

The QuoteServices App Case Study: Test Driver Implementation & Performance

Douglas C. Schmidt

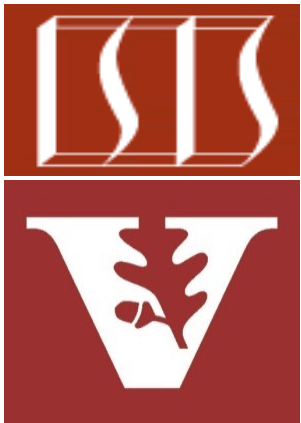
d.schmidt@vanderbilt.edu

www.dre.vanderbilt.edu/~schmidt

Professor of Computer Science

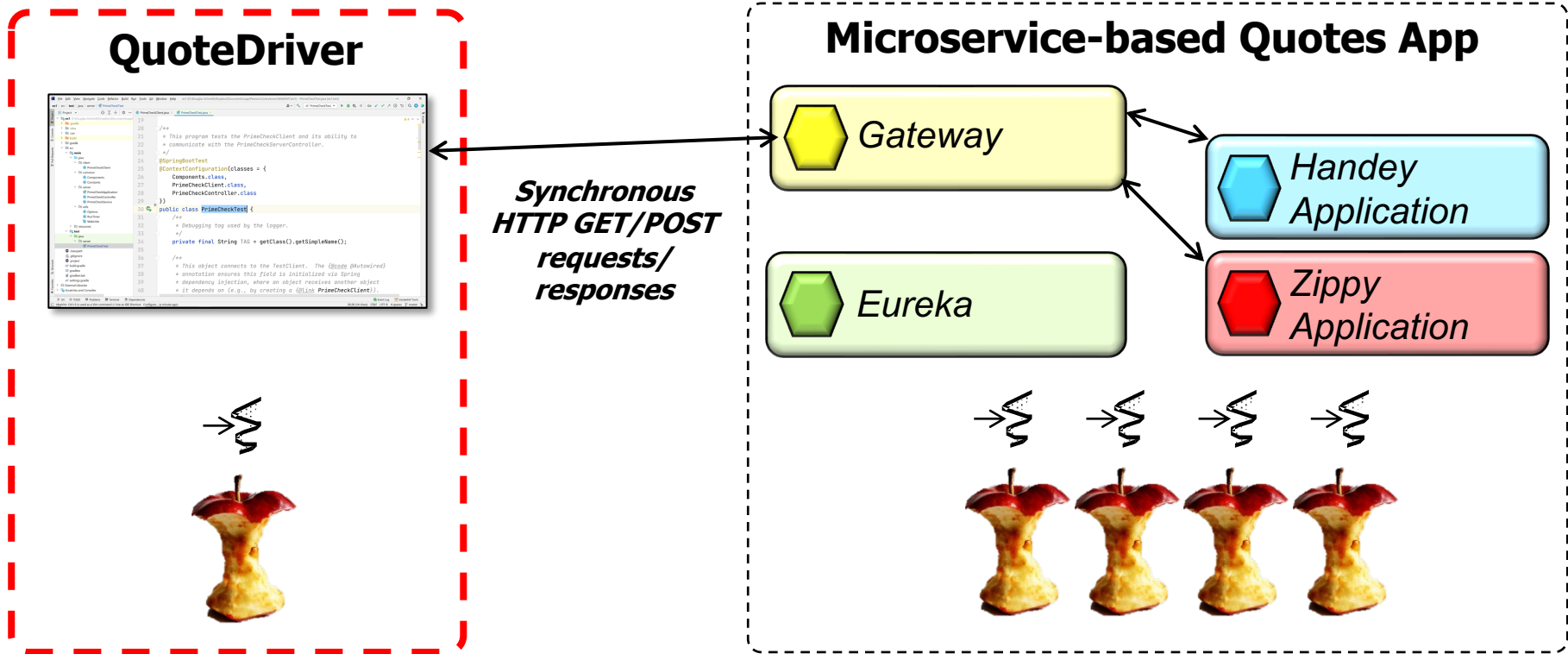
**Institute for Software
Integrated Systems**

