The QuoteServices App Case Study: Handey Microservice Structure & Functionality

Douglas C. Schmidt
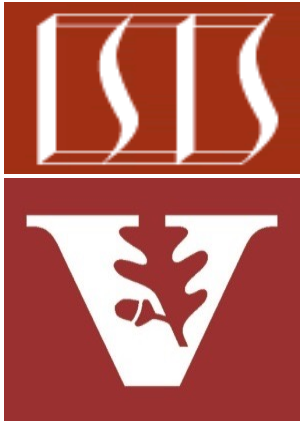### d.schmidt@vanderbilt.edu
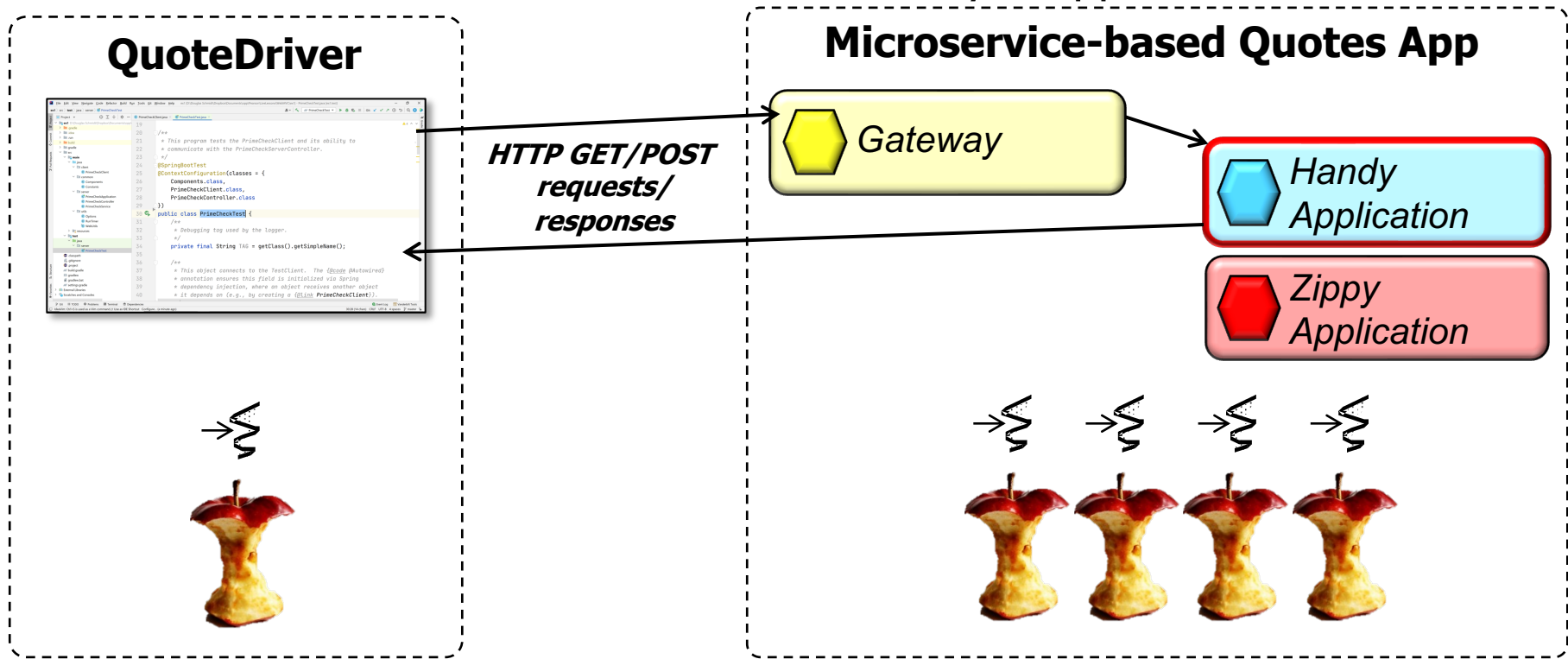### www.dre.vanderbilt.edu/~schmidt

