The QuoteServices App Case Study: Zippy Microservice Structure & Functionality (Part 1)

Douglas C. Schmidt
d.schmidt@vanderbilt.edu
www.dre.vanderbilt.edu/~schmidt

Professor of Computer Science

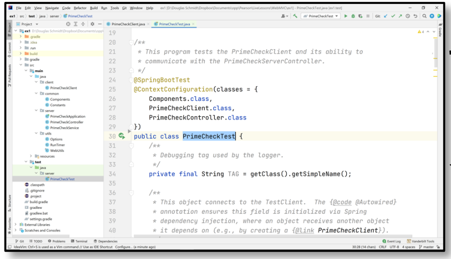
Institute for Software
Integrated Systems

Vanderbilt University
Nashville, Tennessee, USA

Learning Objectives in this Part of the Lesson

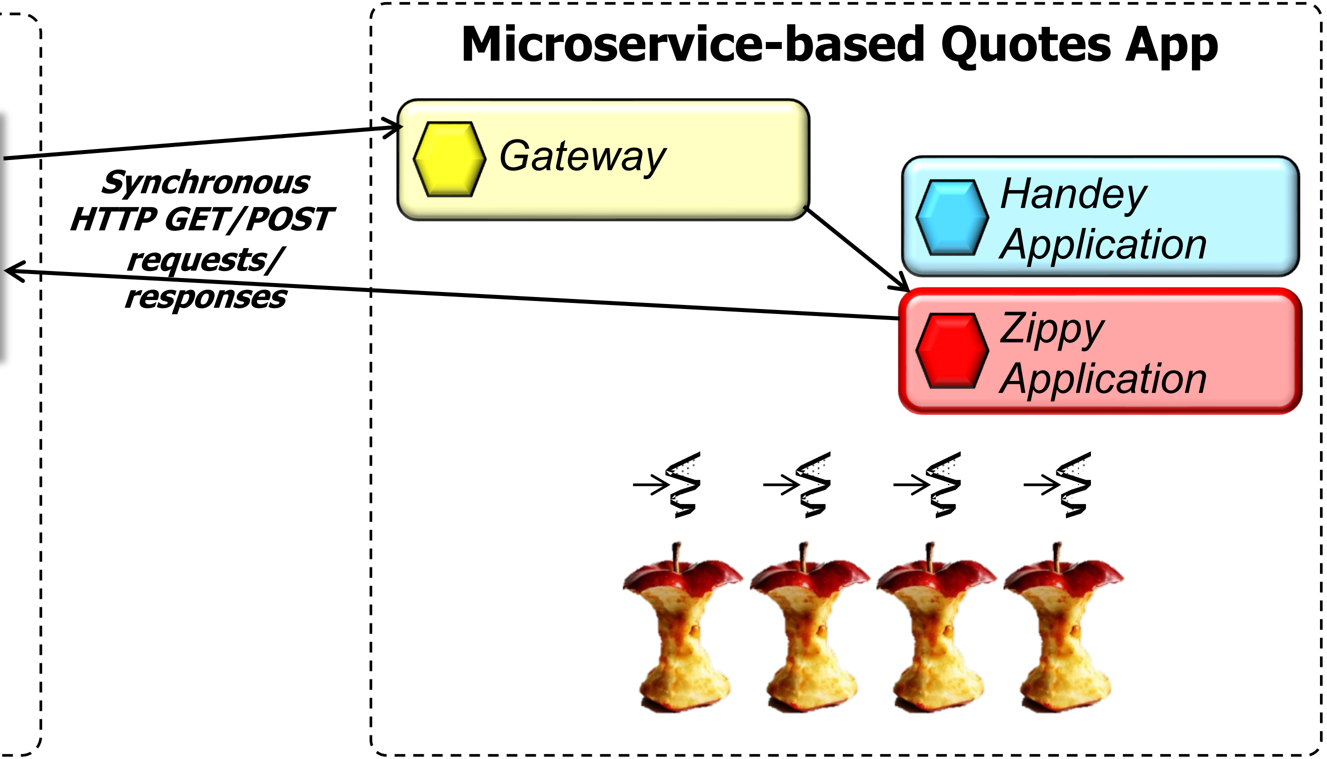
- Understand the structure & functionality of the ZippyApplication microservice & how it applies the Jakarta Persistence API (JPA)

QuoteDriver



```
/**
 * This program tests the PrimeCheckClient and its ability to
 * communicate with the PrimeCheckServerController.
 */
@SpringBootTest
@ContextConfiguration(classes = {
    Components.class,
    PrimeCheckClient.class,
    PrimeCheckController.class
})
public class PrimeCheckTest {
    /**
     * Debugging tag used by the logger.
     */
    private final String TAG = getClass().getSimpleName();

    /**
     * This object connects to the TestClient.  The {@code @Autowired}
     * annotation ensures this field is initialized via Spring
     * dependency injection, where an object receives another object
     * it depends on (e.g., by creating a {@link PrimeCheckClient}).
```



Microservice-based Quotes App
Synchronous HTTP GET/POST requests/ responses
Gateway
Handey Application
Zippy Application

We focus on many new Spring annotations

Structure & Functionality of the ZippyApplication

Structure & Functionality of the ZippyApplication

- Provides the entry point into the Spring WebMVC-based version of the Zippy Quote microservice

```
@SpringBootApplication
@EntityScan(basePackageClasses =
            {Quote.class})
@EnableJpaRepository(basePackageClasses =
                {JPAQuoteRepository.class})
@EnableCaching
public class ZippyApplication
       extends BaseApplication {




  public static void main(String[] args) {
    run(ZippyApplication.class, args);
  }
}
```

