

The QuoteServices App Case Study: Overview

Douglas C. Schmidt

d.schmidt@vanderbilt.edu

www.dre.vanderbilt.edu/~schmidt

Professor of Computer Science

**Institute for Software
Integrated Systems**

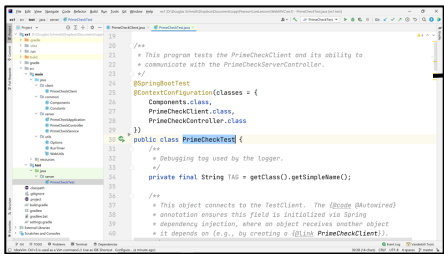
**Vanderbilt University
Nashville, Tennessee, USA**



Learning Objectives in this Part of the Lesson

- Understand how various concurrency & persistency frameworks are applied in a case study using Spring WebMVC to provide two different quote services

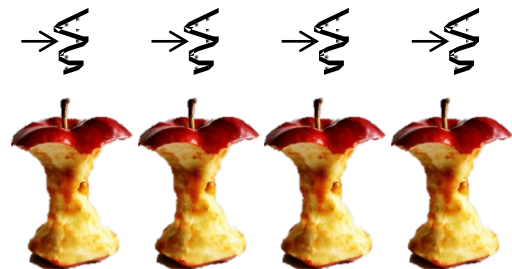
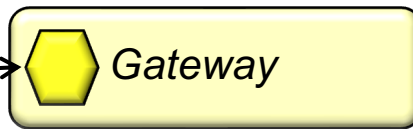
QuoteDriver



***Synchronous
HTTP GET/POST
requests/
responses***



Microservice-based Quotes App



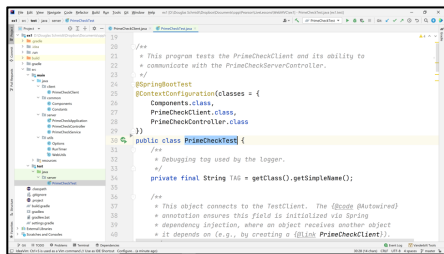
See github.com/douglasraigschmidt/LiveLessons/tree/master/WebMVC/ex4

Overview of the Quote Services App Case Study

Overview of the QuoteServices App Case Study

- This case study shows how Spring WebMVC can send/receive HTTP GET/POST requests synchronously to/from an API gateway & multiple microservices

QuoteDriver



*The client sends requests
to the API gateway (&
only the API gateway)*

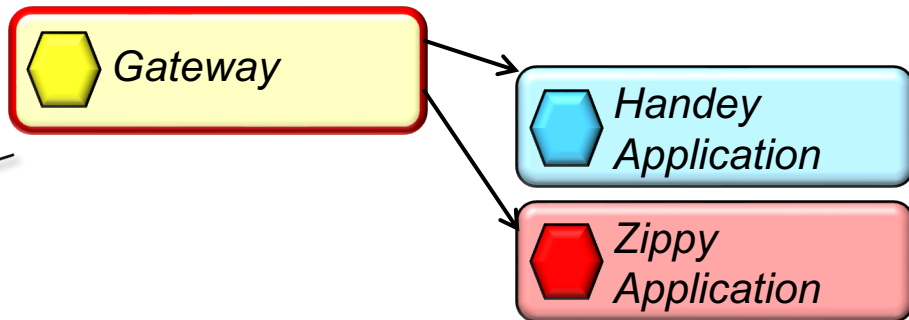


See github.com/douglasraigschmidt/LiveLessons/tree/master/WebMVC/ex4/client

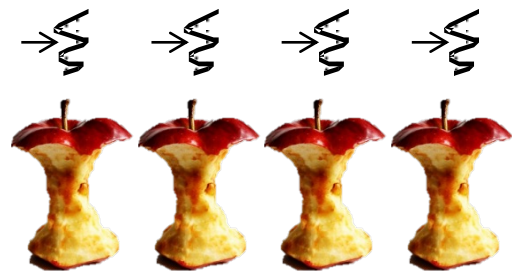
Overview of the QuoteServices App Case Study

- This case study shows how Spring WebMVC can send/receive HTTP GET/POST requests synchronously to/from an API gateway & multiple microservices

Microservice-based Quotes App



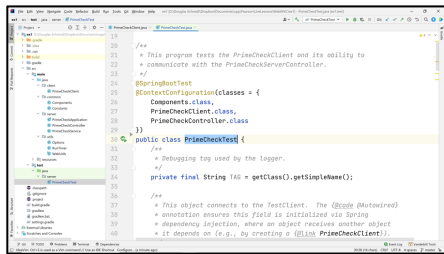
The API gateway receives client HTTP requests & routes them to the designated microservice



Overview of the QuoteServices App Case Study

- This case study shows how Spring WebMVC can send/receive HTTP GET/POST requests synchronously to/from an API gateway & multiple microservices

