

The BerraQuotes 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**

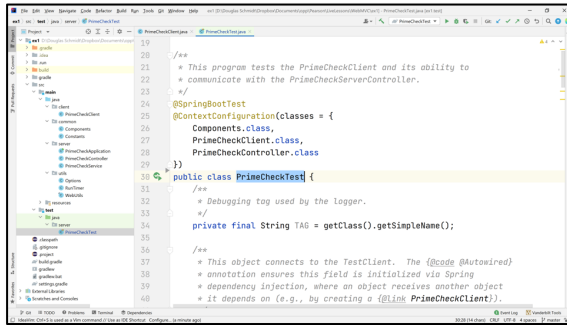
**Vanderbilt University
Nashville, Tennessee, USA**



Learning Objectives in this Part of the Lesson

- Understand how functional programming & various Java frameworks are applied in a case study using Spring WebMVC to obtain Yogi Berra quotes

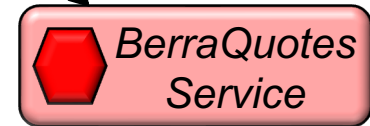
BerraQuotesTest



```
19 /**  
20  * This program tests the PrimeCheckClient and its ability to  
21  * communicate with the PrimeCheckServerController.  
22  */  
23  
24 @SpringBootTest  
25 @ContextConfiguration(classes = {  
26     Components.class,  
27     PrimeCheckClient.class,  
28     PrimeCheckController.class  
29 })  
30 public class PrimeCheckTest {  
31     /**  
32      * Debugging tag used by the logger.  
33      */  
34     private final String TAG = getClass().getSimpleName();  
35  
36     /**  
37      * This object connects to the TestClient. The @Autowired  
38      * annotation ensures this field is initialized via Spring  
39      * dependency injection, where an object receives another object  
40      * it depends on (e.g., by creating a @Link PrimeCheckClient).  
41     */  
42     private PrimeCheckClient client;  
43  
44     /**  
45      * This method tests the PrimeCheckClient and its ability to  
46      * communicate with the PrimeCheckServerController.  
47     */  
48     @Test  
49     public void testPrimeCheckClient() {  
50         // TODO: Add test logic here  
51     }  
52 }
```



BerraQuotesApplication



*HTTP GET
requests/
responses*



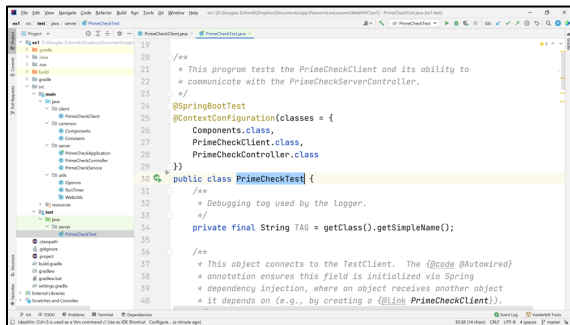
See github.com/douglasraigschmidt/LiveLessons/tree/master/WebMVC/ex2

Overview of the Berra Quotes App Case Study

Overview of the BerraQuotes App Case Study

- This case study shows how Spring WebMVC can be used to send & receive HTTP GET requests to/from a sequential microservice

BerraQuotesTest

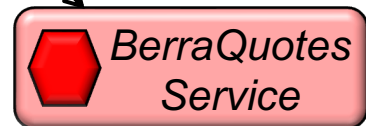
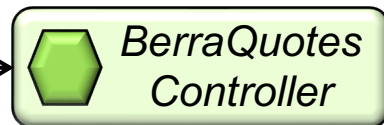


```
20 /**  
21  * This program tests the PrimeCheckClient and its ability to  
22  * communicate with the PrimeCheckServerController.  
23  */  
24  
25 @SpringBootTest  
26 @ContextConfiguration(classes = {  
27     Components.class,  
28     PrimeCheckClient.class,  
29     PrimeCheckController.class  
30 })  
31 public class PrimeCheckTest {  
32     /**  
33      * Debugging tag used by the logger.  
34      */  
35     private final String TAG = getClass().getSimpleName();  
36  
37     /**  
38      * This object connects to the TestClient. The @Autowired  
39      * annotation ensures this field is initialized via Spring  
40      * dependency injection, where an object receives another object  
41      * it depends on (e.g., by creating a @Link PrimeCheckClient).  
42     */  
43     @Autowired  
44     PrimeCheckClient primeCheckClient;  
45 }
```



