The QuoteServices App Case Study: Handey 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 HandeyController & Handey Service microservice classes

**QuoteDriver**

**Microservice-based Quotes App**

- **Gateway**
- **Handy Application**
- **Eureka**
- **Zippy Application**

This microservice uses R2DBC & Project Reactor reactive types
Structure & Functionality of the HandeyApplication
Structure & Functionality of the HandeyApplication

- Provides the entry point into the Spring WebFlux-based version of the Handey Quote microservice

```java
@SpringBootApplication
@EntityScan(basePackageClasses = {Quote.class})
@EnableR2dbcRepositories(basePackageClasses = {ReactiveQuoteRepository.class})
public class HandeyApplication
    extends BaseApplication {

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

See quoteservices/microservice/HandeyApplication.java
Structure & Functionality of the HandeyApplication

- Provides the entry point into the Spring WebFlux-based version of the Handey Quote microservice

```java
@SpringBootApplication
@EntityScan(basePackageClasses = {Quote.class})
@EnableR2dbcRepositories(basePackageClasses = {ReactiveQuoteRepository.class})
public class HandeyApplication extends BaseApplication {

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

*BaseApplication defines the run() method used by both the HandeyApplication & ZippyApplication*

Structure & Functionality of the HandeyApplication

• Provides the entry point into the Spring WebFlux-based version of the Handey Quote microservice

```java
@SpringBootApplication
@EntityScan(basePackageClasses = {Quote.class})
@EnableR2dbcRepositories(basePackageClasses = {ReactiveQuoteRepository.class})
public class HandeyApplication extends BaseApplication {

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

These annotations enable this microservice to use R2DBC

See [springframework/data/r2dbc/repository/config/EnableR2dbcRepositories.html](springframework/data/r2dbc/repository/config/EnableR2dbcRepositories.html)
Structure & Functionality of the HandeyApplication

- Provides the entry point into the Spring WebFlux-based version of the Handey Quote microservice

```java
@SpringBootApplication
@EntityScan(basePackageClasses = {Quote.class})
@EnableR2dbcRepositories(basePackageClasses = {ReactiveQuoteRepository.class})
public class HandeyApplication extends BaseApplication {
    public static void main(String[] args) {
        run(HandeyApplication.class, args);
    }
}
```

The main entry-point method calls the BaseApplication helper method to build & run the Handey microservice & register with the Eureka discovery service.
Structure & Functionality of the HandeyController
Structure & Functionality of the HandeyController

- HandeyController maps HTTP GET & POST requests to endpoint handlers

```java
@RestController
public class HandeyController {
    @Autowired
    ApplicationContext applicationContext;

    @Autowired
    HandeyService mService;

    ...
}
```

See quoteservices/microservice/HandeyController.java
Structure & Functionality of the HandeyController

- HandeyController maps HTTP GET & POST requests to endpoint handlers

```java
@RestController
public class HandeyController {
    @Autowired
    ApplicationContext applicationContext;

    @Autowired
    HandeyService mService;

    ...
}
```

This annotation ensures that request handling methods in the controller class all automatically serialize return objects into HttpResponse objects
Structure & Functionality of the HandeyController

- HandeyController maps HTTP GET & POST requests to endpoint handlers

```java
@RestController
public class HandeyController {
    @Autowired
    ApplicationContext applicationContext;

    @Autowired
    HandeyService mService;

    ...
}
```

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

These fields are auto-wired by Spring’s dependency injection framework.
Structure & Functionality of the HandeyController

- HandeyController maps HTTP GET & POST requests to endpoint handlers

```java
@RestController
public class HandeyController {
    ...
    @GetMapping("/")
    public ResponseEntity<String> info() {
        return ResponseEntity.ok(applicationContext.getId() + " is alive and running at "
                                   + Thread.currentThread() + "\n");
    }
    ...
```

See [spring.io/projects/spring-cloud-netflix](spring.io/projects/spring-cloud-netflix)

A GET request used to test Eureka connectivity
HandeyController maps HTTP GET & POST requests to endpoint handlers

```java
@RestController
public class HandeyController {
    ...