QuoteDriver



Microservice-based Quotes App



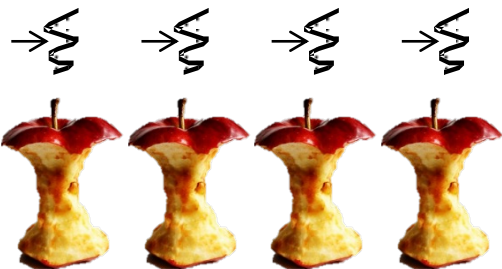
Gateway



Handey
Application



Zippy
Application



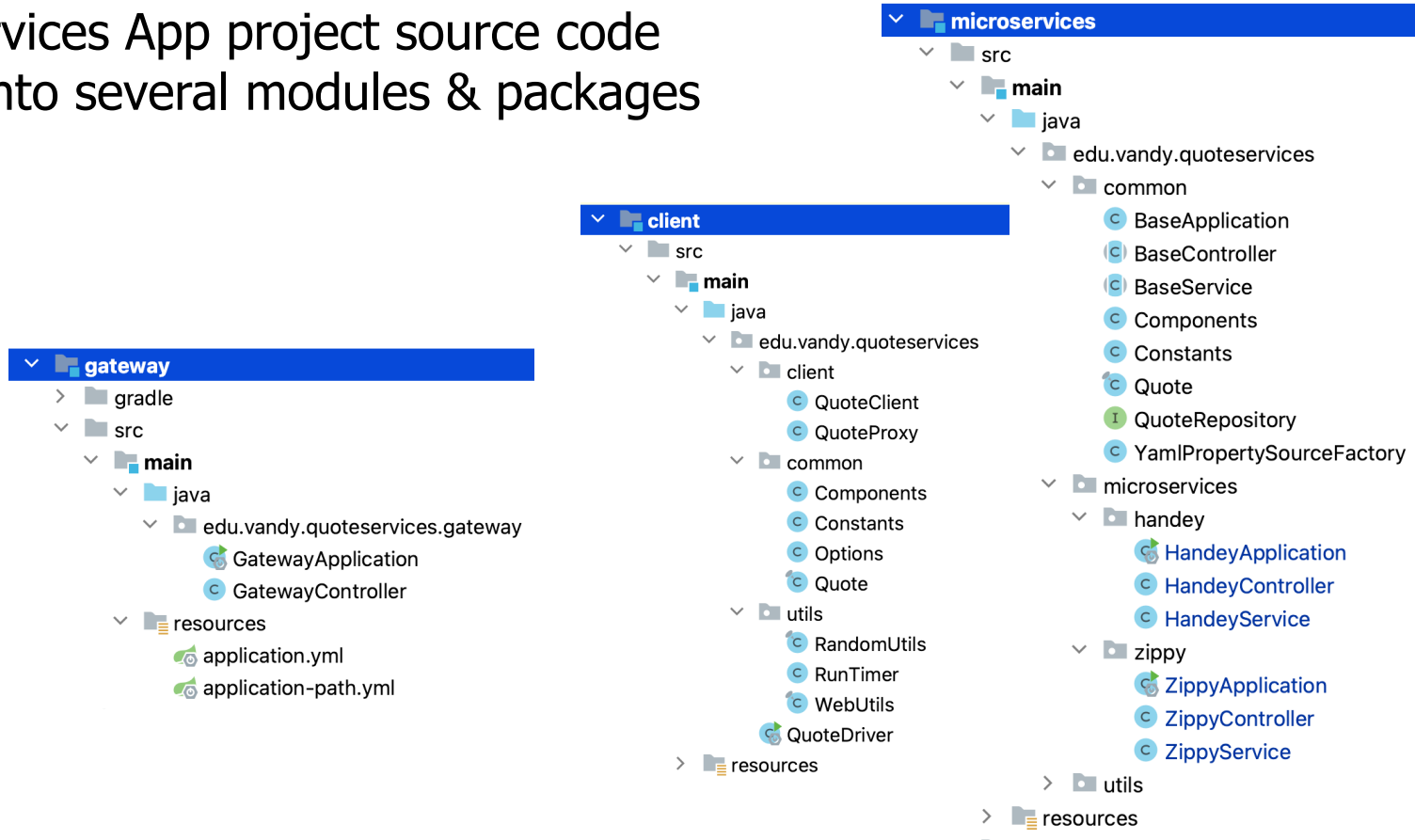
The microservices respond with quotes when the API gateway forwards them HTTP requests

See [WebMVC/ex4/microservices](https://www.baeldung.com/spring-webmvc-ex4-microservices)

