

The QuoteServices App Case Study: Implementing the Client-related Classes

Douglas C. Schmidt

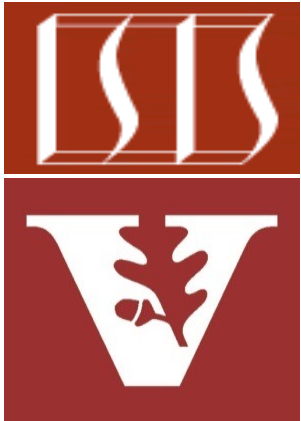
d.schmidt@vanderbilt.edu

www.dre.vanderbilt.edu/~schmidt

Professor of Computer Science

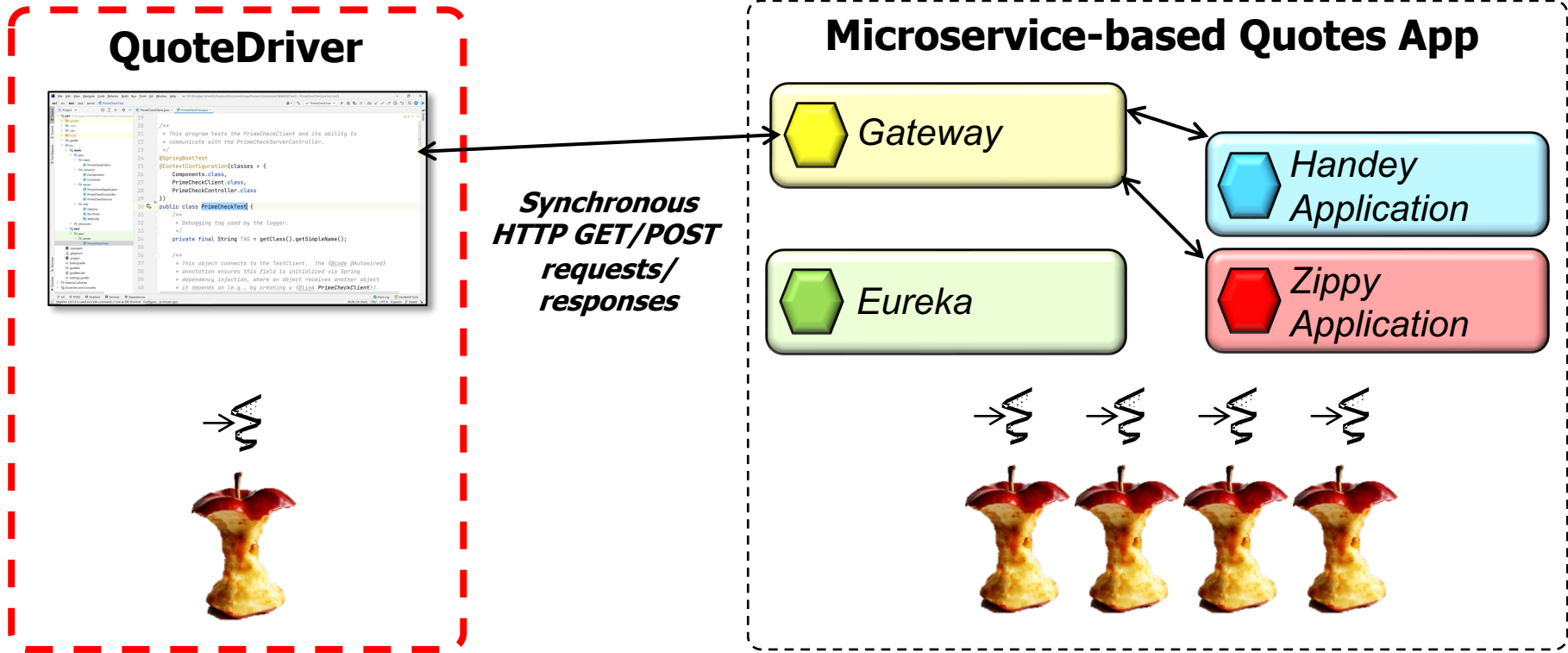
**Institute for Software
Integrated Systems**

