

The QuoteServices App Case Study: Structure & Functionality of Client Classes

Douglas C. Schmidt

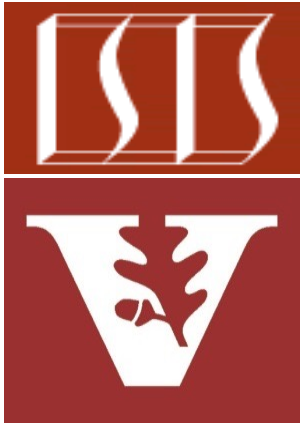
d.schmidt@vanderbilt.edu

www.dre.vanderbilt.edu/~schmidt

Professor of Computer Science

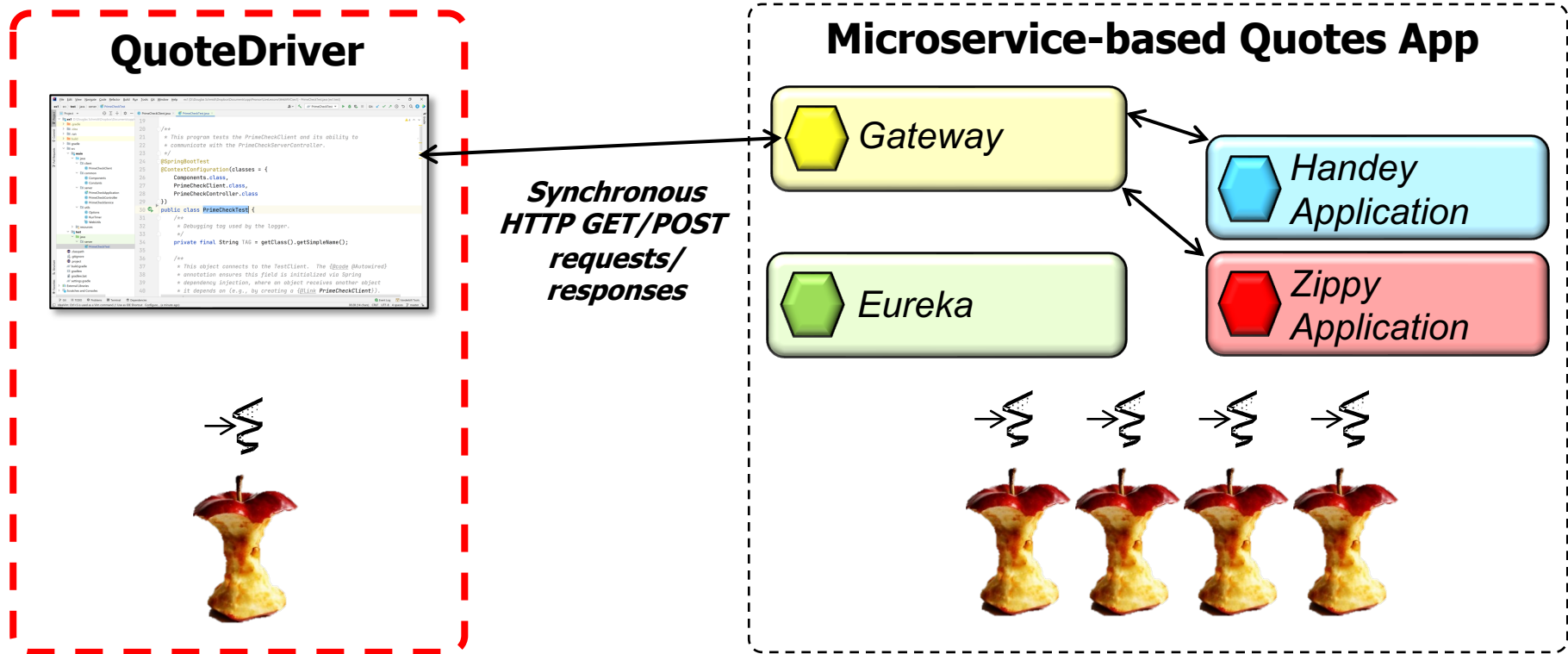
**Institute for Software
Integrated Systems**

