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 ZippyController/ZippyService microservice & how it applies the Jakarta Persistence API (JPA)

QuoteDriver

Microservice-based Quotes App

- Gateway
- Handey Application
- Eureka
- Zippy Application

This microservice also uses Java parallel & sequential streams a bit..
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

```java
@SpringBootApplication
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

```java
@SpringBootApplication
public class ZippyApplication extends BaseApplication {
    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

```java
@SpringBootApplication
public class ZippyApplication
    extends BaseApplication {

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

Main entry-point method calls the BaseApplication helper method to build & run the microservice
Structure & Functionality of the ZippyController
Structure & Functionality of the ZippyController

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

```java
@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

```java
@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

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

}
```

Defines the classic Java 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

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

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

See [www.baeldung.com/spring-controller-vs-restcontroller](http://www.baeldung.com/spring-controller-vs-restcontroller)
Structure & Functionality of the QuoteRepository
Structure & Functionality of the QuoteRepository

- JPAQuoteRepository is a Jakarta Persistence API (JPA)-based repository that contains a database of Quote objects.

```java
@Service
public class JPAQuoteRepository
    extends JPARepository<Quote, Integer>, MultiqQueryRepository {

    List<Quote> findByQuoteContainingIgnoreCase (String query);
}
```

See en.wikipedia.org/wiki/Jakarta_Persistence
Structure & Functionality of the QuoteRepository

- JPAQuoteRepository is a Jakarta Persistence API (JPA)-based repository that contains a database of Quote objects

@Service
class JPAQuoteRepository
    extends JPARepository<Quote, Integer>,
    MultiQueryRepository {

    List<Quote> findByQuoteContainingIgnoreCase(String query);
}
Structure & Functionality of the QuoteRepository

- JPAQuoteRepository is a Jakarta Persistence API (JPA)-based repository that contains a database of Quote objects

```java
@Service
public class JPAQuoteRepository extends JPARepository<Quote, Integer>, MultiQueryRepository {
    List<Quote> findByQuoteContainingIgnoreCase(String query);
}
```

Provides methods for basic CRUD operations (e.g., save(), findAll(), findById(), findAllById(), etc.) that enable interactions with a database while writing no SQL.

See [springframework/data/jpa/repository/JpaRepository.html](http://springframework/data/jpa/repository/JpaRepository.html)
Structure & Functionality of the QuoteRepository

- JPAQuoteRepository is a Jakarta Persistence API (JPA)-based repository that contains a database of Quote objects

```java
@Service
class JPAQuoteRepository
    extends JPARepository<Quote, Integer>,
    MultiQueryRepository {

    List<Quote> findByQuoteContainingIgnoreCase(String query);
}
```

Provides a method for returning a List of Quote objects that match all queries in a List of String objects

See quoteservices/microservices/zippy/MultiQueryRepository.java
Structure & Functionality of the QuoteRepository

- JPAQuoteRepository is a Jakarta Persistence API (JPA)-based repository that contains a database of Quote objects

```java
@Service
public class JPAQuoteRepository
    extends JPARepository<Quote, Integer>,
    MultiQueryRepository
{
    List<Quote> findByQuoteContainingIgnoreCase(String query);
}
```

Find all Quote rows in the database that contain the query String (ignoring case)
### Structure & Functionality of the QuoteRepository

- **JPAQuoteRepository** is a Jakarta Persistence API (JPA)-based repository that contains a database of Quote objects.

```java
@Service
public class JPAQuoteRepository
    extends JPARepository<Quote, Integer>,
    MultiQueryRepository
{
    List<Quote> findByQuoteContainingIgnoreCase(String query);
}
```

*This JPA Data API method implementation is automatically generated as* 

```sql
SELECT * FROM quote WHERE LOWER(quote) LIKE LOWER('%{query}%')
```

See part 2 of this lesson on "Zippy Microservice Structure & Functionality"
Structure & Functionality of the ZippyService
Structure & Functionality of the ZippyService

- ZippyService defines implementation methods called by ZippyController

```java
@Service
public class ZippyService implements BaseService<List<Quote>> {
    @Autowired
    private JPAQuoteRepository mRepository;
    ...
```
Structure & Functionality of the ZippyService

- ZippyService defines implementation methods called by ZippyController

```java
@Service
public class ZippyService implements BaseService<List<Quote>> {
    @Autowired private JPAQuoteRepository mRepository;
    ...
```

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

The BaseService defines the methods overridden by both HandeyService & ZippyService
Structure & Functionality of the ZippyService

- ZippyService defines implementation methods called by ZippyController

```java
@Service
public class ZippyService implements BaseService<List<Quote>> {
    @Autowired private JPAQuoteRepository mRepository;
    ...
```

*Defines the classic Java type returned by endpoint handler methods in the ZippyService*
Structure & Functionality of the ZippyService

- ZippyService defines implementation methods called by ZippyController

```java
@Service
public class ZippyService implements BaseService<List<Quote>> {
    @Autowired private JPAQuoteRepository mRepository;
    ...
}
```

This annotation indicates the class implements "business logic" & enables auto-detection & wiring of dependent classes via classpath scanning

See [www.baeldung.com/spring-component-repository-service](http://www.baeldung.com/spring-component-repository-service)
Structure & Functionality of the ZippyService

- ZippyService defines implementation methods called by ZippyController

```java
@Service
public class ZippyService implements BaseService<List<Quote>> {
    @Autowired
    private JPAQuoteRepository mRepository;
    ...
```

See [www.baeldung.com/spring-autowire](http://www.baeldung.com/spring-autowire)

This field is auto-wired by Spring's dependency injection framework
Structure & Functionality of the ZippyService

- ZippyService defines implementation methods called by ZippyController

```java
@Service
public class ZippyService implements BaseService<List<Quote>> {
    ...
    public List<Quote> getAllQuotes() {
        return mRepository.findAll();
    }
    ...
}
```

This method uses the JPA to return all the Zippy quotes

The generated SQL query would be “SELECT * FROM quote”
Structure & Functionality of the ZippyService

- ZippyService defines implementation methods called by ZippyController

```java
@Service
public class ZippyService implements BaseService<List<Quote>> {  

...  

public List<Quote> postQuotes(List<Integer> quoteIds,  
                              Boolean notUsed) {

    return mRepository  
        .findAllById(quoteIds);  
}
...```

This method uses the JPA to find all quotes matching the quoteIds.

The generated SQL query would be “SELECT * FROM quote WHERE id IN (?, ?, ...)”
Structure & Functionality of the ZippyService

- ZippyService defines implementation methods called by ZippyController

```java
@Service
public class ZippyService implements BaseService<List<Quote>> {
    ...

    public List<Quote> search(List<String> queries, Boolean parallel) {
        return StreamSupport.stream(queries.spliterator(), parallel)
                        .flatMap(query -> mRepository
                                      .findByQuoteContainingIgnoreCase(query)
                                      .stream())
                        .distinct()
                        .toList();
    }
    ...
```

Uses Java parallel & sequential streams & the JPA to find Quote objects matching any query

The SQL query is "SELECT * FROM quote WHERE LOWER(quote) LIKE LOWER("%{query}%")"
Structure & Functionality of the ZippyService

- ZippyService defines implementation methods called by ZippyController

```java
@Service
public class ZippyService implements BaseService<List<Quote>> {
    ...
    public List<Quote> searchEx(List<String> queries, Boolean notUsed) {
        return mRepository.findAllByQuoteContainingAllIn(queries);
    }
} ...
```

This method uses a custom SQL query to find all Quote objects containing all the queries.

See part 2 of this lesson on "Zippy Microservice Structure & Functionality"