See quoteservices/microservices/handey/ZippyController.java

Structure & Functionality of the ZippyApplication

- Provides the entry point into the Spring WebMVC-based version of the Zippy Quote microservice

```
@SpringBootApplication
@EntityScan(basePackageClasses =
            {Quote.class})
@EnableJpaRepository(basePackageClasses =
                A    {JPAQuoteRepository.class})
@EnableCaching
public class ZippyApplication
       extends BaseApplication {
```

BaseApplication defines run() method used by HandeyApplication & ZippyApplication

```
  public static void main(String[] args) {
    run(ZippyApplication.class, args);
  }
}
```

See microservices/src/main/java/edu/vandy/quoteservices/common/BaseApplication.java

Structure & Functionality of the ZippyApplication

- Provides the entry point into the Spring WebMVC-based version of the Zippy Quote microservice

```
@SpringBootApplication
@EntityScan(basePackageClasses =
             {Quote.class})
@EnableJpaRepository(basePackageClasses =
                 {JPAQuoteRepository.class})
@EnableCaching
public class ZippyApplication
       extends BaseApplication {

  public static void main(String[] args) {
    run(ZippyApplication.class, args);
  }
}
```

Main entry-point calls BaseApplication helper method to build & run the microservice

Structure & Functionality of the ZippyApplication

- Provides the entry point into the Spring WebMVC-based version of the Zippy Quote microservice

```
@SpringBootApplication
@EntityScan(basePackageClasses =
               {Quote.class})
@EnableJpaRepository(basePackageClasses =
                   {JPAQuoteRepository.class})
@EnableCaching
public class ZippyApplication
       extends BaseApplication {
```

Allows auto-configuration, component scan & be able to define extra configuration

```
  public static void main(String[] args) {
    run(ZippyApplication.class, args);
  }
}
```

See [reference/html/using-boot-using-springbootapplication-annotation.html]()

Structure & Functionality of the ZippyApplication

- Provides the entry point into the Spring WebMVC-based version of the Zippy Quote microservice

```
@SpringBootApplication
@EntityScan(basePackageClasses =
                {Quote.class})
@EnableJpaRepository(basePackageClasses =
                    {JPAQuoteRepository.class})
@EnableCaching
public class ZippyApplication
       extends BaseApplication {
```

Configures the packages used by auto-configuration when scanning for entity classes

```
    public static void main(String[] args) {
      run(ZippyApplication.class, args);
    }
}
```

See springframework/boot/autoconfigure/domain/EntityScan.html

Structure & Functionality of the ZippyApplication

- Provides the entry point into the Spring WebMVC-based version of the Zippy Quote microservice

```
@SpringBootApplication
@EntityScan(basePackageClasses =
                {Quote.class})
@EnableJpaRepository(basePackageClasses =
                    {JPAQuoteRepository.class})
@EnableCaching
public class ZippyApplication
       extends BaseApplication {

  public static void main(String[] args) {
    run(ZippyApplication.class, args);
  }
}
```

Enable JPA repositories by scanning this class for Spring Data repositories

See springframework/data/jpa/repository/config/EnableJpaRepositories.html

Structure & Functionality of the ZippyApplication

- Provides the entry point into the Spring WebMVC-based version of the Zippy Quote microservice

```
@SpringBootApplication
@EntityScan(basePackageClasses =
            {Quote.class})
@EnableJpaRepository(basePackageClasses =
                {JPAQuoteRepository.class})
@EnableCaching
public class ZippyApplication
       extends BaseApplication {

  public static void main(String[] args) {
    run(ZippyApplication.class, args);
  }
}
```

Enables Spring's annotation-driven cache management capability

See javadoc-api/org/springframework/cache/annotation/EnableCaching.html

Structure & Functionality of the ZippyController

Structure & Functionality of the ZippyController

- Client HTTP GET requests are mapped to endpoint handler methods via the ZippyController subclass

```
@RestController
public class ZippyController
       extends BaseController
                 <List<Quote>> {
}
```

See quoteservices/microservices/handey/ZippyController.java

Structure & Functionality of the ZippyController

- Client HTTP GET requests are mapped to endpoint handler methods via the ZippyController subclass

```
@RestController
public class ZippyController
       extends BaseController
                 <List<Quote>> {
}
```

BaseController defines the endpoint handler methods used by both HandeyController & ZippyController

See microservices/src/main/java/edu/vandy/quoteservices/common/BaseController.java

Structure & Functionality of the ZippyController

- Client HTTP GET requests are mapped to endpoint handler methods via the ZippyController subclass

```
@RestController
public class ZippyController
       extends BaseController
                 <List<Quote>> {



}
```

Defines the classic Java List type returned by endpoint handler methods in the ZippyController

Structure & Functionality of the ZippyController

- Client HTTP GET requests are mapped to endpoint handler methods via the ZippyController subclass

```
@RestController
public class ZippyController
       extends BaseController
                <List<Quote>> {
}
```

This annotation ensures request handling methods in the controller class automatically serialize return objects into HttpResponse objects

See www.baeldung.com/spring-controller-vs-restcontroller

End of the QuoteServices App Case Study: Zippy MicroService Structure & Functionality (Part 1)