**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

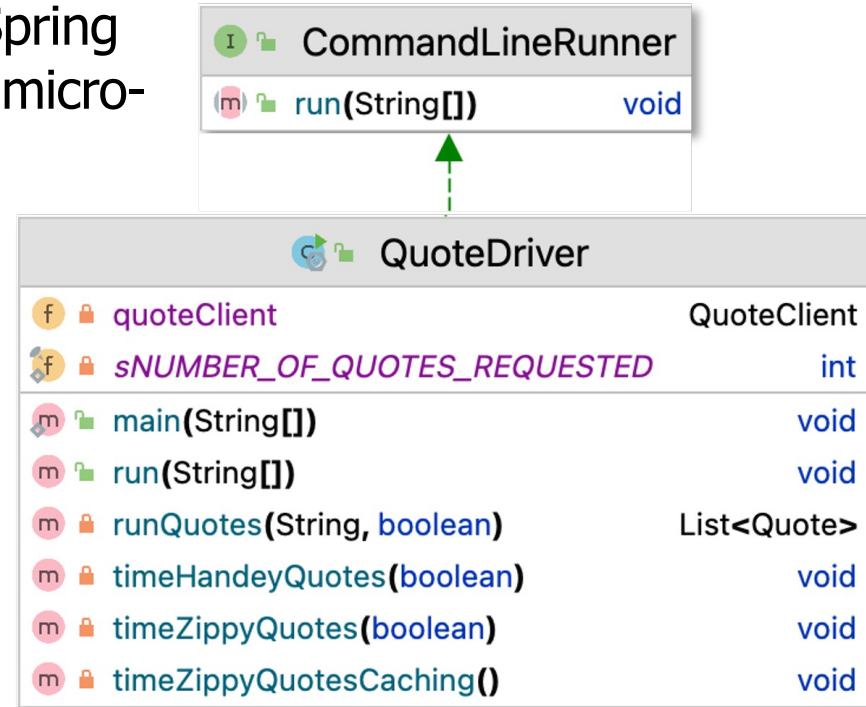


The performance of both sequential & parallel computations is analyzed

Overview of the QuoteDriver Class

Overview of the QuoteServicesTest Driver

- QuoteDriver shows how a client can use Spring WebMVC features to interact with various micro-services via an API Gateway



Overview of the QuoteServicesTest Driver

- QuoteDriver shows how a client can use Spring WebMVC features to interact with various micro-services via an API Gateway

- run(String... args)**

- Orchestrates various operations with & without server-side caching
- e.g., timing the fetching of quotes from Zippy & Handey controllers

```
CommandLineRunner
run(String[]) void
```

```
QuoteDriver
quoteClient QuoteClient
sNUMBER_OF_QUOTES_REQUESTED int
main(String[]) void
run(String[]) void
runQuotes(String, boolean) List<Quote>
timeHandeyQuotes(boolean) void
timeZippyQuotes(boolean) void
timeZippyQuotesCaching() void
```

Overview of the QuoteServicesTest Driver

- QuoteDriver shows how a client can use Spring WebMVC features to interact with various micro-services via an API Gateway

- run(String... args)**

- Orchestrates various operations with & without server-side caching
- Overrides the run() hook method in CommandLineRunner

```
CommandLineRunner  
run(String[]) void
```

```
QuoteDriver  
quoteClient QuoteClient  
sNUMBER_OF_QUOTES_REQUESTED int  
main(String[]) void  
run(String[]) void  
runQuotes(String, boolean) List<Quote>  
timeHandeyQuotes(boolean) void  
timeZippyQuotes(boolean) void  
timeZippyQuotesCaching() void
```

