

The QuoteServices App Case Study: Client Structure & Functionality

Douglas C. Schmidt

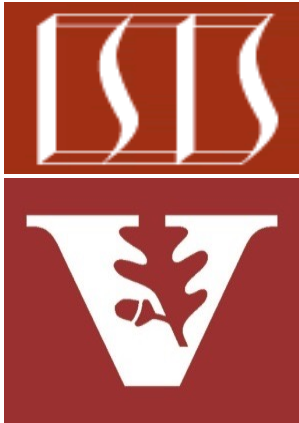
d.schmidt@vanderbilt.edu

www.dre.vanderbilt.edu/~schmidt

Professor of Computer Science

**Institute for Software
Integrated Systems**

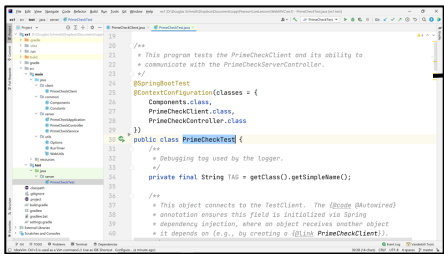
**Vanderbilt University
Nashville, Tennessee, USA**



Learning Objectives in this Part of the Lesson

- Understand the structure & functionality of QuoteClient that uses Retrofit to send /receive HTTP GET requests/responses to/from QuoteServices microservices

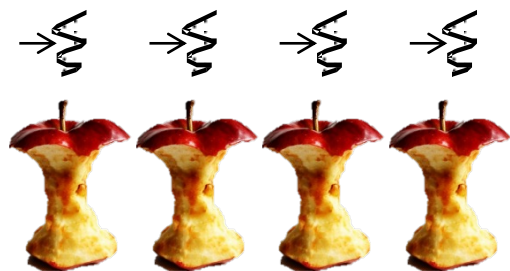
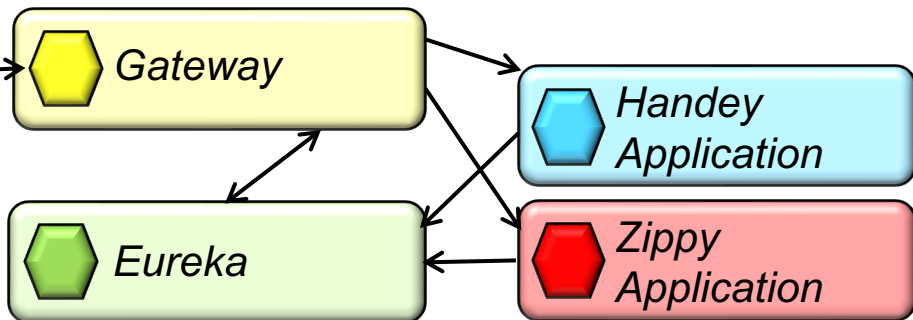
QuoteDriver



*HTTP GET/POST
requests/
responses*



Microservice-based Quotes App



The Structure & Functionality of QuoteClient Classes

The Structure & Functionality of QuoteClient Classes

- The QuoteClient classes performs synchronous remote method invocations on the API Gateway microservice to obtain Handey & Zippy quotes

@Component

```
public class QuoteClient
```

```
    @Autowired QuoteProxy mQuoteProxy;
```

```
    ...
```

```
    public List<Quote> postQuotes
```

```
        (String service, List<Integer> quoteIds, Boolean parallel)
```

```
    { mQuoteProxy.postQuotes(service, quoteIds, parallel); }
```

```
    public List<Quote> searchQuotes
```

```
        (String service, List<String> queries, Boolean parallel)
```

```
    { mQuoteProxy.searchQuotes(service, queries, parallel); }
```

```
}
```

See <WebMVC/ex4/client/src/main/java/edu/vandy/quoteservices/client/QuoteClient.java>

The Structure & Functionality of QuoteClient Classes

- The QuoteClient classes performs synchronous remote method invocations on the API Gateway microservice to obtain Handey & Zippy quotes

@Component

```
public class QuoteClient
    @Autowired QuoteProxy mQuo
```

Enables auto-detection & wiring of dependent implementation classes via classpath scanning

```
...
public List<Quote> postQuotes
    (String service, List<Integer> quoteIds, Boolean parallel)
    { mQuoteProxy.postQuotes(service, quoteIds, parallel); }
```

```
public List<Quote> searchQuotes
    (String service, List<String> queries, Boolean parallel)
    { mQuoteProxy.searchQuotes(service, queries, parallel); }
}
```