*HTTP GET
requests/
responses*

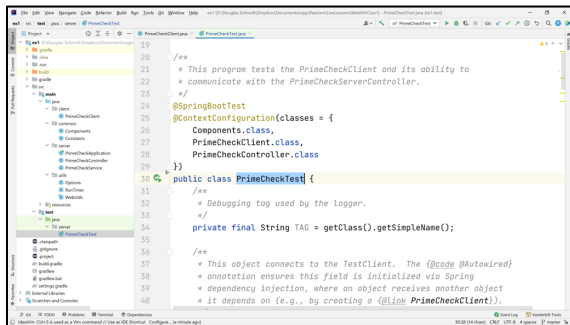
BerraQuotesApplication



Overview of the BerraQuotes App Case Study

- This case study shows how Spring WebMVC can be used to send & receive HTTP GET requests to/from a sequential microservice

BerraQuotesTest



```
20 /**  
21  * This program tests the PrimeCheckClient and its ability to  
22  * communicate with the PrimeCheckServerController.  
23  */  
24 @SpringBootTest  
25 @ContextConfiguration(classes = {  
26     Components.class,  
27     PrimeCheckClient.class,  
28     PrimeCheckController.class  
29 })  
30 public class PrimeCheckTest {  
31     /**  
32      * Debugging tag used by the logger.  
33      */  
34     private final String TAG = getClass().getSimpleName();  
35   
36     /**  
37      * This object connects to the TestClient. The @code @Autowired  
38      * annotation ensures this field is initialized via Spring  
39      * dependency injection, where an object receives another object  
40      * it depends on (e.g., by creating a @code @Link PrimeCheckClient).
```

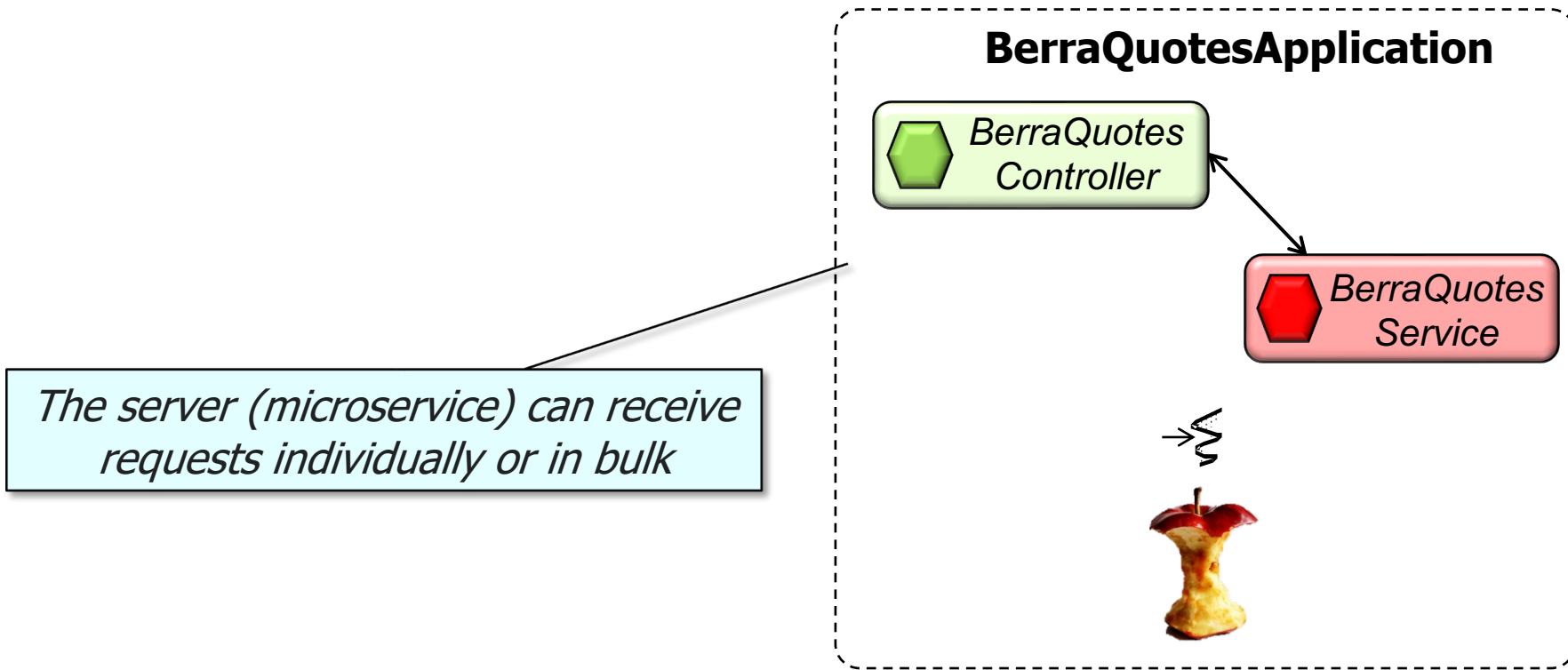
The client can send requests individually or in bulk