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 classes & how Java reactive concurrency is applied

QuoteDriver

HTTP GET/POST requests/ responses

Microservice-based Quotes App

Gateway

Handy Application

Zippy Application

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
@SpringBootApplication
public class HandeyApplication
        extends BaseApplication {




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

See quoteservices/microservices/handey/HandeyApplication.java

Structure & Functionality of the HandeyApplication

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

```java
@SpringBootApplication
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

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

Structure & Functionality of the HandeyApplication

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

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



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

Allows auto-configuration, component scan & be able to define extra configuration

See reference/html/using-boot-using-springbootapplication-annotation.html

Structure & Functionality of the HandeyApplication

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

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

> Main entry-point method calls the BaseApplication helper method to build & run the microservice

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

Structure & Functionality of the HandeyController

Structure & Functionality of the HandeyController

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

```
@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

```
@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

Structure & Functionality of the HandeyController

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

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

}
```

Defines the classic Java type returned by the 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

Structure & Functionality of the HandeyService

Structure & Functionality of the HandeyService

- HandeyService defines implementation methods called by HandeyController

```
@Service
public class HandeyService implements BaseService<List<Quote>> {
    @Autowired private List<Quote> mQuotes;



    } ...
```

See quoteservices/microservices/handey/HandeyService.java

Structure & Functionality of the HandeyService

- HandeyService defines implementation methods called by HandeyController

```
@Service
public class HandeyService implements BaseService<List<Quote>> {
    @Autowired private List<Quote> mQuotes;




} ...
```

> The BaseService defines the methods overridden by both HandeyService & ZippyService

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

Structure & Functionality of the HandeyService

- HandeyService defines implementation methods called by HandeyController

```
@Service
public class HandeyService implements BaseService<List<Quote>> {
    @Autowired private List<Quote> mQuotes;




} ...
```

Defines the classic Java type returned by the endpoint handler methods in the HandeyService

However, the implementation uses Reactive programming & reactive types

Structure & Functionality of the HandeyService

- HandeyService defines implementation methods called by HandeyController

```
@Service
public class HandeyService implements BaseService<List<Quote>> {
    @Autowired private List<Quote> mQuotes;
```

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

```
} ...
```

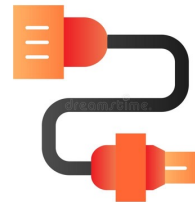See www.baeldung.com/spring-component-repository-service

- HandeyService defines implementation methods called by HandeyController

```
@Service
public class HandeyService implements BaseService<List<Quote>> {
    @Autowired private List<Quote> mQuotes;
```

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

```
} ...
```

See www.baeldung.com/spring-autowire

Structure & Functionality of the HandeyService

- HandeyService defines implementation methods called by HandeyController

```
@Service
public class HandeyService implements BaseService<List<Quote>> {
    @Autowired private List<Quote> mQuotes;
```

It stores all the Jack Handey quote objects in a non-persistent List

```
    ...
    If you define cowardice as running away at the first sign of
    danger, screaming and tripping and begging for mercy, then yes,
    Mr. Brave man, I guess I'm a coward.@
    To me, boxing is like a ballet, except there's no music, no
    choreography, and the dancers hit each other.@
    ...

} ...
```

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 method returns the List of Quote objects that are associated with the given quote IDs

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 implementation uses the Project Reactor Flux & ParallelFlux reactive types

See www.vinsguru.com/reactor-parallel-flux

Structure & Functionality of the HandeyService

- HandeyService defines implementation methods called by HandeyController

```
@Service
public class HandeyService implements BaseService<List<Quote>> {
  public List<Quote> postQuotes(List<Integer> quoteIds,
                                Boolean parallel) {
    return StreamSupport
      .stream(quoteIds.spliterator(), parallel)
      .map(mQuotes::get)
      .toList();
  } ...
```

Here's the equivalent Java streams version

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