**Vanderbilt University
Nashville, Tennessee, USA**



Learning Objectives in this Part of the Lesson

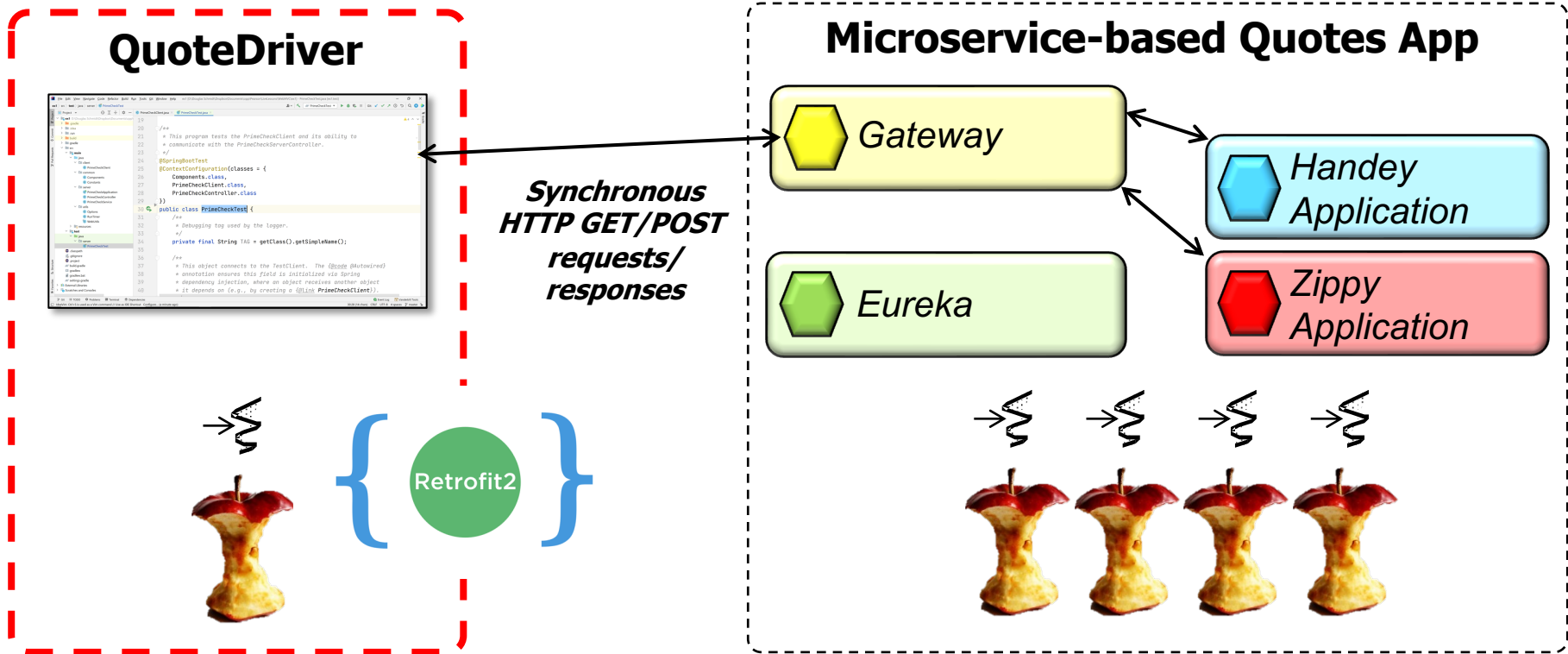
- Understand the structure & functionality of client-related classes that send/receive HTTP GET/POST requests/responses to/from the API Gateway



See github.com/douglasraigschmidt/LiveLessons/tree/master/WebMVC/ex4

Learning Objectives in this Part of the Lesson

- Understand the structure & functionality of client-related classes that send/receive HTTP GET/POST requests/responses to/from the API Gateway