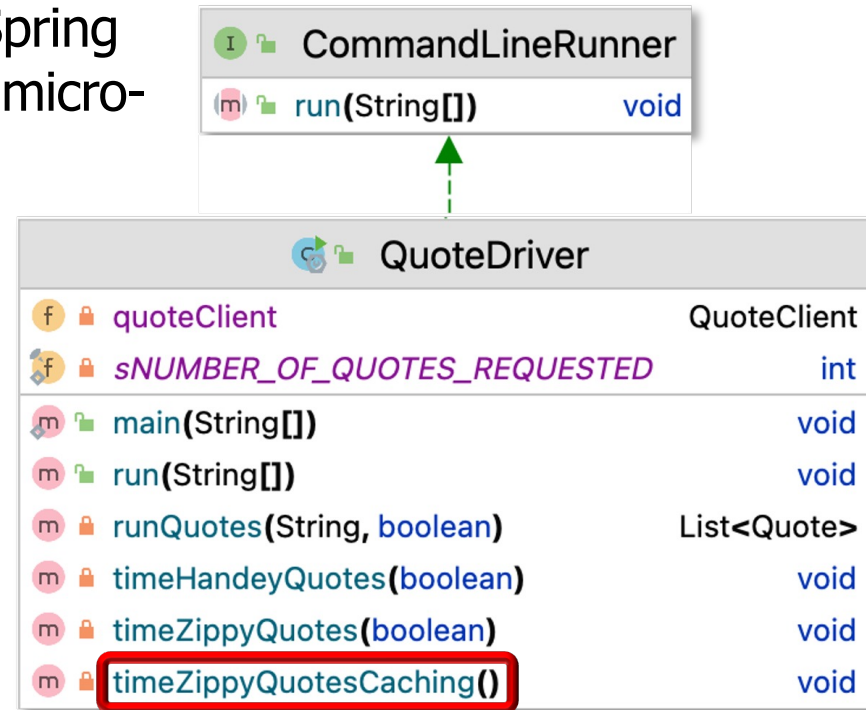
See [springframework/boot/CommandLineRunner.html](https://springframework.org/boot/CommandLineRunner.html)

Overview of the QuoteServicesTest Driver

- QuoteDriver shows how a client can use Spring WebMVC features to interact with various micro-services via an API Gateway

- timeZippyQuotesCaching()**

- Demonstrates benefits of server-side caching by timing the fetching of all quotes from the Zippy service
- Compares the performance of initial requests & subsequent cached requests

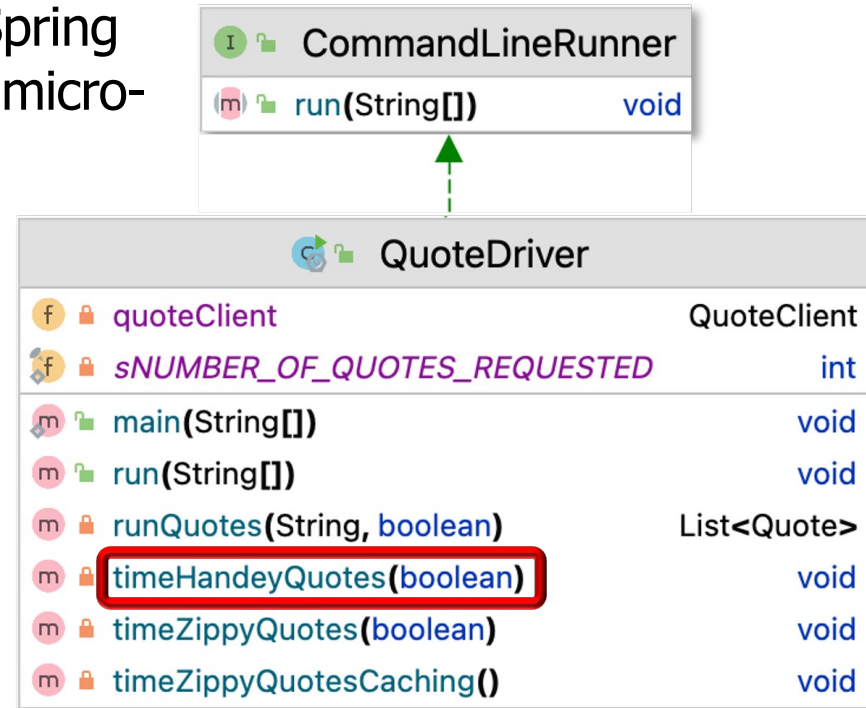


Overview of the QuoteServicesTest Driver

- QuoteDriver shows how a client can use Spring WebMVC features to interact with various micro-services via an API Gateway