    @GetMapping(GET_ALL_QUOTES)
    public Flux<Quote> getAllQuotes()
    { return mService.getAllQuotes(); }

    @PostMapping(POST_QUOTES)
    Flux<Quote> postQuotes(@RequestBody List<Integer> quoteIds)
    { return mService.postQuotes(quoteIds); }

    @PostMapping(POST_SEARCHES)
    public Flux<Quote> search(@RequestBody List<String> queries)
    { return mService.search(queries); }
    ...
}
```

See quoteservices/microservice/HandeyController.java

These methods handle GET & POST HTTP requests by forwarding to the service.
Structure & Functionality of the HandeyController

- HandeyController maps HTTP GET & POST requests to endpoint handlers

```java
@RestController
public class HandeyController {
    ...
    @GetMapping(GET_ALL_QUOTES)
    public Flux<Quote> getAllQuotes() {
        return mService.getAllQuotes();
    }

    @PostMapping(POST_QUOTES)
    Flux<Quote> postQuotes(@RequestBody List<Integer> quoteIds) {
        return mService.postQuotes(quoteIds);
    }

    @PostMapping(POST_SEARCHES)
    public Flux<Quote> search(@RequestBody List<String> queries) {
        return mService.search(queries);
    }
    ...
}
```

These methods return Flux objects based on service responses
Structure & Functionality of the HandeyController

- HandeyController maps HTTP GET & POST requests to endpoint handlers

```java
@RestController
public class HandeyController {
    ...

    @GetMapping(GET_ALL_QUOTES)
    public Flux<Quote> getAllQuotes() {
        return mService.getAllQuotes();
    }

    @PostMapping(POST_QUOTES)
    Flux<Quote> postQuotes(@RequestBody List<Integer> quoteIds) {
        return mService.postQuotes(quoteIds);
    }

    @PostMapping(POST_SEARCHES)
    public Flux<Quote> search(@RequestBody List<String> queries) {
        return mService.search(queries);
    }
    ...

These annotations ensure the HTTP GET & POST requests are properly transformed into Java method calls
```
Structure & Functionality of the HandeyService
Structure & Functionality of the HandeyService

- HandeyService defines implementation methods called by HandeyController

```java
@Service
public class HandeyService {
    @Autowired
    private ReactiveQuoteRepository mRepository;

    ...
```

See quoteservices/microservices/handey/HandeyService.java
Structure & Functionality of the HandeyService

- HandeyService defines implementation methods called by HandeyController

```java
@Service
public class HandeyService {
    @Autowired
    private ReactiveQuoteRepository 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 HandeyService

- HandeyService defines implementation methods called by HandeyController

```java
@Service
public class HandeyService {
    @Autowired
    private ReactiveQuoteRepository mRepository;

    ...
}
```

This field is auto-wired by Spring’s dependency injection framework

See [www.baeldung.com/spring-autowire](http://www.baeldung.com/spring-autowire)
Structure & Functionality of the HandeyService

- HandeyService defines implementation methods called by HandeyController

```java
@Service
public class HandeyService {
    public Flux<Quote> postQuotes(List<Integer> quoteIds) {
        return mRepository.findAllById(quoteIds);
    }

    public Flux<Quote> search(List<String> queries) {
        return mRepository.findAllByQuoteContainingAnyIn(queries);
    }

    public Flux<Quote> searchEx(List<String> queries) {
        return mRepository.findAllByQuoteContainingAllIn(queries);
    }
}
```
Structure & Functionality of the HandeyService

- HandeyService defines implementation methods called by HandeyController

```java
@Service
public class HandeyService {
    public Flux<Quote> postQuotes(List<Integer> quoteIds) {
        return mRepository.findAllById(quoteIds);
    }

    public Flux<Quote> search(List<String> queries) {
        return mRepository.findAllByQuoteContainingAnyIn(queries);
    }

    public Flux<Quote> searchEx(List<String> queries) {
        return mRepository.findAllByQuoteContainingAllIn(queries);
    }
}
```

All methods return the Flux reactive type
Structure & Functionality of the HandeyService

- HandeyService defines implementation methods called by HandeyController

```java
@Service
public class HandeyService {
    public Flux<Quote> postQuotes(List<Integer> quoteIds) {
        return mRepository.findAllById(quoteIds);
    }

    public Flux<Quote> search(List<String> queries) {
        return mRepository.findAllByQuoteContainingAnyIn(queries);
    }

    public Flux<Quote> searchEx(List<String> queries) {
        return mRepository.findAllByQuoteContainingAllIn(queries);
    }
}
```

All methods forward to the ReactiveQuoteRepository, so we could remove the service
End of the QuoteServices App Case Study: Handey MicroService Structure & Functionality (Part 1)