See [WebMVC/ex2/src/test/java/berraquotes/client](https://github.com/leandromoreira/webmvc-ex2/src/test/java/berraquotes/client)

Overview of the BerraQuotes App Case Study

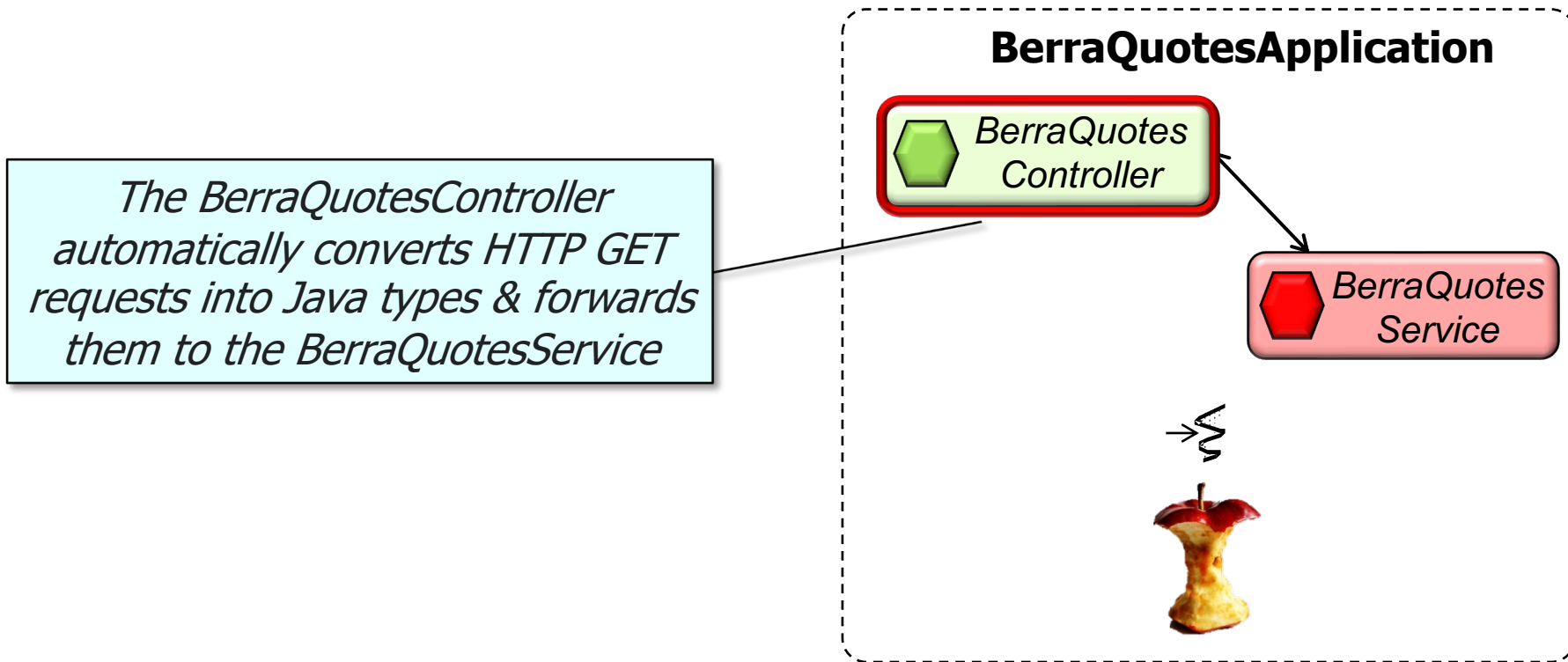
- This case study shows how Spring WebMVC can be used to send & receive HTTP GET requests to/from a sequential microservice



