

The QuoteServices App Case Study: Implementing the QuoteDriver

Douglas C. Schmidt

d.schmidt@vanderbilt.edu

www.dre.vanderbilt.edu/~schmidt

Professor of Computer Science

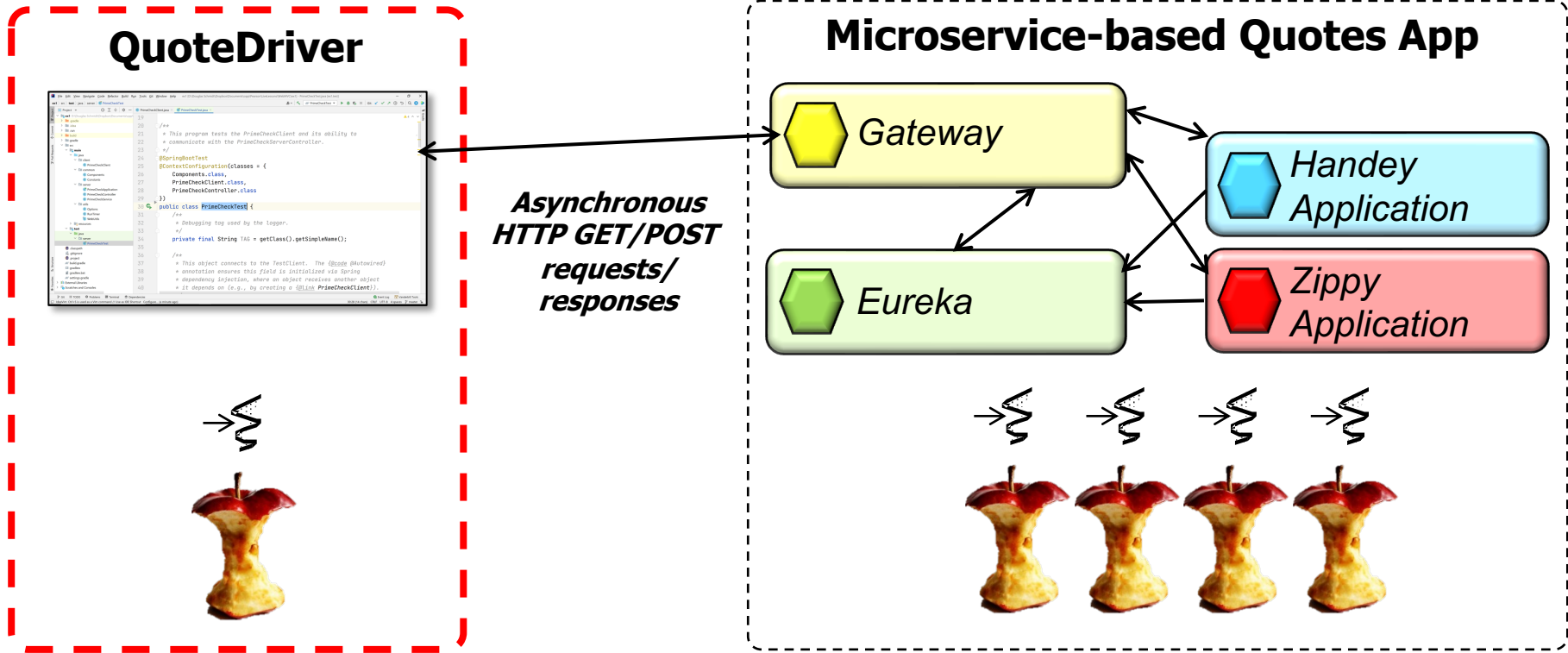
**Institute for Software
Integrated Systems**