- timeHandeyQuotes()**

- Times fetching of quotes from the Handey service
- Compares both sequential & parallel execution

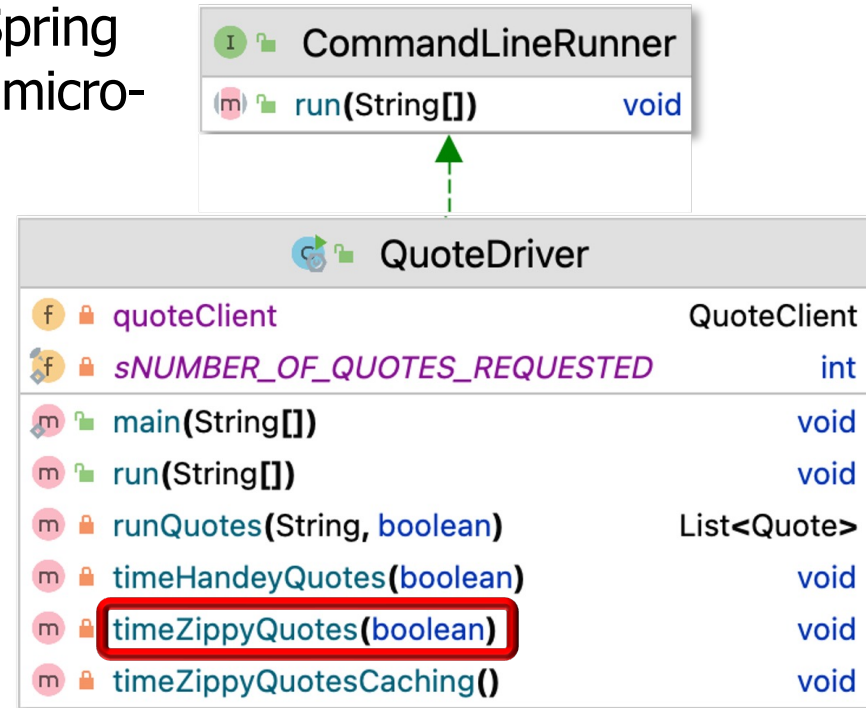


Overview of the QuoteServicesTest Driver

- QuoteDriver shows how a client can use Spring WebMVC features to interact with various micro-services via an API Gateway

- timeZippyQuotes()**

- Times fetching of quotes from the Zippy service
- Compares both sequential & parallel execution

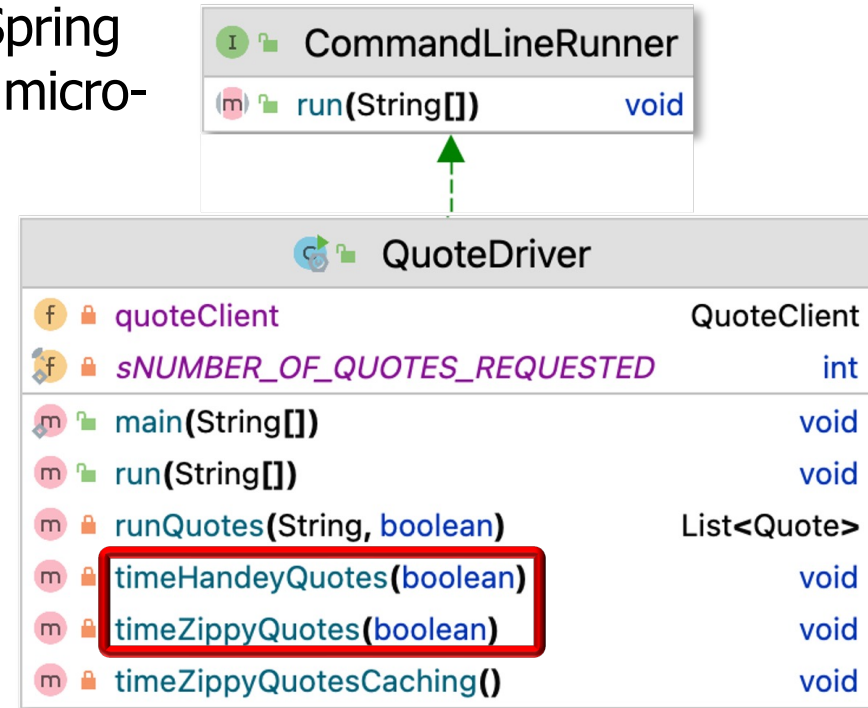


Overview of the QuoteServicesTest Driver

- QuoteDriver shows how a client can use Spring WebMVC features to interact with various micro-services via an API Gateway