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

The Structure & Functionality of QuoteClient Classes

- The QuoteClient classes performs synchronous remote method invocations on the API Gateway microservice to obtain Handey & Zippy quotes

@Component

```
public class QuoteClient
```

```
    @Autowired QuoteProxy mQuoteProxy;
```

```
    ...
```

```
    public List<Quote> postQuotes
```

```
        (String service, List<Integer> quoteIds, Boolean parallel)
```

```
        { mQuoteProxy.postQuotes(service, quoteIds, parallel); }
```

```
    public List<Quote> searchQuotes
```

```
        (String service, List<String> queries, Boolean parallel)
```

```
        { mQuoteProxy.searchQuotes(service, queries, parallel); }
```

```
}
```

Spring's dependency injection framework auto-wires this field



See www.baeldung.com/spring-autowire

The Structure & Functionality of QuoteClient Classes

- The QuoteClient classes performs synchronous remote method invocations on the microservices to perform math operations concurrently

@Component

```
public class QuoteClient
```

```
    @Autowired QuoteProxy mQuoteProxy;
```

```
    ...
```

```
    public List<Quote> postQuotes  
        (String service, List<Integer> quoteIds, Boolean parallel)  
    { mQuoteProxy.postQuotes(service, quoteIds, parallel); }
```

Get a List of Quotes based on their quote Ids from a designated microservice

```
    public List<Quote> searchQuotes  
        (String service, List<String> queries, Boolean parallel)  
    { mQuoteProxy.searchQuotes(service, queries, parallel); }  
}
```

The Structure & Functionality of QuoteClient Classes

- The QuoteClient classes performs synchronous remote method invocations on the microservices to perform math operations concurrently

@Component

```
public class QuoteClient
```

```
    @Autowired QuoteProxy mQuoteProxy;
```

```
    ...
```

```
    public List<Quote> postQuotes
```

```
        (String service, List<Integer> quoteIds, Boolean parallel)
```

```
    { mQuoteProxy.postQuotes(service, quoteIds, parallel); }
```

Get a List of Quotes based on searching quotes in a designated microservice

```
    public List<Quote> searchQuotes
```

```
        (String service, List<String> queries, Boolean parallel)
```

```
    { mQuoteProxy.searchQuotes(service, queries, parallel); }
```

```
}
```

The Structure & Functionality of the QuoteAPI Interface

The Structure & Functionality of the QuoteProxy Class

- QuoteProxy uses an auto-generated Retrofit API class to abstracts details of making remote method invocations using HTTP

```
public interface QuoteAPI {  
    @GET("{route}" + "/" + GET_ALL_QUOTES)  
    Call<List<Quote>> getAllQuotes(@Path("route") String route);  
  
    @POST("{route}" + "/" + POST_QUOTES)  
    Call<List<Quote>> postQuotes(@Path("route") String route,  
                                @Body List<Integer> quoteIds,  
                                @Query("parallel") Boolean parallel);  
  
    @POST("{route}" + "/" + POST_SEARCHES)  
    Call<List<Quote>> search(@Path("route") String route,  
                            @Body List<String> queries,  
                            @Query("parallel") Boolean parallel);  
}
```

See WebMVC/ex4/client/src/main/java/edu/vandy/quoteservices/client/QuoteAPI.java

The Structure & Functionality of the QuoteProxy Class

- QuoteProxy uses an auto-generated Retrofit API class to abstracts details of making remote method invocations using HTTP

```
public interface QuoteAPI {  
    @GET("{route}" + "/" + GET_ALL_QUOTES)  
    Call<List<Quote>> getAllQuotes(@Path("route") String route);  
  
    @POST("{route}" + "/" + POST_QUOTES)  
    Call<List<Quote>> postQuotes(@Path("route") String route,  
                                @Body List<Integer> quoteIds,  
                                @Query("parallel") Boolean parallel);  
  
    @POST("{route}" + "/" + POST_SEARCHES)  
    Call<List<Quote>> search(@Path("route") String route,  
                             @Body List<String> queries,  
                             @Query("parallel") Boolean parallel);  
}
```

