


Web API Design with Spring Boot Week 1 Coding Assignment

Points possible: 70

Category	Criteria	% of Grade
Functionality	Does the code work?	25
Organization	Is the code clean and organized? Proper use of white space, syntax, and consistency are utilized. Names and comments are concise and clear.	25
Creativity	Student solved the problems presented in the assignment using creativity and out of the box thinking.	25
Completeness	All requirements of the assignment are complete.	25

Instructions: In Eclipse, or an IDE of your choice, write the code that accomplishes the objectives listed below. Ensure that the code compiles and runs as directed. Take screenshots of the code and of the running program (make sure to get screenshots of all required functionality) and paste them in this document where instructed below. Create a new repository on GitHub for this week's assignments and push this document, with your Java project code, to the repository. Add the URL for this week's repository to this document where instructed and submit this document to your instructor when complete.

Here's a friendly tip: as you watch the videos, code along with the videos. This will help you with the homework. When a screenshot is required, look for the icon:  You will keep adding to this project throughout this part of the course. When it comes time for the final project, use this project as a starter.

Here's a hint: make sure you are running a version of Java that is 11+. To get the version, open a Windows command window or a Mac Terminal window and type `java -version`. If you need to upgrade, go here: <https://docs.aws.amazon.com/corretto/latest/corretto-11-ug/downloads-list.html>. Pick the .msi installer version (Windows) or the .pkg version (Mac).

Project Resources: <https://github.com/promineotech/Spring-Boot-Course-Student-Resources>

Coding Steps:

- 1) Create a Maven project named `JeepSales` as described in the video.

- a) In Spring Tool Suite, click the "File" menu. Select "New/Project...". In the popup, expand "Maven" and select "Maven Project". Click "Next".
- b) Check "Create a simple project (skip archetype selection)". Click "Next".
- c) Enter the following:

Group Id	com.promineotech
Artifact Id	jeep-sales

Click "Finish".

- 2) Navigate to the Spring Initializr (<https://start.spring.io/>).

- a) Confirm the following settings:

Project	Maven Project
Language	Java
Spring Boot	Select the latest stable version (not SNAPSHOT or RC)
Group	com.promineotech
Artifact	jeep-sales
Name	jeep-sales
Description	Jeep Sales
Package name	com.promineotech
Packaging	Jar
Java	11

- b) Add the dependencies from the Initializr:
 - i) Web
 - ii) Devtools
 - iii) Lombok
- c) Click "Explore" at the bottom of the page.
- d) Click "Copy" to copy the pom.xml generated by the Initializr to the clipboard.

- 3) In Spring Tool Suite, open pom.xml (in the project root directory). Select all the text in the editor and replace it with the XML copied to the clipboard in the prior step.
- 4) Navigate to <https://mvnrepository.com/>. Search for springdoc-openapi-ui. Select the latest version and add the entry to the POM file in the <dependencies> section.
- 5) Create a package in src/main/java named com.promineotech.jeep. In this package:
 - a) Create a Java class with a main method named JeepSales.
 - b) Add a class-level annotation: @SpringBootApplication and the import statement.
 - c) In the main() method, add a call to SpringApplication.run();. Use JeepSales.class as the first parameter, and the args parameter that was passed into the main() method as the second. The entire class should look like this:

```
package com.promineotech.jeep;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class JeepSales {

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

- 6) Refer to README.docx in the supplied project resources. Copy all files in the Files folder in the resources to your project as described in the README. **Do not copy the files in the Entity or Source folders at this time.**
 - a) Load the files that were added: right-click on the project in Package Explorer and select "Refresh".
 - b) Update the project with the new POM dependencies: right-click on the project in Package Explorer, select "Maven/Update Project". When the "Update Maven Project" panel appears, click "OK".
- 7) Using the MySQL Workbench or MySQL command line client (CLI), create a database named "jeep".
- 8) Using dBeaver, or the MySQL client of choice, load the supplied .sql files (v1.0__Jeep_Schema.sql, and v1.1__Jeep_Data.sql) into the MySQL database to create the tables and populate them with data. These files are found in the project folder src/test/resources/flyway/migrations.
- 9) Create a new package in src/test/java named com.promineotech.jeep.controller. Create a Spring Boot integration test named FetchJeepTest using the techniques shown in the video.

- a) Add the `@SpringBootTest`, `@ActiveProfiles`, and `@Sql` annotations as described in the video.
- b) The class must not be public. It should have package-level access (i.e., not public, private, or protected).
- c) The video extended `FetchJeepTestSupport`, but you don't need to do that for the homework. Just put everything in `FetchJeepTest`. It should look like this:

