The QuoteServices App Case Study: Common Structure & Functionality

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 common module in the QuoteServices project, e.g., BaseApplication & Quote

This module provides a customizable foundation for the Quote microservices
Structure & Functionality of the BaseApplication
Structure & Functionality of the BaseApplication

- BaseApplication generically defines a Eureka-enabled microservice

```java
public class BaseApplication {

    public static void run(Class<?> clazz, String[] args) {
        var name = getName(clazz);
        var app = new SpringApplicationBuilder(clazz)
            .properties(singletonMap("spring.application.name", name))
            .build();
        app.setAdditionalProfiles(name);
        app.setLazyInitialization(true);
        app.run(args);
    }

    ...
}
```

This class is extended by Handey Application & ZippyApplication

See common/src/main/java/edu/vandy/quoteservices/common/BaseApplication.java
Structure & Functionality of the BaseApplication

- BaseApplication generically defines a Eureka-enabled microservice

```java
public class BaseApplication {

    public static void run(Class<?> clazz, String[] args) {
        var name = getName(clazz);
        var app = new SpringApplicationBuilder(clazz)
            .properties(singletonMap("spring.application.name", name))
            .build();
        app.setAdditionalProfiles(name);
        app.setLazyInitialization(true);
        app.run(args);
    }

    ...
```

Builds a Spring Boot microservice with the clazz name
Structure & Functionality of the BaseApplication

- BaseApplication generically defines a Eureka-enabled microservice

```java
public class BaseApplication {

    public static void run(Class<?> clazz, String[] args) {
        var name = getName(clazz);
        var app = new SpringApplicationBuilder(clazz)
            .properties(singletonMap("spring.application.name", name))
            ...
    }

    private static String getName(Class<?> clazz) {
        String pkg = clazz.getPackage().getName();
        return pkg.substring(pkg.lastIndexOf('.') + 1);
    }
}
```

Get the name of the microservice application
Structure & Functionality of the BaseApplication

- BaseApplication generically defines a Eureka-enabled microservice

```java
public class BaseApplication {

    public static void run(Class<?> clazz, String[] args) {
        var name = getName(clazz);
        var app = new SpringApplicationBuilder(clazz)
            .properties(singletonMap("spring.application.name", name))
            ...
    }

    private static String getName(Class<?> clazz) {
        String pkg = clazz.getPackage().getName();
        return pkg.substring(pkg.lastIndexOf('.') + 1);
    }

    Set the name as a property of the app, which can then be used with Eureka
```
Structure & Functionality of the Quote Class
Structure & Functionality of the Quote Class

- The Quote class stores quote objects in the JPA & R2DBC databases

```java
@Entity
@Table(name = "QUOTE")
public class Quote implements Comparable<Quote> {

    @Id // For JPA
    @org.springframework.data.annotation.Id // For R2DBC
    public Integer id;

    public String quote;

    @Override
    public int compareTo(Quote other) {
        return this.quote.compareTo(other.quote);
    }
    ...
}
```

See common/src/main/java/edu/vandy/quoteservices/common/Quote.java
Structure & Functionality of the Quote Class

• The Quote class stores quote objects in the JPA & R2DBC databases

```java
@Entity
@Table(name = "QUOTE")
public class Quote implements Comparable<Quote> {
    @Id // For JPA
    @org.springframework.data.annotation.Id // For R2DBC
    public Integer id;

    public String quote;

    @Override
    public int compareTo(Quote other) {
        return this.quote.compareTo(other.quote);
    }
    ...
```
Structure & Functionality of the Quote Class

- The Quote class stores quote objects in the JPA & R2DBC databases

```java
@Entity
@Table(name = "QUOTE")
public class Quote implements Comparable<Quote> {
    @Id // For JPA
    @org.springframework.data.annotation.Id // For R2DBC
    public Integer id;

    public String quote;

    @Override
    public int compareTo(Quote other) {
        return this.quote.compareTo(other.quote);
    }
    ...
```
Structure & Functionality of the Quote Class

- The Quote class stores quote objects in the JPA & R2DBC databases

```java
@Entity
@Table(name = "QUOTE")
public class Quote implements Comparable<Quote> {

    @Id // For JPA
    @org.springframework.data.annotation.Id // For R2DBC
    public Integer id;

    public String quote;

    @Override
    public int compareTo(Quote other) {
        return this.quote.compareTo(other.quote);
    }
    ...
```

This interface enables sorting of Quote objects
Structure & Functionality of the Quote Class

- The Quote class stores quote objects in the JPA & R2DBC databases

```java
@Entity
@Table(name = "QUOTE")
public class Quote implements Comparable<Quote> {
    @Id // For JPA
    @org.springframework.data.annotation.Id // For R2DBC
    public Integer id;

    public String quote;

    @Override
    public int compareTo(Quote other) {
        return this.quote.compareTo(other.quote);
    }
}
```

The ID # of the quote
Structure & Functionality of the Quote Class

- The Quote class stores quote objects in the JPA & R2DBC databases

```java
@Entity
@Table(name = "QUOTE")
public class Quote
    implements Comparable<Quote> {

    @Id // For JPA
    @org.springframework.data.annotation.Id // For R2DBC
    public Integer id;

    public String quote;

    @Override
    public int compareTo(Quote other) {
        return this.quote.compareTo(other.quote);
    }
    ...
```

Annotations needed by each type of database repository to specify ‘id’ as primary key

It can be tricky to design entity classes that work with both JPA & R2DBC
• The Quote class stores quote objects in the JPA & R2DBC databases

```java
@Entity
@Table(name = "QUOTE")
public class Quote implements Comparable<Quote> {
    @Id // For JPA
    @org.springframework.data.annotation.Id // For R2DBC
    public Integer id;

    public String quote;

    @Override
    public int compareTo(Quote other) {
        return this.quote.compareTo(other.quote);
    }
    ...
```

A Zippy or Handey quote that’s stored in the database
Structure & Functionality of the Quote Class

• The Quote class stores quote objects in the JPA & R2DBC databases

```java
@Entity
@Table(name = "QUOTE")
public class Quote implements Comparable<Quote> {
    @Id // For JPA
    @org.springframework.data.annotation.Id // For R2DBC
    public Integer id;

    public String quote;

    @Override
    public int compareTo(Quote other) {
        return this.quote.compareTo(other.quote);
    }
    ...
}
```

Compare two Quote objects to determine how they relation for sorting purposes
End of the QuoteServices App Case Study: Common Structure & Functionality