These proxy methods mimic the signature of controller methods

See WebMVC/ex4/client/src/main/java/edu/vandy/quoteservices/client/QuoteAPI.java

The Structure & Functionality of the QuoteProxy Class

The Structure & Functionality of the QuoteProxy Class

- QuoteProxy uses an auto-generated Retrofit API class to abstracts details of making remote method invocations using HTTP

```
@Component
```

```
public class QuoteProxy {  
    @Autowired QuoteAPI mQuoteAPI;  
    ...  
}
```

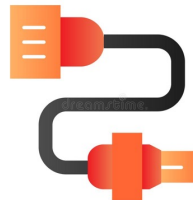
See WebMVC/ex4/client/src/main/java/edu/vandy/quoteservices/client/QuoteProxy.java

The Structure & Functionality of the QuoteProxy Class

- QuoteProxy uses an auto-generated Retrofit API class to abstracts details of making remote method invocations using HTTP

`@Component`

```
public class QuoteProxy {  
    @Autowired QuoteAPI mQuoteAPI;  
    ...  
}
```



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

See www.baeldung.com/spring-autowire

The Structure & Functionality of the QuoteProxy Class

- QuoteProxy uses an auto-generated Retrofit API class to abstracts details of making remote method invocations using HTTP

@Component

public class Components {

@Bean

public **QuoteAPI** getQuoteAPI() {

return new Retrofit

.Builder()

.baseUrl(GATEWAY_BASE_URL)

.addConverterFactory(GsonConverterFactory.create())

.build()

.create(QuoteAPI.class);

}

}

This @Bean annotation can be injected into classes using Spring's @Autowired annotation

See WebMVC/ex4/client/src/main/java/edu/vandy/quoteservices/common/Components.java

The Structure & Functionality of the QuoteProxy Class

- QuoteProxy uses an auto-generated Retrofit API class to abstracts details of making remote method invocations using HTTP

@Component

```
public class Components {
```

```
    @Bean
```

```
    public QuoteAPI getQuoteAPI() {
```

```
        return new Retrofit
```

```
            .Builder()
```

```
            .baseUrl(GATEWAY_BASE_URL)
```

```
            .addConverterFactory(GsonConverterFactory.create())
```

```
            .build()
```

```
            .create(QuoteAPI.class);
```

```
    }
```

```
}
```



Create a QuoteAPI Retrofit client that's used to make HTTP requests to the GatewayApplication microservice

The Structure & Functionality of the QuoteProxy Class

- QuoteProxy uses an auto-generated Retrofit API class to abstracts details of making remote method invocations using HTTP

@Component

```
public class QuoteProxy { ...  
    public List<Quote> postQuotes  
        (String route, (List<Integer> quoteIds, Boolean parallel) {  
        var response = mQuoteAPI  
            .postQuotes(route, quoteIds, parallel)  
            .execute();  
  
        if (response.isSuccessful())  
            return response.body();  
        else  
            ...  
    }  
}
```

This proxy method shields clients from low-level HTTP programming details

The Structure & Functionality of the QuoteProxy Class

- QuoteProxy uses an auto-generated Retrofit API class to abstracts details of making remote method invocations using HTTP

@Component

```
public class QuoteProxy { ...
    public List<Quote> postQuotes
        (String route, (List<Integer> quoteIds, Boolean parallel){
    var response = mQuoteAPI
        .postQuotes(route, quoteIds, parallel)
        .execute();

    if (response.isSuccessful())
        return response.body();
    else
        ...
}
```

Execute the postQuotes() Retrofit API remote method invocation

e.g., <http://localhost:8080/handey/quotes?parallel=true>

The Structure & Functionality of the QuoteProxy Class

- QuoteProxy uses an auto-generated Retrofit API class to abstracts details of making remote method invocations using HTTP

@Component

```
public class QuoteProxy { ...
```

```
    public List<Quote> postQuotes
```

```
        (String route, (List<Integer> quoteIds, Boolean parallel){
```

```
        var response = mQuoteAPI
```

```
            .postQuotes(route, quoteIds, parallel)
```

```
            .execute();
```

```
        if (response.isSuccessful())
```

```
            return response.body();
```

```
        else
```

```
            ...
```

```
    }
```

If the remote method invocation succeeded return the response body

End of the QuoteServices App Case Study: Structure & Functionality of the Client