Structure of the Quote Services App Project

Structure of the QuoteServices App Project

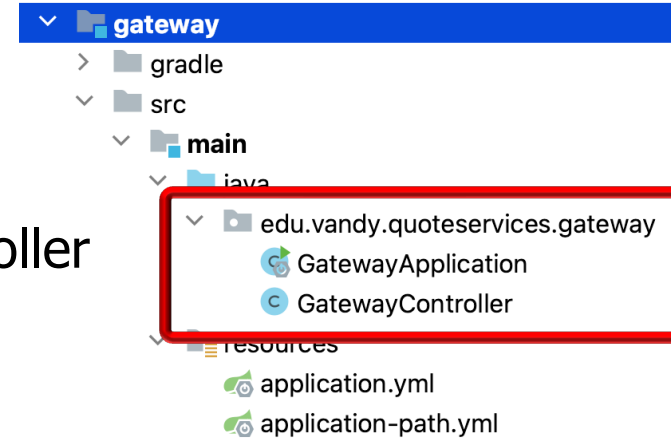
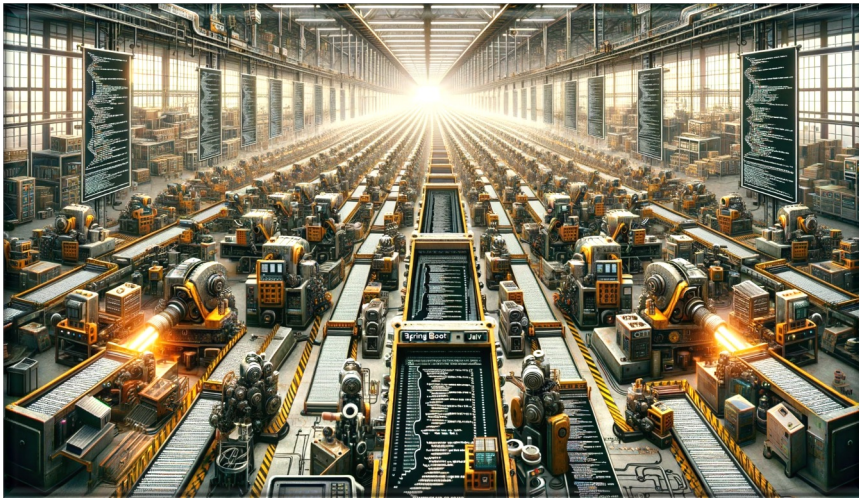
- The QuoteServices App project source code is organized into several modules & packages



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

Structure of the QuoteServices App Project

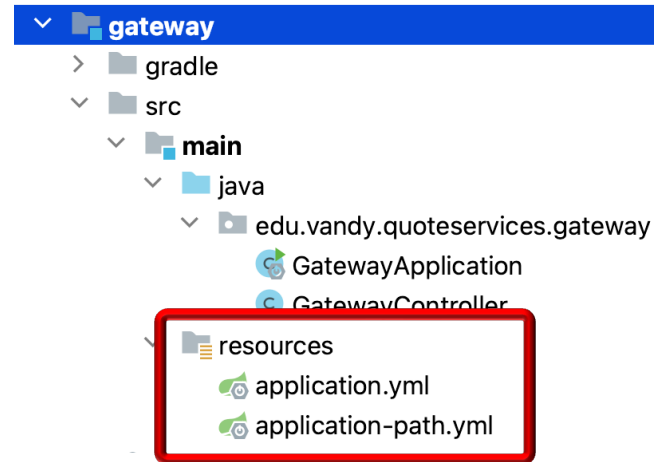
- The QuoteServices App project source code is organized into several modules & packages
 - main
 - gateway
 - Contains the “app” entry points & the controller



This code is largely “boilerplate”

Structure of the QuoteServices App Project

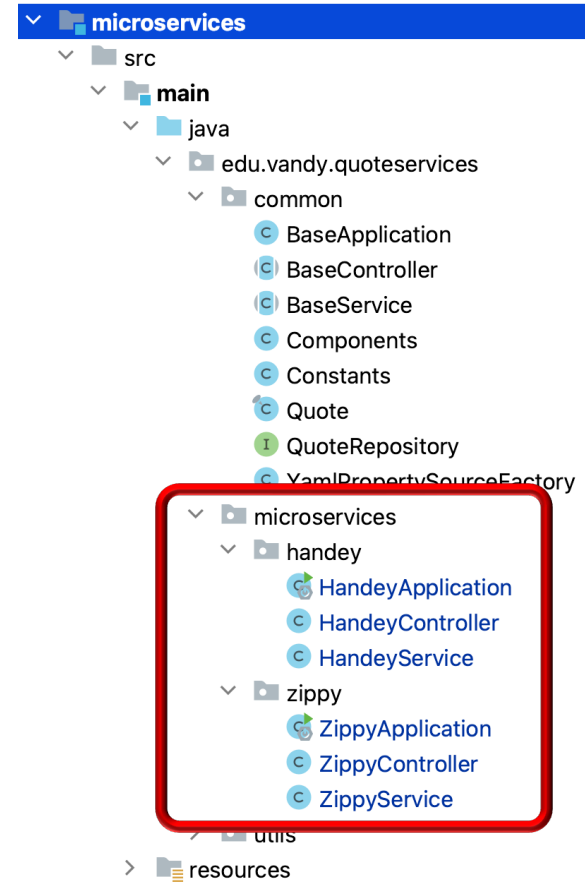
- The QuoteServices App project source code is organized into several modules & packages
 - main
 - gateway
 - resources
 - Specifies the port numbers & microservices exposed by the API gateway