- runQuotes()**

- A helper method that encapsulates common logic for fetching quotes from a specified service
- The methods `timeZippyQuotes` & `timeHandeyQuotes` use this helper method internally



Overview of the QuoteServicesTest Driver

- QuoteDriver shows how a client can use Spring WebMVC features to interact with various micro-services via an API Gateway

- runQuotes()**

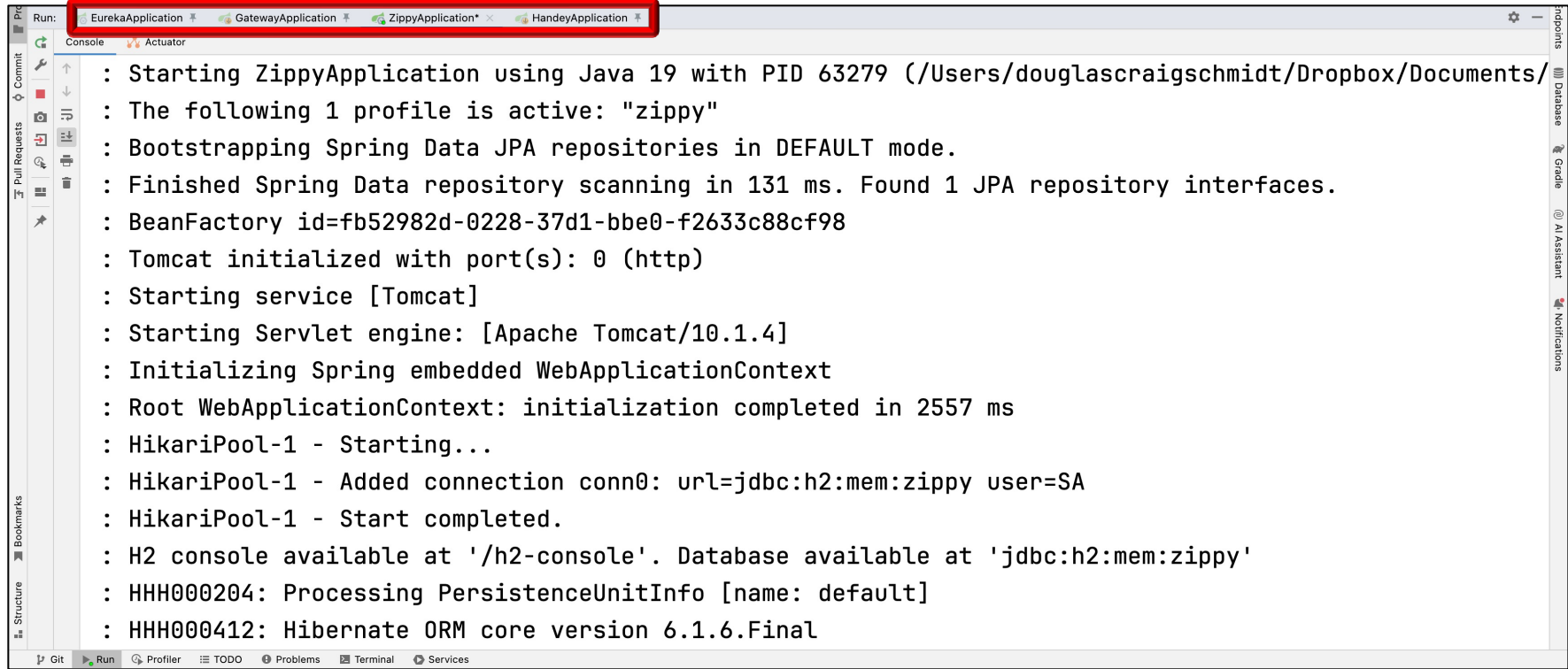
- A helper method that encapsulates common logic for fetching quotes from a specified service
- The methods `timeZippyQuotes` & `timeHandeyQuotes` use this helper method internally
- Supports both sequential & parallel execution

```
CommandLineRunner
run(String[]) void
```

```
QuoteDriver
quoteClient QuoteClient
sNUMBER_OF_QUOTES_REQUESTED int
main(String[]) void
run(String[]) void
runQuotes(String, boolean) List<Quote>
timeHandeyQuotes(boolean) void
timeZippyQuotes(boolean) void
timeZippyQuotesCaching() void
```