See [WebMVC/ex2/src/main/java/server](https://github.com/spring-projects/spring-webmvc/tree/master/samples/ex2/src/main/java/server)

Overview of the BerraQuotes App Case Study

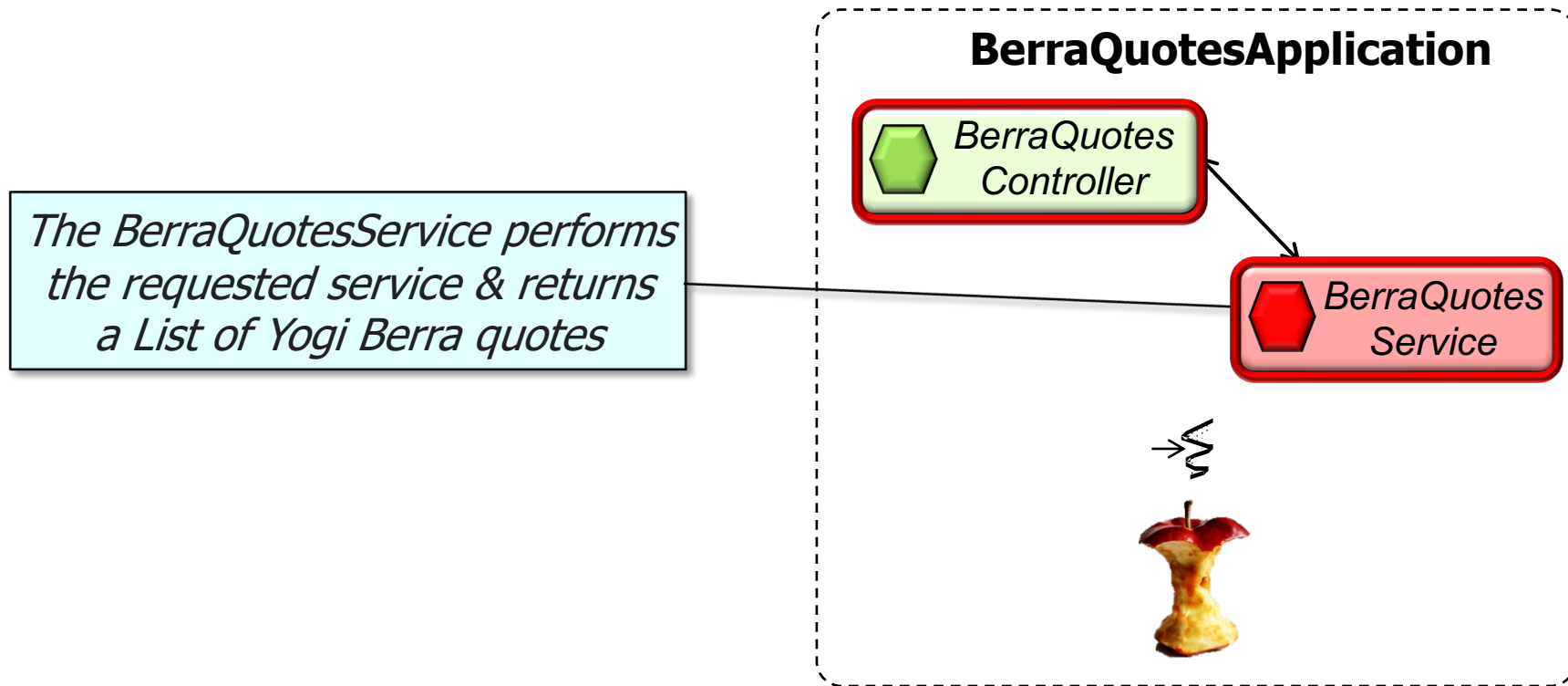
- This case study shows how Spring WebMVC can be used to send & receive HTTP GET requests to/from a sequential microservice