```
@SpringBootTest(webEnvironment = WebEnvironment.RANDOM_PORT)
@ActiveProfiles("test")
@Sql(scripts = {
    "classpath:flyway/migrations/V1.0__Jeep_Schema.sql",
    "classpath:flyway/migrations/V1.1__Jeep_Data.sql"},
    config = @SqlConfig(encoding = "utf-8"))
class FetchJeepTest {
}
```

- d) Create a test method in `FetchJeepTest`. The method must have the following method signature:

```
void testThatJeepsAreReturnedWhenAValidModelAndTrimAreSupplied()
```

- e) Inject a `TestRestTemplate` in the test class. Name the variable `restTemplate`. Inject the port used in the test using the `@LocalServerPort` annotation. Name the variable `serverPort`. The variables and annotations should look like this:

```
@Autowired
private TestRestTemplate restTemplate;

@LocalServerPort
private int serverPort;
```

- 10) Create a new package in `src/main/java` named `com.promineotech.jeep.entity`. In that package, create an enum named `JeepModel`. Add all the jeep models from the `model_id` column in the `models` table in the database. You can use this query in dBeaver: `SELECT DISTINCT model_id FROM models`.

- 11) Create a `Jeep` class in the `com.promineotech.jeep.entity` package. Add the columns from the `models` table into this class as instance variables. Annotate the class with the Lombok annotations `@Data`, `@Builder` (and optionally both `@NoArgsConstructor` and `@AllArgsConstructor`). Note that `modelId` should be of type `JeepModel` and `basePrice` should be of type `BigDecimal`. The class should look like this (remember to add the appropriate import statements):

```
@Data
@Builder
@NoArgsConstructor
@AllArgsConstructor
public class Jeep {
    private Long modelPK;
```

```

private JeepModel modelId;
private String trimLevel;
private int numDoors;
private int wheelSize;
private BigDecimal basePrice;
}

```

- 12) In the supplied resources, copy all files in the Entities folder to the `src/main/java/com/promineotech/jeep/entity` folder. **Do not copy anything from the Source folder at this time.**
- 13) Back in the test method that you were writing, create local variables for `JeepModel`, `trim`, and `uri`. Set them appropriately like this:

Variable Type	Variable Name	Variable Value
JeepModel	<code>model</code>	<code>JeepModel.WRANGLER</code>
String	<code>trim</code>	<code>"Sport"</code>
String	<code>uri</code>	<code>String.format("http://localhost:%d/jeeps?model=%s&trim=%s", serverPort, model, trim);</code>

- a) Send an HTTP request to the REST service that passes a `JeepModel` and trim level as URI parameters (as shown in the video). Use this method call:

```

ResponseEntity<List<Jeep>> response = restTemplate.exchange(uri,
    HttpMethod.GET, null, new ParameterizedTypeReference<>() {});

```


Make sure to use the import `java.util.List` and `org.springframework.http.HttpMethod`.

- b) Using [AssertJ](#), test that the response that comes back from the server is 200 (success) – or as is shown in the video: `HttpStatus.OK`. The code should look like this:

```
assertThat(response.getStatusCode()).isEqualTo(HttpStatus.OK);
```

Use the import statements:

```
import static org.assertj.core.api.Assertions.assertThat;
```

- c) Produce a screenshot showing the completed test class. 

- 14) In `src/main/java`, create a new package `com.promineotech.jeep.controller`. In this package, create an interface named `JeepSalesController`.

- a) Add the class-level annotation `@RequestMapping("/jeeps")`.
- b) Add the `fetchJeeps` method in a controller interface with the following signature:


```
List<Jeep> fetchJeeps(JeepModel model, String trim);
```

Make sure you use the List from java.util.List.

- c) Add OpenAPI documentation to document the four possible outcomes: 200 (success), 400 (bad input), 404 (not found) and 500 (unplanned error) as shown in the video.
- d) Add the parameter annotations in the OpenAPI documentation to describe the model and trim parameters.
- e) Add the @GetMapping annotation and the @ResponseStatus(code = HttpStatus.OK) annotation as method-level annotations to the fetchJeeps method.
- f) Add the @RequestParam annotations to the parameters as described in the video. The interface should look like this (omitting the OpenAPI annotations):

