

# Implementing the RSocket Shakespeare Quotes Case Study App Responder (Part 3)

**Douglas C. Schmidt**

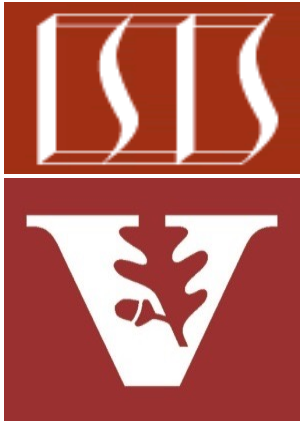
**[d.schmidt@vanderbilt.edu](mailto:d.schmidt@vanderbilt.edu)**

**[www.dre.vanderbilt.edu/~schmidt](http://www.dre.vanderbilt.edu/~schmidt)**

**Professor of Computer Science**

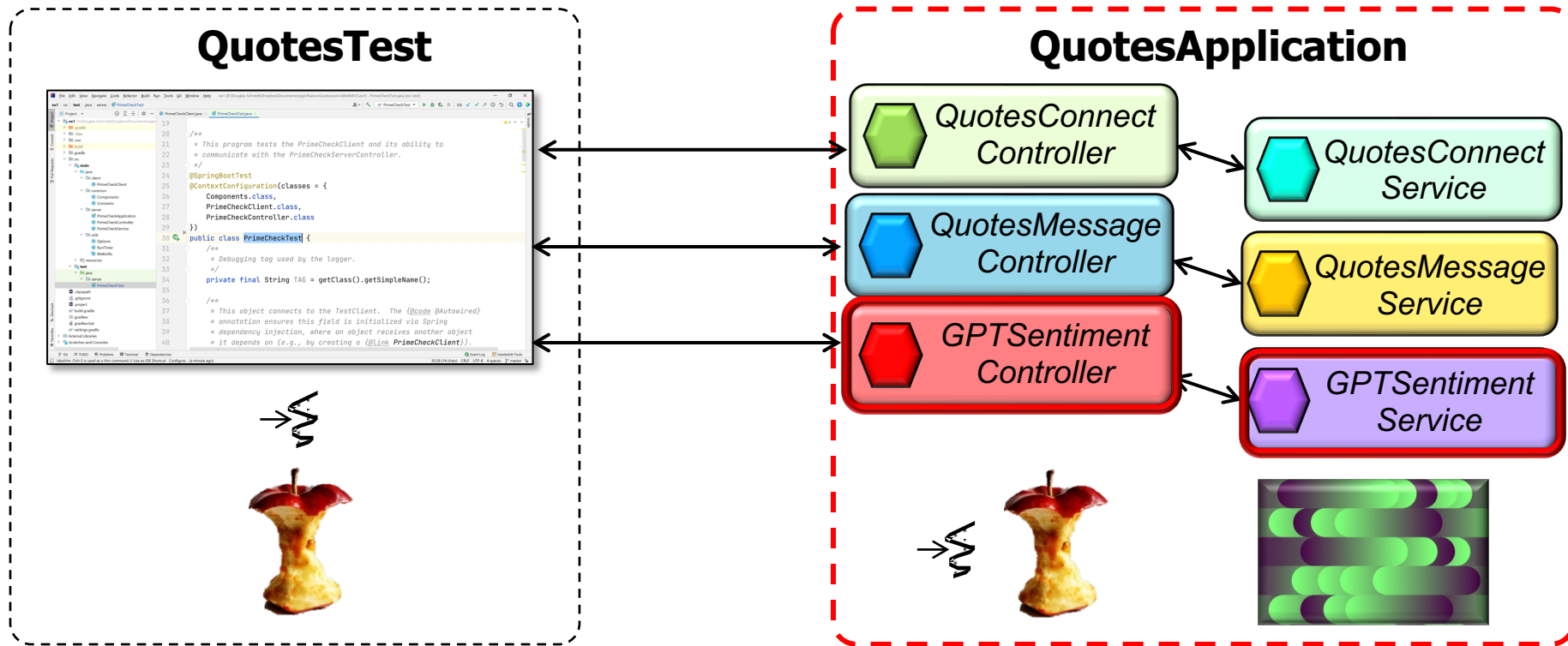
**Institute for Software  
Integrated Systems**

**Vanderbilt University  
Nashville, Tennessee, USA**



# Learning Objectives in this Lesson

- Understand the implementation of the sentiment analysis portion of the RSocket Shakespeare Quotes app responder



Uses the OpenAiService to interact with ChatGPT

---

# Implementing the RSocket Shakespeare Quotes App Responder

# Implementing the RSocket Shakespeare Quotes App Responder

The screenshot shows an IDE window with the following components:

- Project Explorer (Left):** Shows a project structure with folders like `model`, `repository`, `responder`, `connect`, `quoter`, `utils`, `resources`, and `test`. The `GPTSentimentController` file is selected under `responder`.
- Code Editor (Center):** Displays the source code for `GPTSentimentController.java`. The code includes:
  - Line 31: `@Controller` annotation.
  - Line 32: `public class GPTSentimentController {`
  - Line 33: `/**`
  - Lines 34-39: Javadoc comment describing the controller's role and dependencies.
  - Line 40: `@Autowired` annotation.
  - Line 41: `private GPTSentimentService mService;`
  - Line 42: `/**`
  - Lines 43-48: Javadoc comment for the `analyzeSentiment` method.
- Run and Debug Console (Bottom):** Shows the status of tests: "Tests failed: 1, passed: 1 (yesterday 9:08 PM)".
- Right Side Panel:** Contains toolbars for Endpoints, Database, Gradle, Device Manager, and Notifications.

See [github.com/douglasraigschmidt/LiveLessons/tree/master/RSocket/ex3](https://github.com/douglasraigschmidt/LiveLessons/tree/master/RSocket/ex3)

---

# End of Implementing the RSocket Shakespeare Quotes Case Study App Responder (Part 3)