/tree/master/WebMVC/ex4

End of the QuoteServices App Case Study: Test Driver