```
@RequestMapping("/jeeps")
public interface JeepSalesController {
    @GetMapping
    @ResponseStatus(code = HttpStatus.OK)
    List<Jeep> fetchJeeps(@RequestParam JeepModel model,
        @RequestParam String trim);
}
```

- g) Produce a screenshot showing the interface and OpenAPI documentation. 

- 15) Add the controller implementation class named DefaultJeepSalesController. Don't forget the @RestController annotation.
- 16) Run the application within the IDE and show the resulting OpenAPI (Swagger) documentation produced in the browser. Produce a screenshot of the documentation showing all four possible outcomes. 

Screenshots of Code:

FetchJeepTest.java FetchJeepTestSup... BaseTest.java JeepModel.java Jeep.java JeepSales.java jeep-sales/pom.x

```
1 package com.promineotech.jeeptest.controller;
2
3 import java.util.List;
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20 @RequestMapping("/jeeps")
21 // @OpenAPIDefinition(info = @Info(title = "jeep Sales Service"),
22 //     servers = {@Server(url = "http://localhost: 8080", description = "Local server.")})
23
24 public interface JeepSalesController {
25     // @formatter: off
26     @Operation(
27         summary = "Returns a list of Jeeps",
28         description = "Returns a list of Jeeps given an optional model and/or trim",
29         responses = {
30             @ApiResponse(
31                 responseCode = "200",
32                 description = "A list of Jeeps is returned",
33                 content = @Content(
34                     mediaType = "application/json",
35                     schema = @Schema(implementation = Jeep.class) ),
36             @ApiResponse(
37                 responseCode = "400",
38                 description = "The request parameters are invalid",
39                 content = @Content(mediaType = "application/json"),
40             @ApiResponse(
41                 responseCode = "404",
42                 description = "No Jeeps were found with the input criteria",
43                 content = @Content(mediaType = "application/json"),
44             @ApiResponse(
45                 responseCode = "500",
46                 description = "An unplanned error occurred",
47                 content = @Content(mediaType = "application/json"))
48         },
49     ),
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
```


*FetchJeepTest.java × FetchJeepTestSup... *BaseTest.java JeepModel.java *Jeep.java JeepSales.java jeep-sales/pom....

```
1 package com.promineotech.jeeptest.controller;
2
3 import static org.assertj.core.api.Assertions.assertThat;
4 import java.util.List;
5 import org.junit.jupiter.api.Test;
6 import org.springframework.beans.factory.annotation.Autowired;
7 import org.springframework.boot.test.context.SpringBootTest;
8 import org.springframework.boot.test.context.SpringBootTest.WebEnvironment;
9 import org.springframework.boot.test.web.client.TestRestTemplate;
10 import org.springframework.boot.test.web.server.LocalServerPort;
11 import org.springframework.core.ParameterizedTypeReference;
12 import org.springframework.http.HttpMethod;
13 import org.springframework.http.HttpStatus;
14 import org.springframework.http.ResponseEntity;
15 import org.springframework.test.context.ActiveProfiles;
16 import org.springframework.test.context.jdbc.Sql;
17 import org.springframework.test.context.jdbc.SqlConfig;
18 import com.promineotech.jeeptest.entity.Jeep;
19 import com.promineotech.jeeptest.entity.JeepModel;
20
21
22 @SpringBootTest(webEnvironment = WebEnvironment.RANDOM_PORT)
23 @ActiveProfiles("test")
24 @Sql(scripts = {
25     "classpath:flyway/migrations/v1.0 Jeep Schema.sql",
26     "classpath:flyway/migrations/V1.1 Jeep Data.sql"},
27     config = @SqlConfig(encoding = "utf-8"))
28 class FetchJeepTest { //extends FetchJeepTestSupport {
29     @Autowired
30     private TestRestTemplate restTemplate ;
31
32     @LocalServerPort
33     private int serverPort;
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
```



```
*FetchJeepTest.java  FetchJeepTestSup...  *BaseTest.java  JeepModel.java  *Jeep.java  JeepSales.java
1 package com.promineotech.jeep.controller.support;
2
3 public class FetchJeepTestSupport extends BaseTest {
4
5 }
6
```