See en.wikipedia.org/wiki/YAML

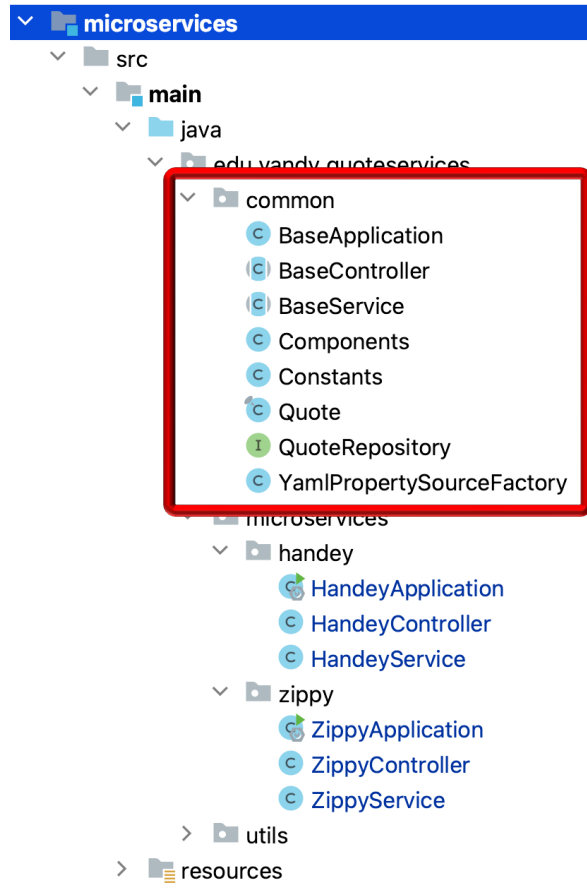
Structure of the QuoteServices App Project

- The QuoteServices App project source code is organized into several modules & packages
 - main
 - microservices
 - Contains the “app” entry points & the controller



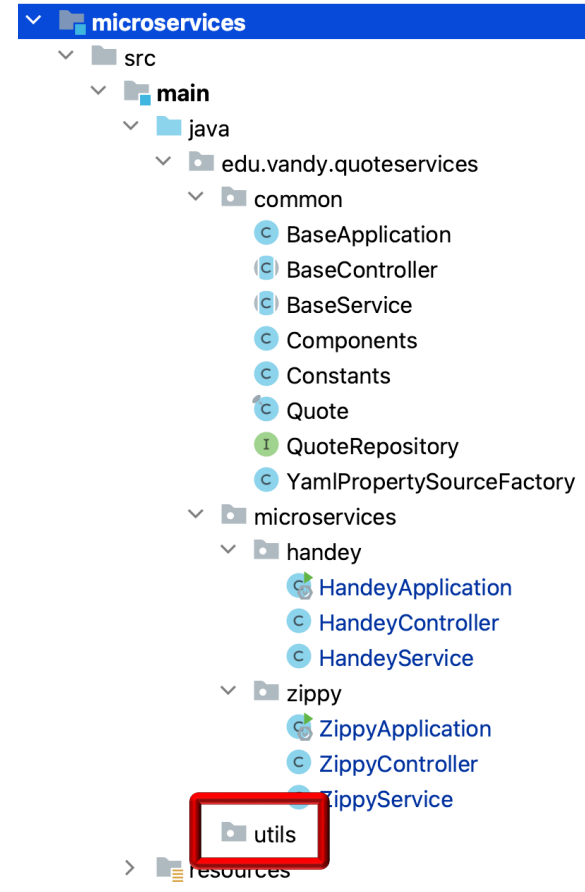
Structure of the QuoteServices App Project

- The QuoteServices App project source code is organized into several modules & packages
 - main
 - `microservices`
 - `common`
 - Consolidates various project-specific helper classes