Overview of the QuoteServicesTest Driver

- The EurekaApplication, GatewayApplication, HandeyApplication, & Zippy Application must be started before the ClientDriver program can be run



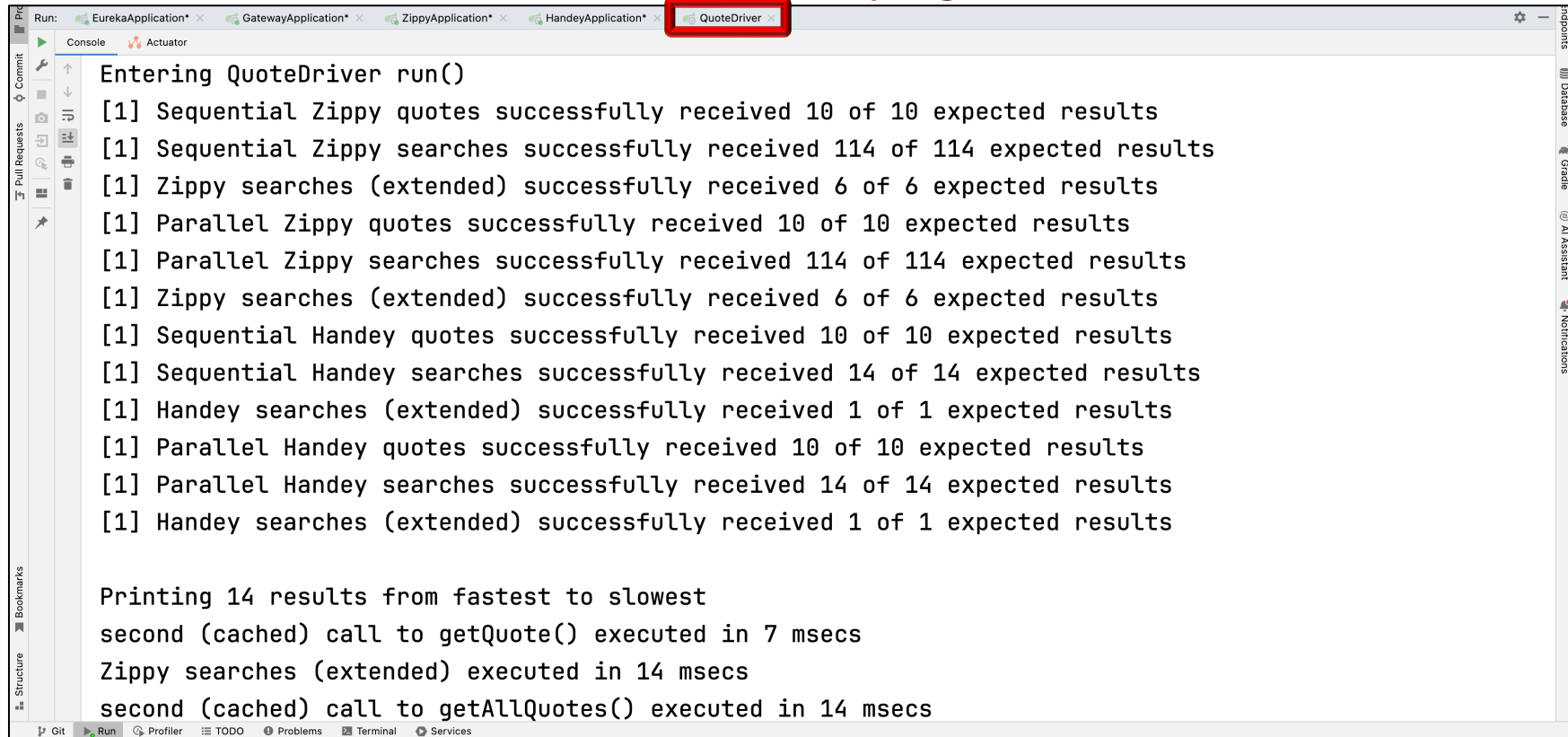
The screenshot shows the IntelliJ IDEA Run console with four tabs: EurekaApplication, GatewayApplication, ZippyApplication, and HandeyApplication. The ZippyApplication tab is active and displays the following startup logs:

```
: Starting ZippyApplication using Java 19 with PID 63279 (/Users/douglasraigschmidt/Dropbox/Documents/
: The following 1 profile is active: "zippy"
: Bootstrapping Spring Data JPA repositories in DEFAULT mode.
: Finished Spring Data repository scanning in 131 ms. Found 1 JPA repository interfaces.
: BeanFactory id=fb52982d-0228-37d1-bbe0-f2633c88cf98
: Tomcat initialized with port(s): 0 (http)
: Starting service [Tomcat]
: Starting Servlet engine: [Apache Tomcat/10.1.4]
: Initializing Spring embedded WebApplicationContext
: Root WebApplicationContext: initialization completed in 2557 ms
: HikariPool-1 - Starting...
: HikariPool-1 - Added connection conn0: url=jdbc:h2:mem:zippy user=SA
: HikariPool-1 - Start completed.
: H2 console available at '/h2-console'. Database available at 'jdbc:h2:mem:zippy'
: HHH000204: Processing PersistenceUnitInfo [name: default]
: HHH000412: Hibernate ORM core version 6.1.6.Final
```

