The QuoteServices App Case Study: Handey Microservice 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 HandeyController/Handey Service microservice & how it applies Java reactive concurrency

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

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

```java
@RestController
public class HandeyApplication extends BaseApplication {

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

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

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

```java
@RestController
public class HandeyApplication extends BaseApplication {

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

*BaseApplication defines the run() method used by both Handey Application & ZippyApplication*
Structure & Functionality of the HandeyApplication

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

```java
@RestController
public class HandeyApplication
    extends BaseApplication {

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

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

- Client HTTP GET & POST requests are mapped to endpoint handlers via the HandeyController subclass

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

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

- Client HTTP GET & POST requests are mapped to endpoint handlers via the HandeyController subclass

```java
@RestController
public class HandeyController
    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](microservices/src/main/java/edu/vandy/quoteservices/common/BaseController.java)
Structure & Functionality of the HandeyController

- Client HTTP GET & POST requests are mapped to endpoint handlers via the HandeyController subclass

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

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

- Client HTTP GET & POST requests are mapped to endpoint handlers via the HandeyController subclass

```java
@RestController
public class HandeyController
    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](http://www.baeldung.com/spring-controller-vs-restcontroller)
Structure & Functionality of the HandeyService
Structure & Functionality of the HandeyService

- HandeyService defines implementation methods called by HandeyController

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

  public List<Quote> getQuotes(List<Integer> quoteIds, Boolean parallel) {
    return Flux
      .fromIterable(quoteIds)
      .parallel()
      .runOn(parallel ? parallel() : single())
      .map(mQuotes::get)
      .sequential()
      .collectList()
      .share()
      .block();
  }
```

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

- HandeyService defines implementation methods called by HandeyController

```java
@Service
public class HandeyService implements BaseService<List<Quote>> {
    public List<Quote> getQuotes(List<Integer> quoteIds, Boolean parallel) {
        return Flux.fromIterable(quoteIds).parallel().runOn(parallel ? parallel() : single()).map(mQuotes::get).sequential().collectList().share().block();
    }
}
```

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 HandeyService

- HandeyService defines implementation methods called by HandeyController

```java
@Service
public class HandeyService implements BaseService<List<Quote>> {
   public List<Quote> getQuotes(List<Integer> quoteIds, Boolean parallel) {
      return Flux.fromIterable(quoteIds)
               .parallel()
               .runOn(parallel ? parallel() : single())
               .map(mQuotes::get)
               .sequential()
               .collectList()
               .share()
               .block();
   }
   ...
}
```

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

• HandeyService defines implementation methods called by HandeyController

```java
@Service
public class HandeyService implements BaseService<List<Quote>> {
    public List<Quote> getQuotes(List<Integer> quoteIds, Boolean parallel) {
        return Flux.fromIterable(quoteIds).parallel().runOn(parallel ? parallel() : single()).map(mQuotes::get).sequential().collectList().share().block();
    }
    ...
```
Structure & Functionality of the HandeyService

HandeyService defines implementation methods called by HandeyController

```java
@Service
public class HandeyService implements BaseService<List<Quote>> {
    public List<Quote> postQuotes(List<Integer> quoteIds,
                                   Boolean parallel) {
        return Flux
            .fromIterable(quoteIds)
            .parallel()
            .runOn(parallel ? parallel() : single())
            .map(mQuotes::get)
            .sequential()
            .collectList()
            .share()
            .block();
    }
    ...
}
```

This service uses Project Reactor Flux & ParallelFlux reactive types

End of the QuoteServices App Case Study: Handey MicroService Structure & Functionality