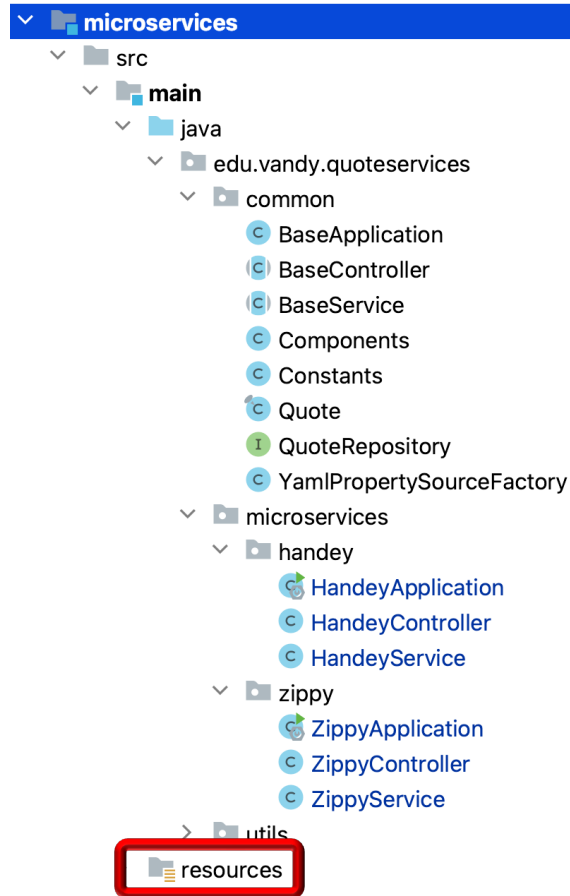
Structure of the QuoteServices App Project

- The QuoteServices App project source code is organized into several modules & packages
 - main
 - microservices
 - common
 - utils
 - Consolidates various reusable helper classes



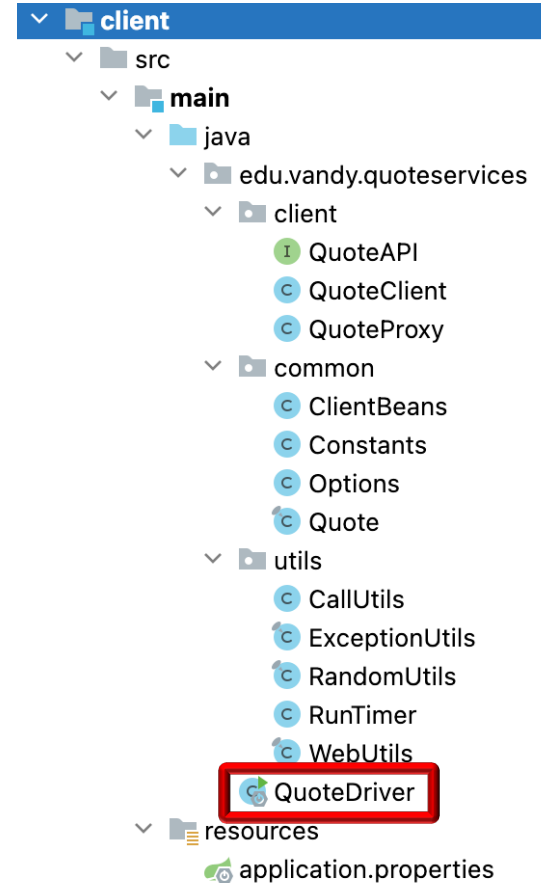
Structure of the QuoteServices App Project

- The QuoteServices App project source code is organized into several modules & packages
 - main
 - microservices
 - common
 - utils
 - resources
 - Defines various application properties
 - e.g., microservice names & port numbers, schema definitions for quotes



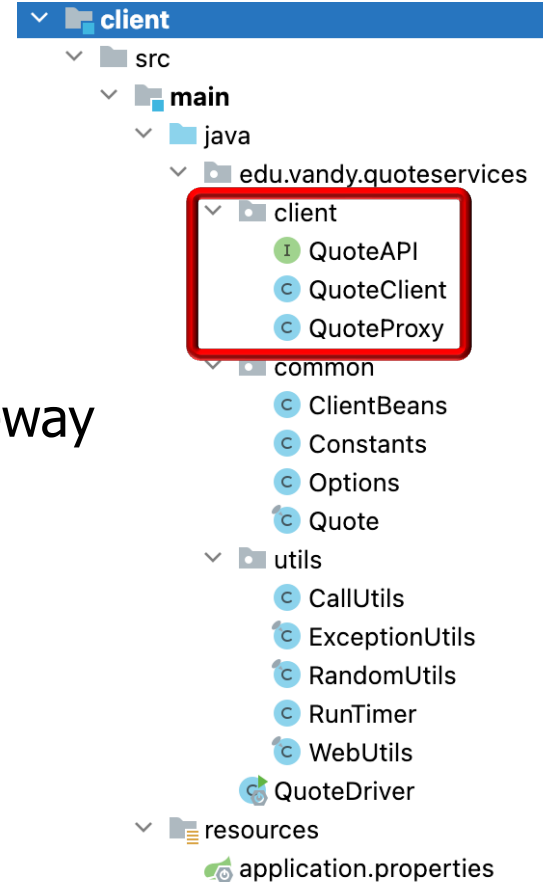
Structure of the QuoteServices App Project