See WebMVC/ex2/src/main/java/berraquotes/server/BerraQuotesController.java

Overview of the BerraQuotes App Case Study

- This case study shows how Spring WebMVC can be used to send & receive HTTP GET requests to/from a sequential microservice

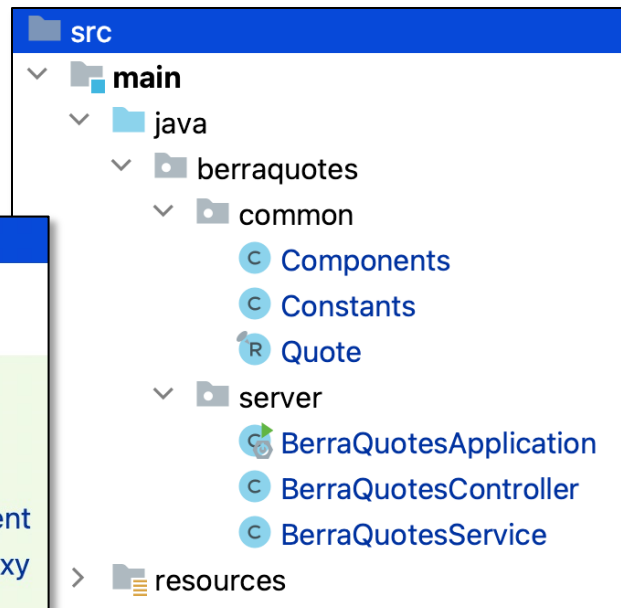
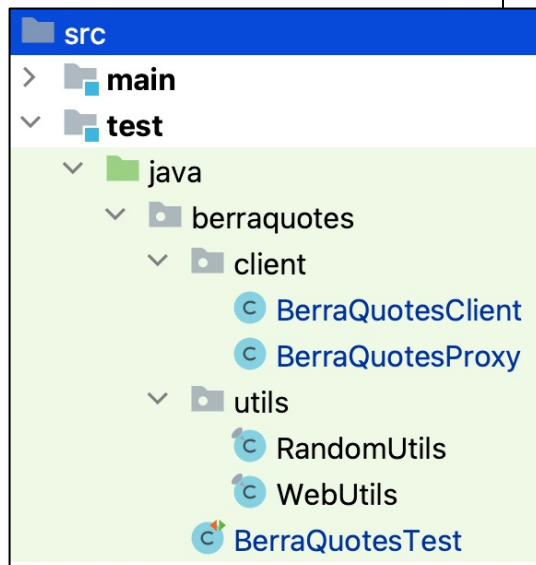