We'll launch them manually via IntelliJ

Overview of the QuoteServicesTest Driver

- The ClientDriver runs as a “command-line” program



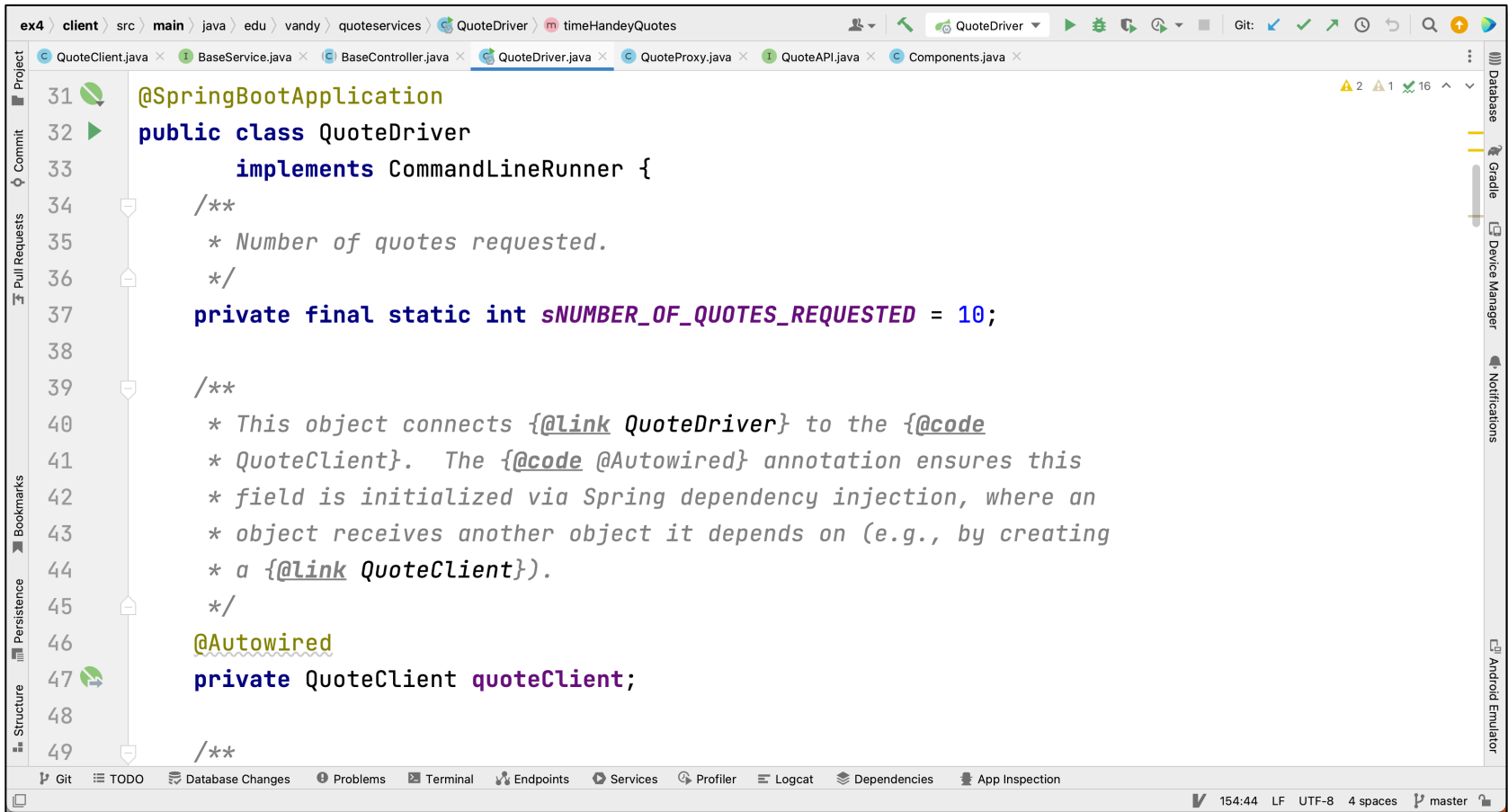
```
Run: EurekaApplication* x GatewayApplication* x ZippyApplication* x HandeyApplication* x QuoteDriver
Console Actuator
Entering QuoteDriver run()
[1] Sequential Zippy quotes successfully received 10 of 10 expected results
[1] Sequential Zippy searches successfully received 114 of 114 expected results
[1] Zippy searches (extended) successfully received 6 of 6 expected results
[1] Parallel Zippy quotes successfully received 10 of 10 expected results
[1] Parallel Zippy searches successfully received 114 of 114 expected results
[1] Zippy searches (extended) successfully received 6 of 6 expected results
[1] Sequential Handey quotes successfully received 10 of 10 expected results
[1] Sequential Handey searches successfully received 14 of 14 expected results
[1] Handey searches (extended) successfully received 1 of 1 expected results
[1] Parallel Handey quotes successfully received 10 of 10 expected results
[1] Parallel Handey searches successfully received 14 of 14 expected results
[1] Handey searches (extended) successfully received 1 of 1 expected results

Printing 14 results from fastest to slowest
second (cached) call to getQuote() executed in 7 msecs
Zippy searches (extended) executed in 14 msecs
second (cached) call to getAllQuotes() executed in 14 msecs
```

The client prints results received from the API gateway in the console window

Implementing the QuoteDriver Class

Implementing the QuoteServicesTest Driver



The screenshot shows an IDE window with the following details:

- Path:** ex4 > client > src > main > java > edu > vandy > quoteservices > QuoteDriver > timeHandeyQuotes
- Open Files:** QuoteClient.java, BaseService.java, BaseController.java, QuoteDriver.java (active), QuoteProxy.java, QuoteAPI.java, Components.java
- Code:**

```
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     /**
```
- IDE Features:** Left sidebar (Project, Commit, Pull Requests, Bookmarks, Persistence, Structure), right sidebar (Database, Gradle, Device Manager, Notifications, Android Emulator), bottom toolbar (Git, TODO, Database Changes, Problems, Terminal, Endpoints, Services, Profiler, Logcat, Dependencies, App Inspection), status bar (154.44, LF, UTF-8, 4 spaces, master).

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

End of the QuoteServices App Case Study: Test Driver