```
*FetchJeepTest.java  FetchJeepTestSup...  *BaseTest.java  JeepModel.java  *Jeep.java  JeepSales.java  jeep-sa
1 package com.promineotech.jeep.controller.support;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4 import org.springframework.boot.test.web.client.TestRestTemplate;
5 import org.springframework.boot.test.web.server.LocalServerPort;
6 import lombok.Getter;
7
8 public class BaseTest {
9     @LocalServerPort
10     private int serverPort;
11
12     @Autowired
13     @Getter
14     private TestRestTemplate restTemplate;
15
16     protected String getBaseUri() {
17         return String.format("http://localhost:%d/jeeps", serverPort);
18     }
19
20 }
21
```

*FetchJeepTest.java FetchJeepTestSup... *BaseTest.java JeepModel.java × *Jeep.java JeepSales.java jeep-s

```
1 package com.promineotech.jeep.entity;
2
3 public enum JeepModel {
4     WRANGLER, GRAND_CHEROKEE, CHEROKEE, COMPASS, RENEGADE, GLADIATOR, WRANGLER_4XE
5 }
6
```

*FetchJeepTest.java FetchJeepTestSup... *BaseTest.java JeepModel.java *Jeep.java × JeepSales.java jeep-s

```
1 package com.promineotech.jeep.entity;
2
3 import java.math.BigDecimal;
4 import lombok.AllArgsConstructor;
5 import lombok.Builder;
6 import lombok.Data;
7 import lombok.NoArgsConstructor;
8
9 @Data
10 @Builder
11 @NoArgsConstructor
12 @AllArgsConstructor
13 public class Jeep {
14     private Long modelOK;
15     private JeepModel modelId;
16     private String trimLevel;
17     private int wheelSize;
18     private BigDecimal basePrice;
19
20 }
21
```

```

*FetchJeepTest.java  FetchJeepTestSup...  *BaseTest.java  JeepModel.java  *Jeep.java  JeepSales.java X
1 package com.promineotech.jeep;
2
3 import org.springframework.boot.autoconfigure.SpringBootApplication;
4 import org.springframework.boot.CommandLineRunner;
5 import org.springframework.boot.SpringApplication;
6
7 @SpringBootApplication
8 public class JeepSales implements CommandLineRunner{
9
10 public static void main(String[] args) {
11     SpringApplication.run(JeepSales.class, args);
12
13
14 }
15
16
17 @Override
18 public void run(String... args) throws Exception {
19     // TODO Auto-generated method stub
20
21 }
22
23 }
24

```

```

*FetchJeepTest.java  FetchJeepTestSup...  *BaseTest.java  JeepModel.java  *Jeep.java  JeepSales.java  *j
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/
3 XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://
4 maven.apache.org/xsd/maven-4.0.0.xsd">
5     <modelVersion>4.0.0</modelVersion>
6     <parent>
7         <groupId>org.springframework.boot</groupId>
8         <artifactId>spring-boot-starter-parent</artifactId>
9         <version>2.7.3</version>
10        <relativePath /> <!-- lookup parent from repository -->
11    </parent>
12    <groupId>com.promineotech</groupId>
13    <artifactId>jeep-sales</artifactId>
14    <version>0.0.1-SNAPSHOT</version>
15    <name>jeep-sales</name>
16    <description>jeepsales</description>
17    <properties>
18        <java.version>17</java.version>
19    </properties>
20    <dependencies>
21        <dependency>
22            <groupId>org.springframework.boot</groupId>
23            <artifactId>spring-boot-starter-web</artifactId>
24        </dependency>
25
26        <dependency>
27            <groupId>org.springframework.boot</groupId>
28            <artifactId>spring-boot-devtools</artifactId>
29            <scope>runtime</scope>
30            <optional>true</optional>
31        </dependency>
32        <dependency>

```

```

33     <groupId>org.projectlombok</groupId>
34     <artifactId>lombok</artifactId>
35     <optional>true</optional>
36 </dependency>
37Ⓜ <dependency>
38     <groupId>org.springframework.boot</groupId>
39     <artifactId>spring-boot-starter-test</artifactId>
40     <scope>test</scope>
41 </dependency>
42
43
44 <!-- OpenAPI dependency=====-->
45Ⓜ <dependency>
46     <groupId>org.springdoc</groupId>
47     <artifactId>springdoc-openapi-ui</artifactId>
48     <version>1.6.11</version>
49 </dependency>
50
51
52 <!-- https://mvnrepository.com/artifact/org.springdoc/springdoc-openapi-ui -->
53
54Ⓜ <dependency>
55     <groupId>org.springdoc</groupId>
56     <artifactId>springdoc-openapi-ui</artifactId>
57     <version>1.6.11</version>
58 </dependency>
59
60 </dependencies>
61

```