- The QuoteServices App project source code is organized into several modules & packages
 - client
 - QuoteDriver
 - This test driver causes the client to send/receive requests/responses to/from the API gateway running on the server & displays the results
 - The API gateway, in turn, routes the requests to the designated microservice



Structure of the QuoteServices App Project

- The QuoteServices App project source code is organized into several modules & packages
 - client
 - QuoteDriver
 - client
 - Sends HTTP GET/POST requests to the API gateway



Structure of the QuoteServices App Project

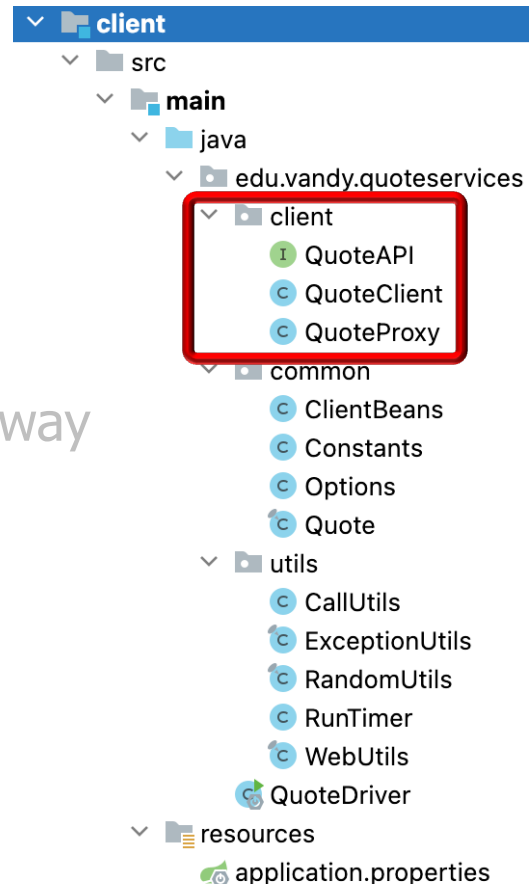
- The QuoteServices App project source code is organized into several modules & packages

- client

- QuoteDriver

- client

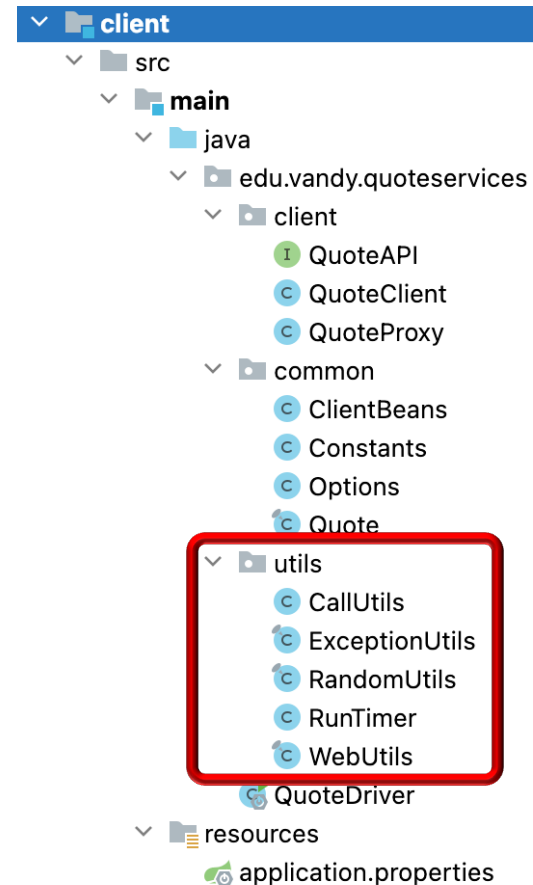
- Sends HTTP GET/POST requests to the API gateway
 - Uses Retrofit to auto-generate proxies