Retrofit is applied to create Java proxy methods for the Gateway HTTP APIs

The Structure & Functionality of QuoteClient Classes

The Structure & Functionality of QuoteClient Classes

- The QuoteClient class performs synchronous remote method invocations on the API Gateway 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); }...
```

QuoteClient		
f	mQuoteProxy	QuoteProxy
m	getAllQuotes(String)	List<Quote>
m	getQuote(String, Integer)	Quote
m	postQuotes(String, List<Integer>, Boolean)	List<Quote>
m	searchQuotes(String, List<String>, Boolean)	List<Quote>
m	searchQuotesEx(String, List<String>, Boolean)	List<Quote>

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

The Structure & Functionality of QuoteClient Classes

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

@Component

```
public class QuoteClient
    @Autowired QuoteProxy mQuoteProxy;
```

Enable 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 class performs synchronous remote method invocations on the API Gateway 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); }
```

Spring's dependency injection framework auto-wires this field

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

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

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

See www.baeldung.com/spring-annotation

The Structure & Functionality of QuoteClient Classes

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

```
@Component
```

```
public class QuoteClient
```

```
    @Autowired QuoteProxy mQuoteProxy;
```

*Get List of Quotes via
their quote Ids from the
designated microservice*

```
    ...
```

```
    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); }...
```


The Structure & Functionality of QuoteClient Classes

- The QuoteClient class performs synchronous remote method invocations on the API Gateway 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); }...
```

Get List of Quotes by searching for them in the designated microservice

The Structure & Functionality of QuoteClient Classes

- The QuoteClient class performs synchronous remote method invocations on the API Gateway 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); }
```

The QuoteClient methods forward to the QuoteProxy methods, which do the work

```
    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 QuoteAPI Interface

- QuoteProxy uses Retrofit to turn an HTTP API into the QuoteAPI interface that shields client programmers from details of invoking remote methods via HTTP



QuoteAPI		
(m)	<code>getAllQuotes(String)</code>	<code>Call<List<Quote>></code>
(m)	<code>getQuote(String, Integer)</code>	<code>Call<Quote></code>
(m)	<code>postQuotes(String, List<Integer>, Boolean)</code>	<code>Call<List<Quote>></code>
(m)	<code>search(String, List<String>, Boolean)</code>	<code>Call<List<Quote>></code>
(m)	<code>searchEx(String, List<String>, Boolean)</code>	<code>Call<List<Quote>></code>

See square.github.io/retrofit

The Structure & Functionality of the QuoteAPI Interface

- QuoteProxy uses Retrofit to turn an HTTP API into the QuoteAPI interface that shields client programmers from details of invoking remote methods via HTTP

```
public interface QuoteAPI {
    @GET(SERVICE_PREFIX + "/" + GET_ALL_QUOTES)
    Call<List<Quote>> getAllQuotes(@Path(SERVICE) String route);

    @POST(SERVICE_PREFIX + "/" + POST_QUOTES)
    Call<List<Quote>> postQuotes(@Path(SERVICE) String route,
                                @Body List<Integer> quoteIds,
                                @Query(PARALLEL) Boolean parallel);

    @POST(SERVICE_PREFIX + "/" + POST_SEARCHES)
    Call<List<Quote>> search(@Path(SERVICE) 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 QuoteAPI Interface

- QuoteProxy uses Retrofit to turn an HTTP API into the QuoteAPI interface that shields client programmers from details of invoking remote methods via HTTP

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

These proxy methods mimic the signature of controller methods

The Structure & Functionality of the QuoteAPI Interface

- QuoteProxy uses Retrofit to turn an HTTP API into the QuoteAPI interface that shields client programmers from details of invoking remote methods via HTTP

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

Retrofit annotations are similar to Spring controller annotations

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

The Structure & Functionality of the QuoteAPI Interface

- QuoteProxy uses Retrofit to turn an HTTP API into the QuoteAPI interface that shields client programmers from details of invoking remote methods via HTTP

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

All proxy method return Call objects that can be executed to interact with the Gateway