**Vanderbilt University
Nashville, Tennessee, USA**



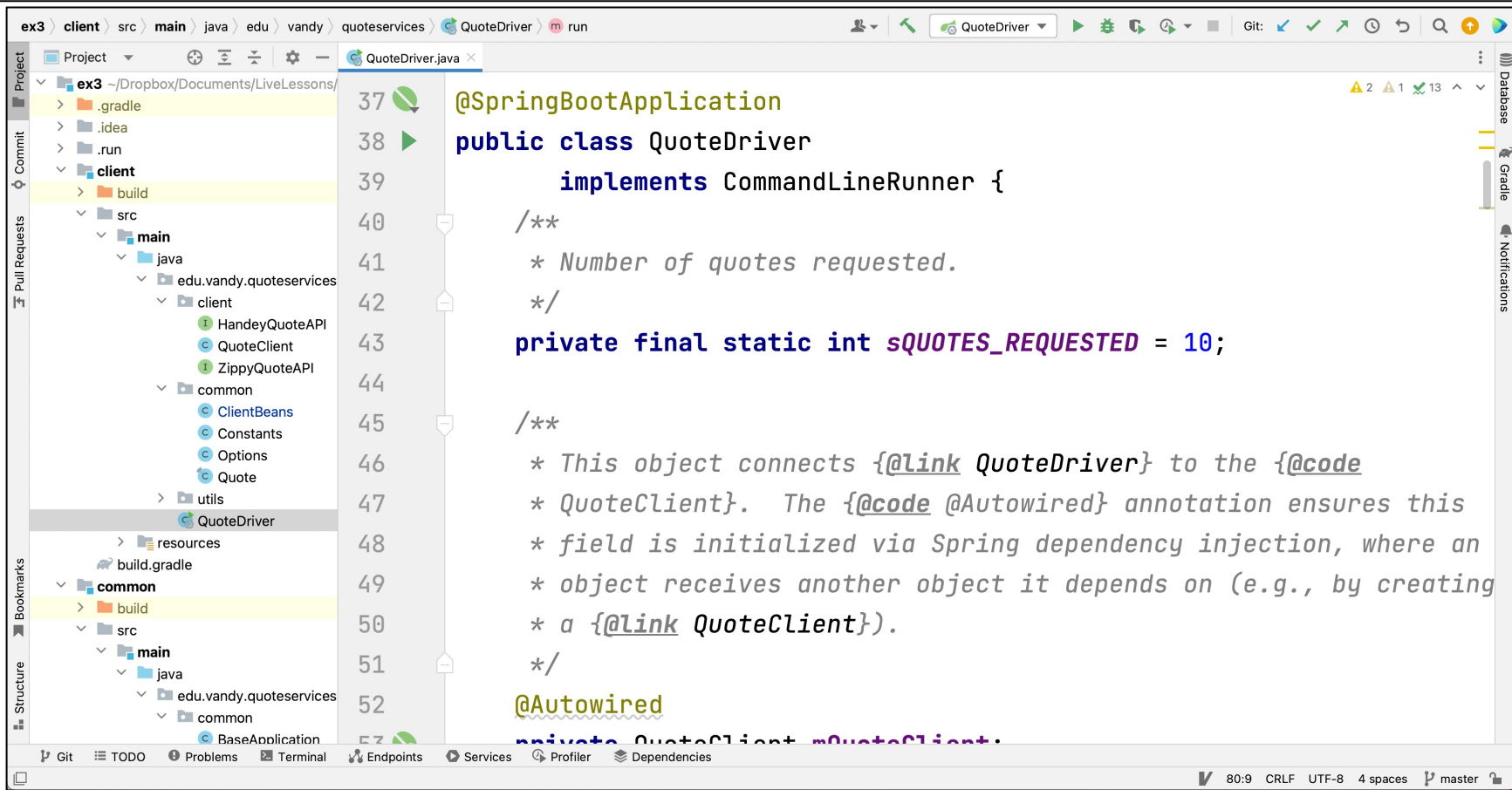
Learning Objectives in this Part of the Lesson

- Understand the implementation of the QuoteDriver class that sends/receives HTTP GET/POST requests/responses to/from the QuoteServices microservices



Implementing the Quote Services App QuoteDriver

Implementing the QuoteServices App QuoteDriver



The screenshot shows an IDE window with the following structure:

- Project: ex3
- src/main/java/edu/vandy/quoteservices/QuoteDriver.java

```
37 @SpringBootApplication
38 public class QuoteDriver
39     implements CommandLineRunner {
40     /**
41      * Number of quotes requested.
42      */
43     private final static int sQUOTES_REQUESTED = 10;
44
45     /**
46      * This object connects {@link QuoteDriver} to the {@code
47      * QuoteClient}. The {@code @Autowired} annotation ensures this
48      * field is initialized via Spring dependency injection, where an
49      * object receives another object it depends on (e.g., by creating
50      * a {@link QuoteClient}).
51      */
52     @Autowired
53     private QuoteClient mQuoteClient;
```

The IDE interface includes a Project view on the left, a central code editor, and a right-hand sidebar with Database, Gradle, and Notifications tabs. The status bar at the bottom shows 80:9 CRLF UTF-8 4 spaces master.

See [WebFlux/ex3/client/src/main/java/edu/vandy/quoteservices/QuoteDriver.java](https://github.com/webfluxio/ex3/client/src/main/java/edu/vandy/quoteservices/QuoteDriver.java)

End of the QuoteServices App Case Study: Implementing the Client- related Classes