See WebMVC/ex2/src/main/java/berraquotes/server/BerraQuotesService.java

Structure of the BerraQuotes App Project

Structure of the BerraQuotes App Project

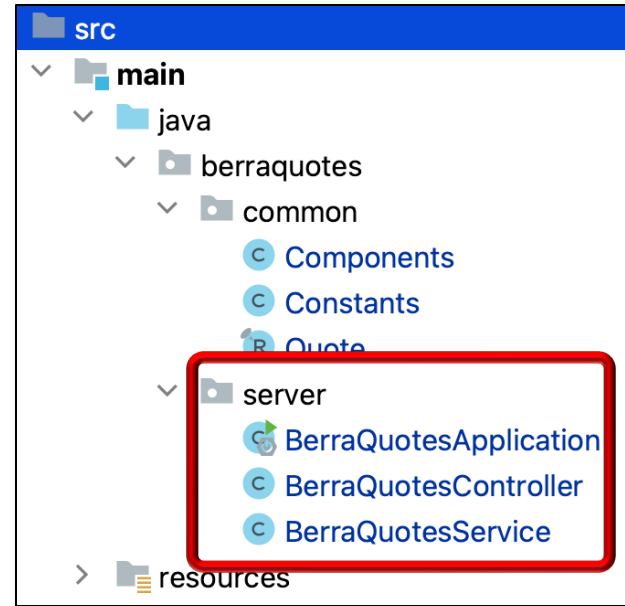
- The BerraQuotes App project source code is organized into several packages



See github.com/douglasraigschmidt/LiveLessons/tree/master/WebMVC/ex2

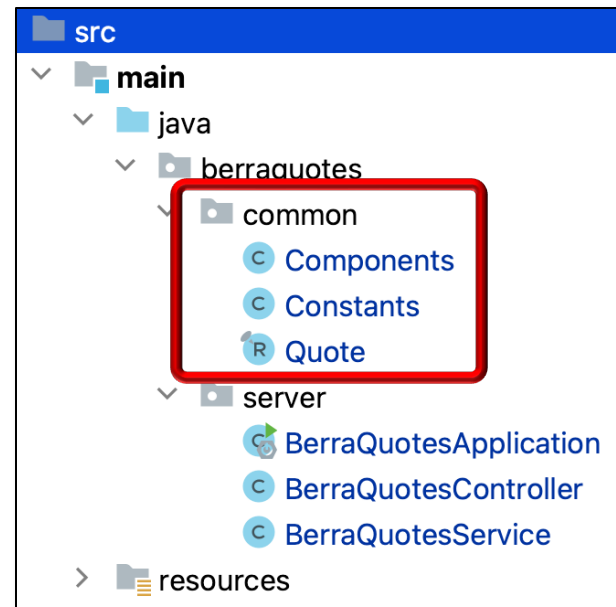
Structure of the BerraQuotes App Project

- The BerraQuotes App project source code is organized into several packages
 - main
 - berraquotes
 - Contains the “app” entry point, the controller, & the service



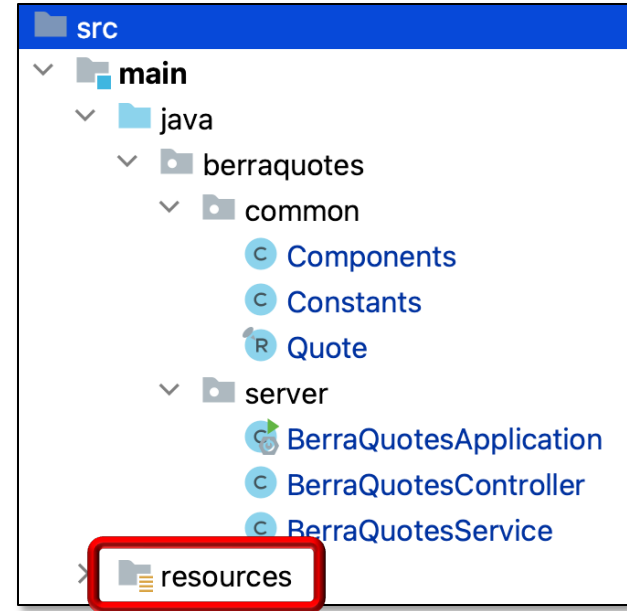
Structure of the BerraQuotes App Project

- The BerraQuotes App project source code is organized into several packages
 - main
 - berraquotes
 - Contains the “app” entry point, the controller, & the service
 - Consolidates various project-specific helper classes & the model