See square.github.io/retrofit/2.x/retrofit/retrofit2/Call.html


The Structure & Functionality of the QuoteAPI Interface

- QuoteProxy uses Retrofit to turn an HTTP API into the QuoteAPI interface that shields client programmers from details of invoking remote methods via HTTP

ChatGPT 4 ▾

ChatGPT can generate QuoteAPI automatically via the BaseController

You

 **BaseController.java**
File

Please analyze the attached Java file containing a Spring BaseController and generate the corresponding Retrofit 2 interface named QuoteAPI that can be used by a client to invoke remote method calls on this controller.

See youtu.be/UzJoqoUOH3Q

The Structure & Functionality of the QuoteProxy Class

The Structure & Functionality of the QuoteProxy Class

- QuoteProxy uses auto-generated Retrofit QuoteAPI interface to shield client programmers from details of making remote method invocations using HTTP

@Component

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

QuoteProxy	
f	mQuoteAPI QuoteAPI
m	getAllQuotes(String) List<Quote>
m	getQuote(String, Integer) Quote
m	postQuotes(String, List<Integer>, Boolean) List<Quote>
m	search(String, List<String>, Boolean) List<Quote>
m	searchEx(String, List<String>, Boolean) List<Quote>

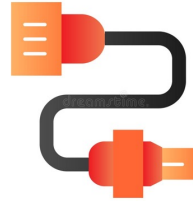
See [WebMVC/ex4/client/src/main/java/edu/vandy/quoteservices/client/QuoteProxy.java](https://github.com/vandymcquinn/WebMVC/ex4/client/src/main/java/edu/vandy/quoteservices/client/QuoteProxy.java)

The Structure & Functionality of the QuoteProxy Class

- QuoteProxy uses auto-generated Retrofit QuoteAPI interface to shield client programmers from 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

The Structure & Functionality of the QuoteProxy Class

- QuoteProxy uses auto-generated Retrofit QuoteAPI interface to shield client programmers from details of making remote method invocations using HTTP

@Component

```
public class ClientBeans {
```

```
    @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/ClientBeans.java

The Structure & Functionality of the QuoteProxy Class

- QuoteProxy uses auto-generated Retrofit QuoteAPI interface to shield client programmers from details of making remote method invocations using HTTP

@Component

```
public class ClientBeans {  
    @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 API Gateway

See [retrofit/2.x/converter-gson/retrofit2/converter/gson/GsonConverterFactory.html](https://retrofit2.github.io/retrofit2/converter/gson/GsonConverterFactory.html)

The Structure & Functionality of the QuoteProxy Class

- QuoteProxy uses auto-generated Retrofit QuoteAPI interface to shield client programmers from details of making remote method invocations using HTTP

@Component

```
public class QuoteProxy { ...  
    public List<Quote> postQuotes  
        (String route, List<Integer> quoteIds, Boolean parallel) {  
        return CallUtils  
            .executeCall(mQuoteAPI  
                .postQuotes(route,  
                    quoteIds,  
                    parallel));  
    }  
}
```

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

The Structure & Functionality of the QuoteProxy Class

- QuoteProxy uses auto-generated Retrofit QuoteAPI interface to shield client programmers from details of making remote method invocations using HTTP

```
public class QuoteProxy { ...
    public List<Quote> postQuotes
        (String route, List<Integer> quoteIds, Boolean parallel) {
    return CallUtils
        .executeCall(mQuoteAPI
            .postQuotes(route,
                quoteIds,
                parallel));
    }
```

These params will be encoded & passed to the API Gateway

The Structure & Functionality of the QuoteProxy Class

- QuoteProxy uses auto-generated Retrofit QuoteAPI interface to shield client programmers from details of making remote method invocations using HTTP

@Component

```
public class QuoteProxy { ...
    public List<Quote> postQuotes
        (String route, List<Integer> quoteIds, Boolean parallel) {
    return CallUtils
        .executeCall (mQuoteAPI
                        .postQuotes (route,
                                    quoteIds,
                                    parallel));
}
}
```