```

61
62Ⓜ <build>
63Ⓜ   <plugins>
64Ⓜ     <plugin>
65         <groupId>org.springframework.boot</groupId>
66         <artifactId>spring-boot-maven-plugin</artifactId>
67Ⓜ     <configuration>
68Ⓜ       <excludes>
69Ⓜ         <exclude>
70             <groupId>org.projectlombok</groupId>
71             <artifactId>lombok</artifactId>
72         </exclude>
73       </excludes>
74     </configuration>
75   </plugin>
76 </plugins>
77 </build>
78
79 </project>

```

```

*FetchJeepTest.java  FetchJeepTestSup...  *BaseTest.java  JeepModel.java  *Jeep.java  JeepSales.java
1  package com.promineotech.jeep.controller;
2
3  import java.util.List;
4  import org.springframework.http.HttpStatus;
5  import org.springframework.web.bind.annotation.GetMapping;
6  import org.springframework.web.bind.annotation.RequestMapping;
7  import org.springframework.web.bind.annotation.RequestParam;
8  import org.springframework.web.bind.annotation.ResponseStatus;
9  import com.promineotech.jeep.entity.Jeep;
10 import com.promineotech.jeep.entity.JeepModel;
11 import io.swagger.v3.oas.annotations.Operation;
12 import io.swagger.v3.oas.annotations.Parameter;
13 import io.swagger.v3.oas.annotations.media.Content;
14 import io.swagger.v3.oas.annotations.media.Schema;
15 import io.swagger.v3.oas.annotations.responses.ApiResponse;
16
17 @RequestMapping("/jeeps")
18 // @OpenAPIDefinition(info = @Info(title = "jeep Sales Service"),
19 //     servers = {@Server(url = "http://localhost: 8080", description = "Local server.")})
20
21 public interface JeepSalesController {
22     // @formatter: off
23     @Operation(
24         summary = "Returns a list of Jeeps",
25         description = "Returns a list of Jeeps given an optional model and/or trim",
26         responses = {
27             @ApiResponse(
28                 responseCode = "200",
29                 description = "A list of Jeeps is returned",
30                 content = @Content(
31                     mediaType = "application/json",
32                     schema = @Schema(implementation = Jeep.class)),
33             @ApiResponse(
34                 responseCode = "400",

```