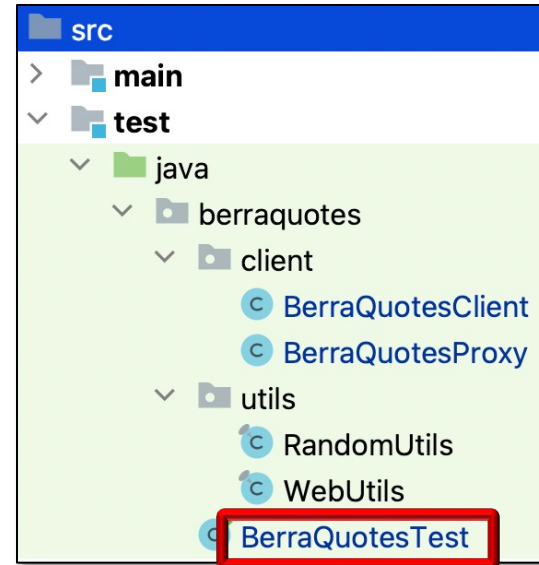
Structure of the BerraQuotes App Project

- The BerraQuotes App project source code is organized into several packages
 - main
 - berraquotes
 - common
 - Components
 - Constants
 - Quote
 - server
 - BerraQuotesApplication
 - BerraQuotesController
 - BerraQuotesService
 - resources



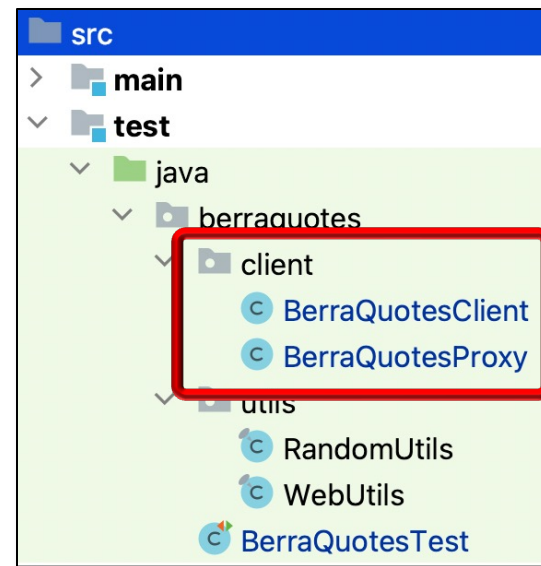
Structure of the BerraQuotes App Project

- The BerraQuotes App project source code is organized into several packages
 - test
 - BerraQuotesTest
 - This test driver initiates calls to the BerraQuotes microservice



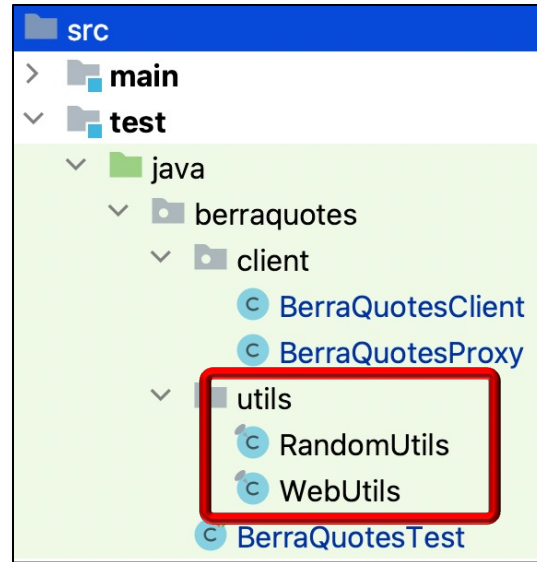
Structure of the BerraQuotes App Project

- The BerraQuotes App project source code is organized into several packages
 - test
 - BerraQuotesTest
 - client
 - Sends HTTP GET requests to the BerraQuotes microservice



Structure of the BerraQuotes App Project

- The BerraQuotes App project source code is organized into several packages
 - test
 - BerraQuotesTest
 - client
 - BerraQuotesClient
 - BerraQuotesProxy
 - utils
 - RandomUtils
 - WebUtils
 - Consolidates various reusable helper classes



End of the BerraQuotes App Case Study: Overview