

The QuoteServices App Case Study: Common Structure & Functionality

Douglas C. Schmidt

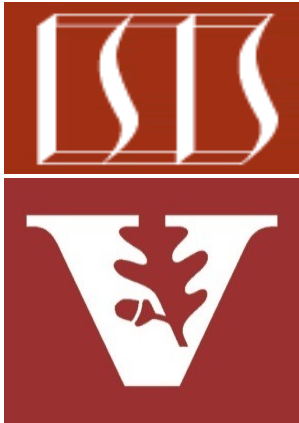
d.schmidt@vanderbilt.edu

www.dre.vanderbilt.edu/~schmidt

Professor of Computer Science

**Institute for Software
Integrated Systems**

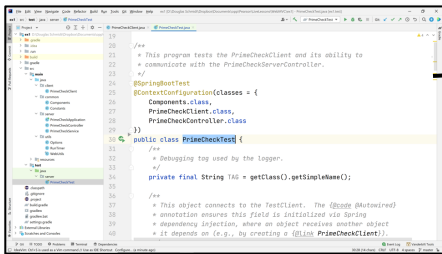
**Vanderbilt University
Nashville, Tennessee, USA**



Learning Objectives in this Part of the Lesson

- Understand the structure & functionality of the common module in the QuoteServices project, e.g., BaseApplication & Quote

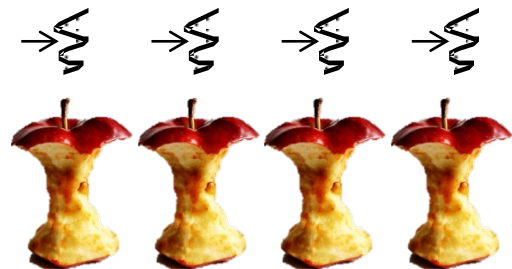
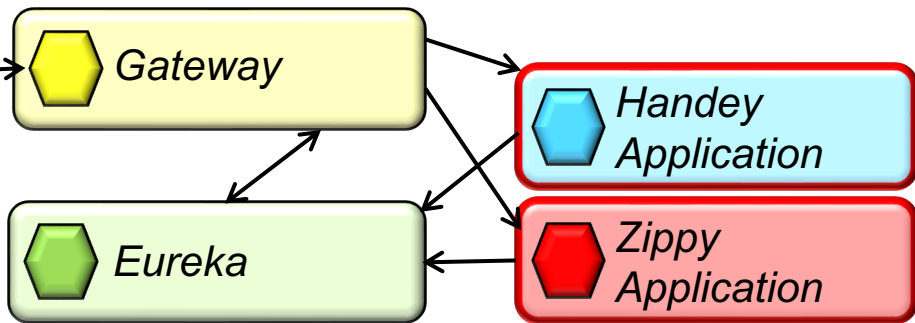
QuoteDriver



*HTTP GET/POST
requests/
responses*



Microservice-based Quotes App



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

```
public class BaseApplication {
```

This class is extended by Handey Application & ZippyApplication

```
    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);  
    }  
    ...
```

See [common/src/main/java/edu/vandy/quoteservices/common/BaseApplication.java](https://github.com/vandy/quoteservices/blob/master/common/src/main/java/edu/vandy/quoteservices/common/BaseApplication.java)

Structure & Functionality of the BaseApplication

- BaseApplication generically defines a Eureka-enabled microservice

```
public class BaseApplication {
```

Builds a Spring Boot micro service with the clazz name

```
    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);  
    }  
    ...
```

Structure & Functionality of the BaseApplication

- BaseApplication generically defines a Eureka-enabled microservice

```
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))
```

```
        ...
```

```
    }
```

*Get the name of the
microservice application*

```
private static String getName(Class<?> clazz) {
```

```
    String pkg = clazz.getPackage().getName();
```

```
    return pkg.substring(pkg.lastIndexOf('.') + 1);
```

```
}
```

Structure & Functionality of the BaseApplication

- BaseApplication generically defines a Eureka-enabled microservice

```
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))
```

*Set the name as a property of the app,
which can then be used with Eureka*

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

Structure & Functionality of the Quote Class

Structure & Functionality of the Quote Class

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

```
@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](https://github.com/vandy/quoteservices/blob/master/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

@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);

} ...

Indicate that a Quote is a persistence object stored as a record in the database

Structure & Functionality of the Quote Class

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

@Entity

@Table(name = "QUOTE")

Specifies the table in the database with which each Quote entity is mapped

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

```
@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

```
@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

```
@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

Structure & Functionality of the Quote Class

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

```
@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

```
@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