```

35         description = "The request parameters are invalid",
36         content = @Content(mediaType = "application/json")),
37     @ApiResponse(
38         responseCode = "404",
39         description = "No Jeeps were found with the input criteria",
40         content = @Content(mediaType = "application/json")),
41     @ApiResponse(
42         responseCode = "500",
43         description = "An unplanned error occurred",
44         content = @Content(mediaType = "application/json"))
45 },
46
47 parameters = {
48
49     @Parameter(
50         name = "model",
51         allowEmptyValue = false,
52         required = false,
53         description = "The model name (i.e., 'Wrangler')"),
54
55     @Parameter(
56         name = "trim",
57         allowEmptyValue = false,
58         required = false, description = "The trim level (i.e., 'sport')"))
59
60
61
62 @GetMapping
63 @ResponseStatus(code = HttpStatus.OK)
64 List<Jeep> fetchJeeps(
65     @RequestParam JeepModel model,
66     @RequestParam String trim);
67 // @formatter: on

```

Screenshots of Running Application:

workspace - jeep-sales/src/main/java/com/promineotech/jeep/JeepSales.java - Spring Tool Suite 4

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer

Finished after 2.65 seconds

Runs: 1/1 Errors: 1 Failures: 0

Failure Trace

org.springframework.web.client.RestClientException: Error while extracting response body: JSON parse error: Cannot deserialize value of type 'java.util.ArrayList' from token: deserialize value of type 'java.util.ArrayList' com.promineotech.jee...
at [Source: (org.springframework.util.StreamUtils\$NonClosingInputStream); line: 1, column: 1]
at org.springframework.web.client.HttpMessageConverterExtractor.extractData(Ht...
at org.springframework.web.client.RestTemplate\$ResponseEntityResponseExtractor...
at org.springframework.web.client.RestTemplate\$ResponseEntityResponseExtractor...
at org.springframework.web.client.RestTemplate.doExecute(RestTemplate.java:778)
at org.springframework.web.client.RestTemplate.execute(RestTemplate.java:711)
at org.springframework.web.client.RestTemplate.execute(RestTemplate.java:631)
at org.springframework.web.client.RestTemplate.execute(RestTemplate.java:631)

Boot Dashboard

Type tags, projects, or working set names to match (incl. * and ? wildcards)

jeep-sales - JeepSales (Spring Boot App) C:\Users\bmaj\STSuite-4\16.0.RELEASE\plugins\org.eclipse.jdt.launcher\hotspot\jre.full.win32.x86_64_17.0.4\20220903-1038\jre\bin\java.exe (Sep 23, 2022, 9:30:41 PM) [pid: 10804]

2022-09-23 21:30:44.778 INFO 10804 --- [restartedMain] com.promineotech.jee... : Starting JeepSales using Java 17.0.4.1 on LAPTOP-FFFC0P4E8 with PID 10804 (C...
2022-09-23 21:30:44.781 INFO 10804 --- [restartedMain] com.promineotech.jee... : No active profile set, falling back to default profile: 'default'
2022-09-23 21:30:44.807 INFO 10804 --- [restartedMain] o.s.devtools.properties.DefaultPropertySourceLocator : DevTools property defaults active! Set 'spring.devtools.add-properties' to '...
2022-09-23 21:30:44.807 INFO 10804 --- [restartedMain] o.s.devtools.properties.DefaultPropertySourceLocator : For additional web related logging consider setting the 'logging.level.web' to '...
2022-09-23 21:30:45.319 INFO 10804 --- [restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
2022-09-23 21:30:45.325 INFO 10804 --- [restartedMain] org.apache.catalina.core.StandardService : Starting service [Tomcat]
2022-09-23 21:30:45.328 INFO 10804 --- [restartedMain] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.65]
2022-09-23 21:30:45.368 INFO 10804 --- [restartedMain] o.s.c.o.c.TomcatLocalhost : Initializing Spring embedded WebApplicationContext
2022-09-23 21:30:45.368 INFO 10804 --- [restartedMain] w.s.s.s.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 560 ms
2022-09-23 21:30:45.744 INFO 10804 --- [restartedMain] o.s.b.d.a.OptionalLiveReloadServer : LiveReload server is running on port 35729
2022-09-23 21:30:45.745 INFO 10804 --- [restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''
2022-09-23 21:30:45.771 INFO 10804 --- [restartedMain] com.promineotech.jee... : Started JeepSales in 1.177 seconds (JVM running for 1.587)

```
FetchJeepTest.java X FetchJeepTestSup... BaseTest.java JeepModel.java Jeep.java JeepSales.java

1 package com.promineotech.jee.controller;
2
3 import static org.assertj.core.api.Assertions.assertThat;
4
5 import static org.junit.jupiter.api.Assertions.*;
6 import java.util.List;
7 import org.junit.jupiter.api.Test;
8 import org.springframework.beans.factory.annotation.Autowired;
9 import org.springframework.boot.test.context.SpringBootTest;
10 import org.springframework.boot.test.context.SpringBootTest.WebEnvironment;
11 import org.springframework.boot.test.web.client.TestRestTemplate;
12 import org.springframework.boot.test.web.server.LocalServerPort;
13 import org.springframework.core.ParameterizedTypeReference;
14 import org.springframework.http.HttpStatus;
15 import org.springframework.http.ResponseEntity;
16 import org.springframework.test.context.ActiveProfiles;
17 import org.springframework.test.context.jdbc.Sql;
18 import org.springframework.test.context.jdbc.SqlConfig;
19 import com.promineotech.jee.controller.support.FetchJeepTestSupport;
20 import com.promineotech.jee.entity.Jee;
21 import com.promineotech.jee.entity.JeeModel;
22 import org.springframework.http.HttpMethod;
23 import static org.assertj.core.api.Assertions.assertThat;
24
25 @SpringBootTest(webEnvironment = WebEnvironment.RANDOM_PORT)
26 @ActiveProfiles("test")
27 @Sql(scripts = {
28     "classpath:flyway/migrations/v1.0 Jeep Schema.sql",
29     "classpath:flyway/migrations/V1.1 Jeep Data.sql"},
30     config = @SqlConfig(encoding = "utf-8"))
31 class FetchJeepTest { //extends FetchJeepTestSupport
32     @Autowired
33     private TestRestTemplate restTemplate;
34
35     @LocalServerPort
36     private int serverPort;
37
```


URL to GitHub Repository:

<https://github.com/bmason1969/Week-13.git>