Invoke the postQuotes() Retrofit API remote method invocation, which returns a Call object

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

The Structure & Functionality of the QuoteProxy Class

- QuoteProxy uses auto-generated Retrofit QuoteAPI interface to shield client programmers from details of making remote method invocations using HTTP

@Component

```
public class QuoteProxy { ...
    public List<Quote> postQuotes
        (String route, List<Integer> quoteIds, Boolean parallel) {
    return CallUtils
        .executeCall (mQuoteAPI
            .postQuotes (route,
                quoteIds,
                parallel));
    }
```

Execute the Call & return the List of Quote objects received from the server on success or throw an IOException on failure

The Structure & Functionality of the CallUtils Class

The Structure & Functionality of the CallUtils Class

- The CallUtils Java utility class defines a method that is useful in conjunction with Retrofit

```
public class CallUtils { ...
    public static <T> T executeCall(Call<T> call) {
        return rethrowSupplier(() -> {
            Response<T> response = call.execute();
            if (response.isSuccessful())
                return response.body();
            else {
                int statusCode = response.code();
                String errorMessage = response.errorBody().string();
                throw new IOException(errorMessage);
            }
        }).get();
    } ...
}
```

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

The Structure & Functionality of the CallUtils Class

- The CallUtils Java utility class defines a method that is useful in conjunction with Retrofit

```
public class CallUtils { ...
    public static <T> T executeCall(Call<T> call) {
        return rethrowSupplier(() -> {
            Response<T> response = call.execute();
            if (response.isSuccessful())
                return response.body();
            else {
                int statusCode = response.code();
                String errorMessage = response.errorBody().string();
                throw new IOException(errorMessage);
            }
        }).get();
    } ...
}
```

*Factor common code to invoke
a Call & handle the response*

The Structure & Functionality of the CallUtils Class

- The CallUtils Java utility class defines a method that is useful in conjunction with Retrofit

```
public class CallUtils { ...
    public static <T> T executeCall(Call<T> call) {
        return rethrowSupplier(() -> {
            Response<T> response = call.execute();
            if (response.isSuccessful())
                return response.body();
            else {
                int statusCode = response.code();
                String errorMessage = response.errorBody().string();
                throw new IOException(errorMessage);
            }
        }).get();
    } ...
}
```

Invoke the call via Retrofit

The Structure & Functionality of the CallUtils Class

- The CallUtils Java utility class defines a method that is useful in conjunction with Retrofit

```
public class CallUtils { ...
    public static <T> T executeCall(Call<T> call) {
        return rethrowSupplier(() -> {
            Response<T> response = call.execute();
            if (response.isSuccessful())
                return response.body();
            else {
                int statusCode = response.code();
                String errorMessage = response.errorBody().string();
                throw new IOException(errorMessage);
            }
        }).get();
    } ...
}
```

Return the result on success

The Structure & Functionality of the CallUtils Class

- The CallUtils Java utility class defines a method that is useful in conjunction with Retrofit

```
public class CallUtils { ...
    public static <T> T executeCall(Call<T> call) {
        return rethrowSupplier(() -> {
            Response<T> response = call.execute();
            if (response.isSuccessful())
                return response.body();
            else {
                int statusCode = response.code();
                String errorMessage = response.errorBody().string();
                throw new IOException(errorMessage);
            }
        }).get();
    } ...
}
```

Throw an IOException on failure

The Structure & Functionality of the CallUtils Class

- The CallUtils Java utility class defines a method that is useful in conjunction with Retrofit

```
public class CallUtils { ...
    public static <T> T executeCall(Call<T> call) {
        return rethrowSupplier() -> {
            Response<T> response = call.execute();
            if (response.isSuccessful())
                return response.body();
            else {
                int statusCode = response.code();
                String errorMessage = response.errorBody().string();
                throw new IOException(errorMessage);
            }
        }
    }
} ...
```

Handle checked exceptions cleanly

See stackoverflow.com/a/27644392

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