See square.github.io/retrofit

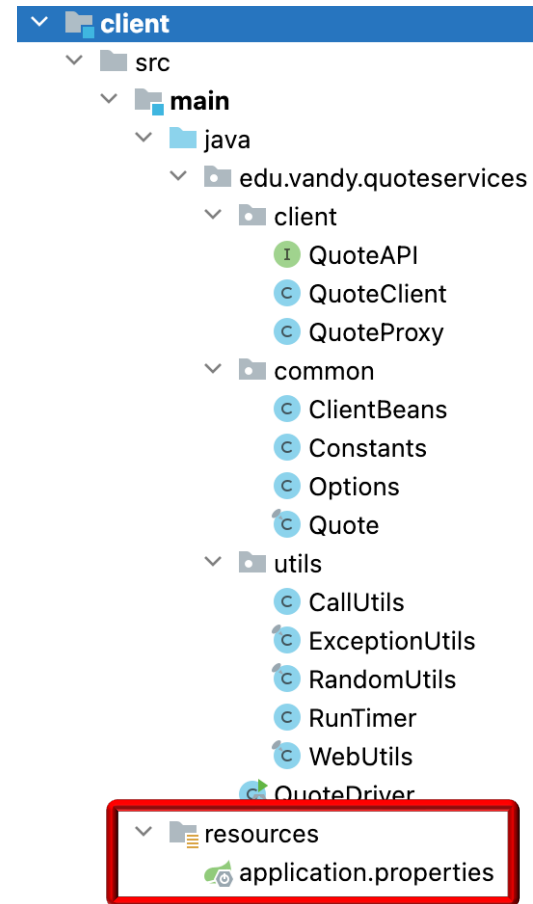
Structure of the QuoteServices App Project

- The QuoteServices App project source code is organized into several modules & packages
 - client
 - QuoteDriver
 - client
 - utils
 - Consolidates various reusable helper classes



Structure of the QuoteServices App Project

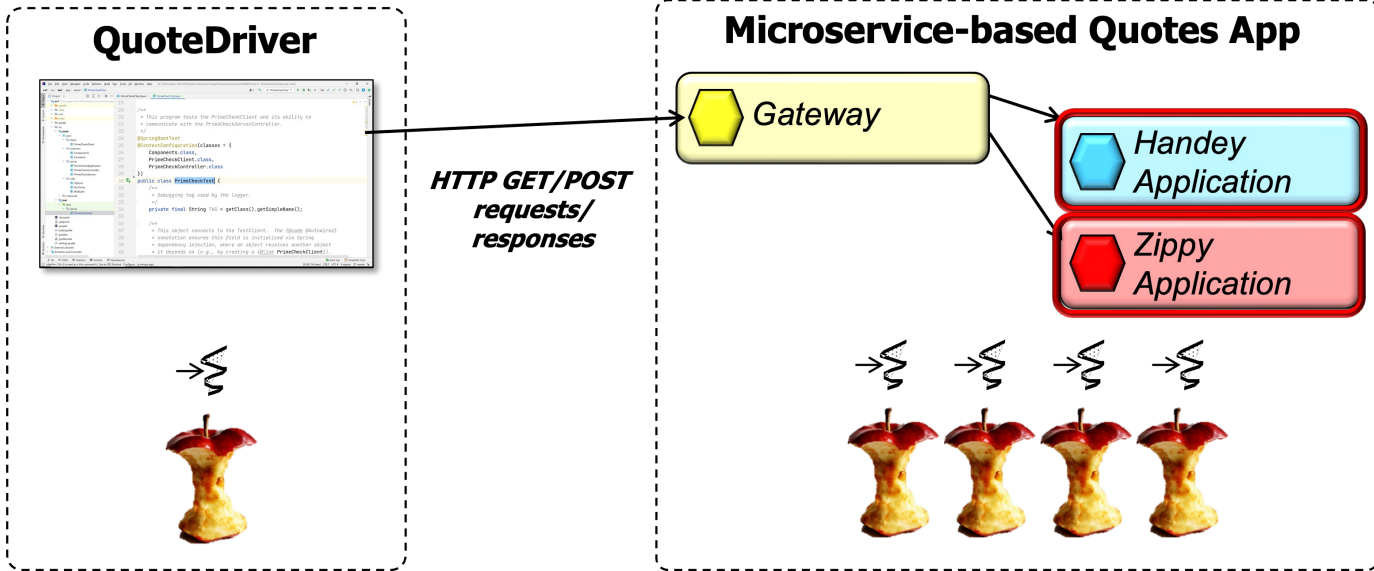
- The QuoteServices App project source code is organized into several modules & packages
 - client
 - QuoteDriver
 - client
 - utils
 - resources
 - Defines various application properties
 - e.g., disable/enable logging & sets the client driver port number