**Vanderbilt University
Nashville, Tennessee, USA**



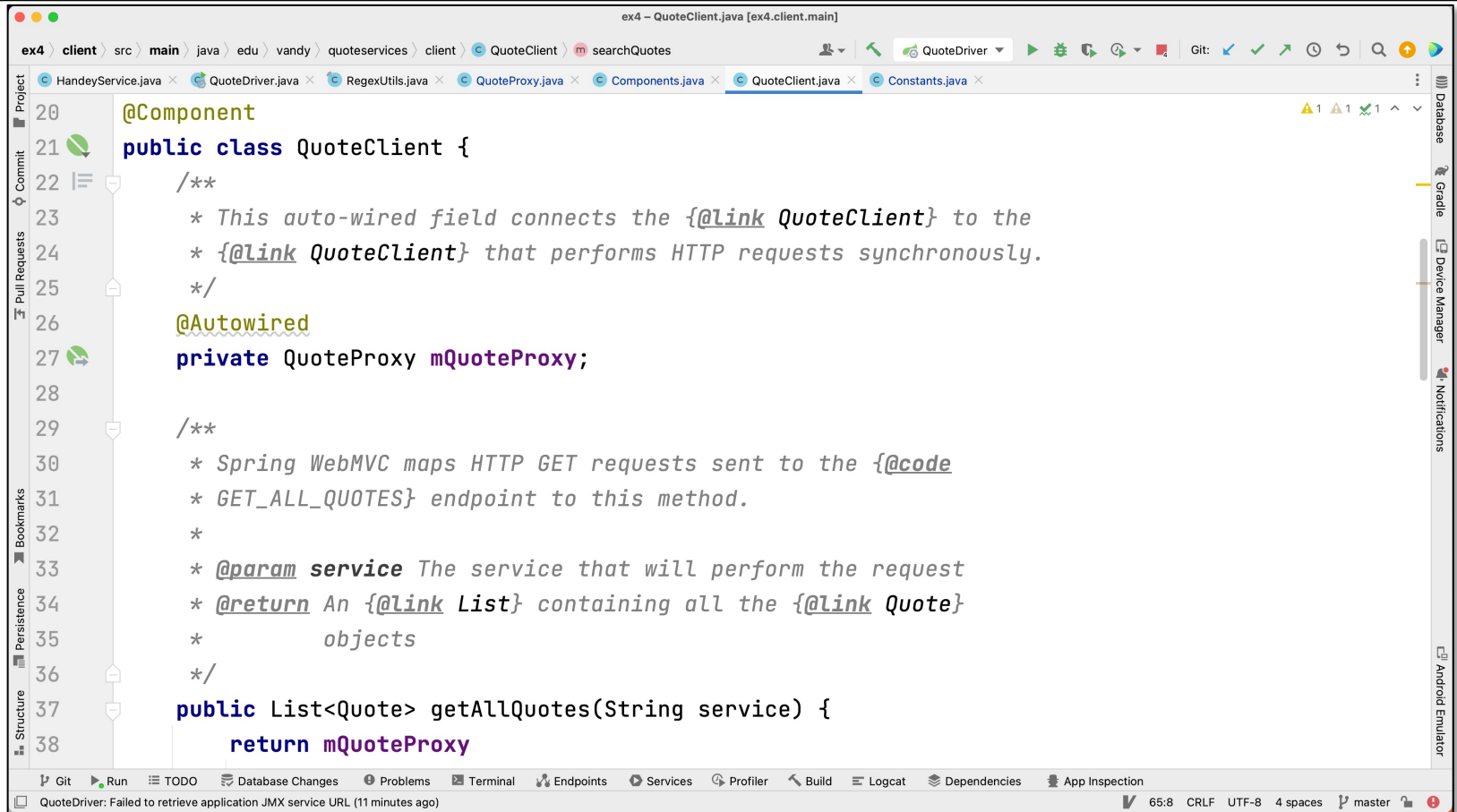
Learning Objectives in this Part of the Lesson

- Understand the implementation of the client-related classes that send/receive HTTP GET/POST requests/responses to/from the QuoteServices microservices



Implementing the QuoteServices App Client

Implementing the QuoteServices App Client



The screenshot shows an IDE window titled "ex4 - QuoteClient.java [ex4.client.main]". The code is as follows:

```
20 @Component
21 public class QuoteClient {
22     /**
23      * This auto-wired field connects the {@link QuoteClient} to the
24      * {@link QuoteClient} that performs HTTP requests synchronously.
25      */
26     @Autowired
27     private QuoteProxy mQuoteProxy;
28
29     /**
30      * Spring WebMVC maps HTTP GET requests sent to the {@code
31      * GET_ALL_QUOTES} endpoint to this method.
32      *
33      * @param service The service that will perform the request
34      * @return An {@link List} containing all the {@link Quote}
35      *         objects
36      */
37     public List<Quote> getAllQuotes(String service) {
38         return mQuoteProxy
```

The IDE interface includes a project structure on the left, a toolbar at the top with icons for search, run, and other actions, and a status bar at the bottom showing "QuoteDriver: Failed to retrieve application JMX service URL (11 minutes ago)".

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

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