Pros & Cons of the QuoteServices App

Pros & Cons of the QuoteServices App

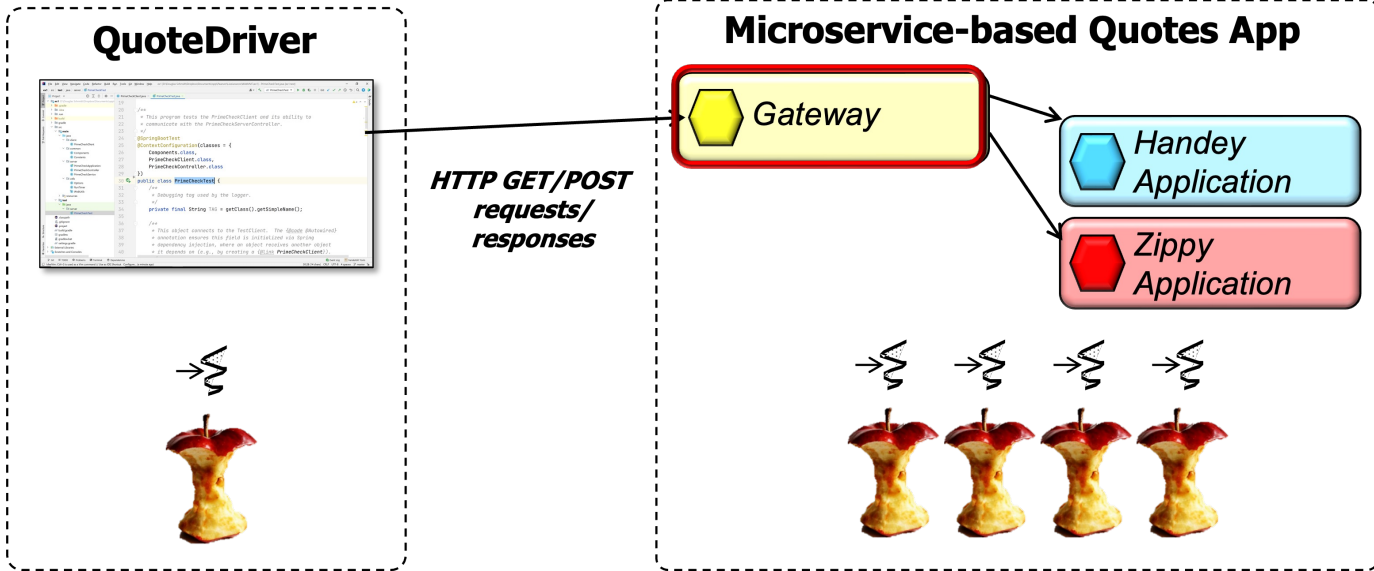
- Pros
 - Each microservice runs in its own process (& potentially its own computer in a data center or cloud environment)



Can improve system scalability & reliability

Pros & Cons of the QuoteServices App

- Pros
 - Each microservice runs in its own process (& potentially its own computer in a data center or cloud environment)
 - Clients only need to communicate with the API Gateway

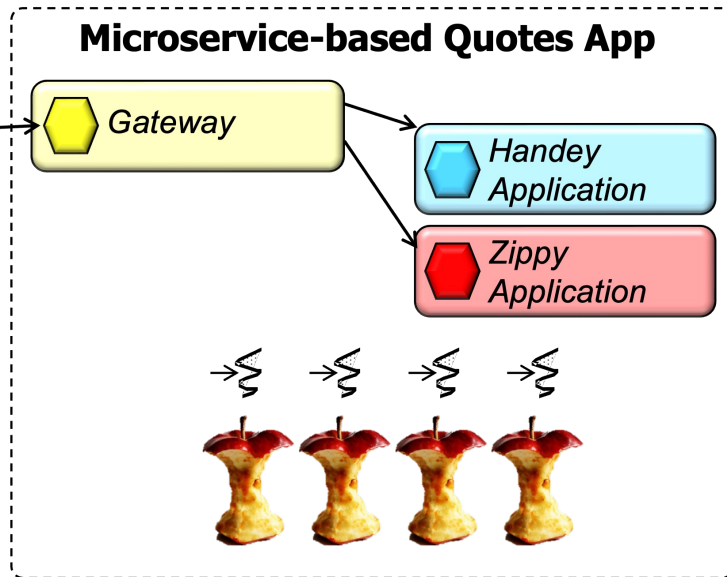
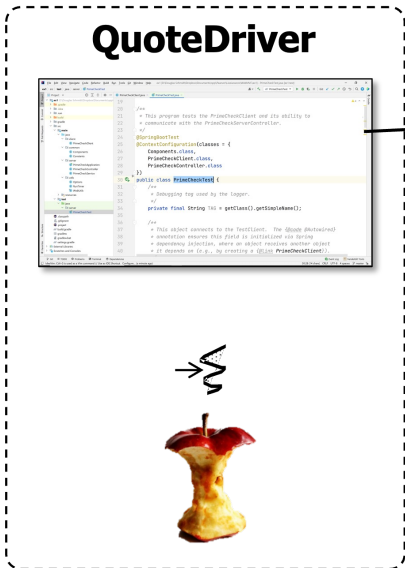


Simplifies system security & client programming complexity

- Cons
 - Some aspects of system configuration, deployment, & testing remain complicated



- Cons
 - Some aspects of system configuration, deployment, & testing remain complicated
 - Synchronous two-way requests/responses may not scale well



We'll address this limitation in our next round of case studies that use WebFlux

End of the QuoteServices App Case Study: Overview