

Week 16 coding Assignment requirements and screenshots.

Screenshots of Code:

The screenshot shows an IDE window titled "Java - jeep-sales/src/test/java/com/promineotech/jeep/controller/CreateOrderTestSupport.java - Spring Tool Suite 4". The editor displays the following code:

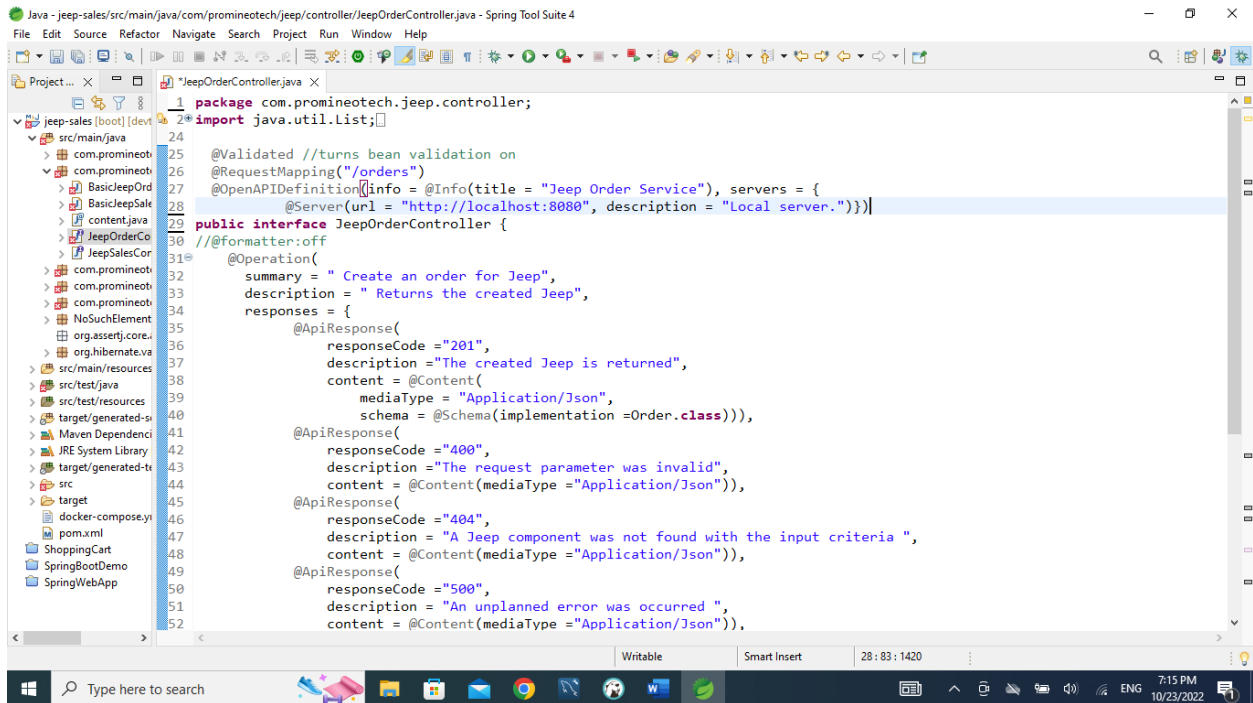
```
1 package com.promineotech.jeepp.controller;  
2  
3 public class CreateOrderTestSupport extends BaseTest {  
4     //formatter:off  
5  
6     /**  
7      *  
8      * @return  
9     */  
10    //return
```

The console output shows the following log messages:

```
<terminated> jeep-sales - JeepSales [Spring Boot App] C:\Program Files\Development\sts-4.16.0.RELEASE\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_17.0.4.v20220903-1038\jre\bin\javaw.exe  
20:05:20.626 [Thread-0] DEBUG org.springframework.boot.devtools.restart.classloader.RestartClassLoader - Created RestartCL  
  
:: Spring Boot :: (v2.7.4)  
  
2022-10-14 20:05:21.075 INFO 7600 --- [ restartedMain] com.promineotech.jeepp.JeeppSales : Starting JeepSales usi  
2022-10-14 20:05:21.077 INFO 7600 --- [ restartedMain] com.promineotech.jeepp.JeeppSales : No active profile set, fa  
2022-10-14 20:05:21.124 INFO 7600 --- [ restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : Devtools property defe  
2022-10-14 20:05:21.125 INFO 7600 --- [ restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : For additional web rel  
2022-10-14 20:05:21.868 INFO 7600 --- [ restartedMain] .s.d.r.c.RepositoryConfigurationDelegate : Multiple Spring Data m  
2022-10-14 20:05:21.870 INFO 7600 --- [ restartedMain] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data  
2022-10-14 20:05:21.885 INFO 7600 --- [ restartedMain] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data r  
2022-10-14 20:05:21.907 INFO 7600 --- [ restartedMain] .s.d.r.c.RepositoryConfigurationDelegate : Multiple Spring Data n  
2022-10-14 20:05:21.907 INFO 7600 --- [ restartedMain] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring C  
2022-10-14 20:05:21.910 INFO 7600 --- [ restartedMain] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data r
```

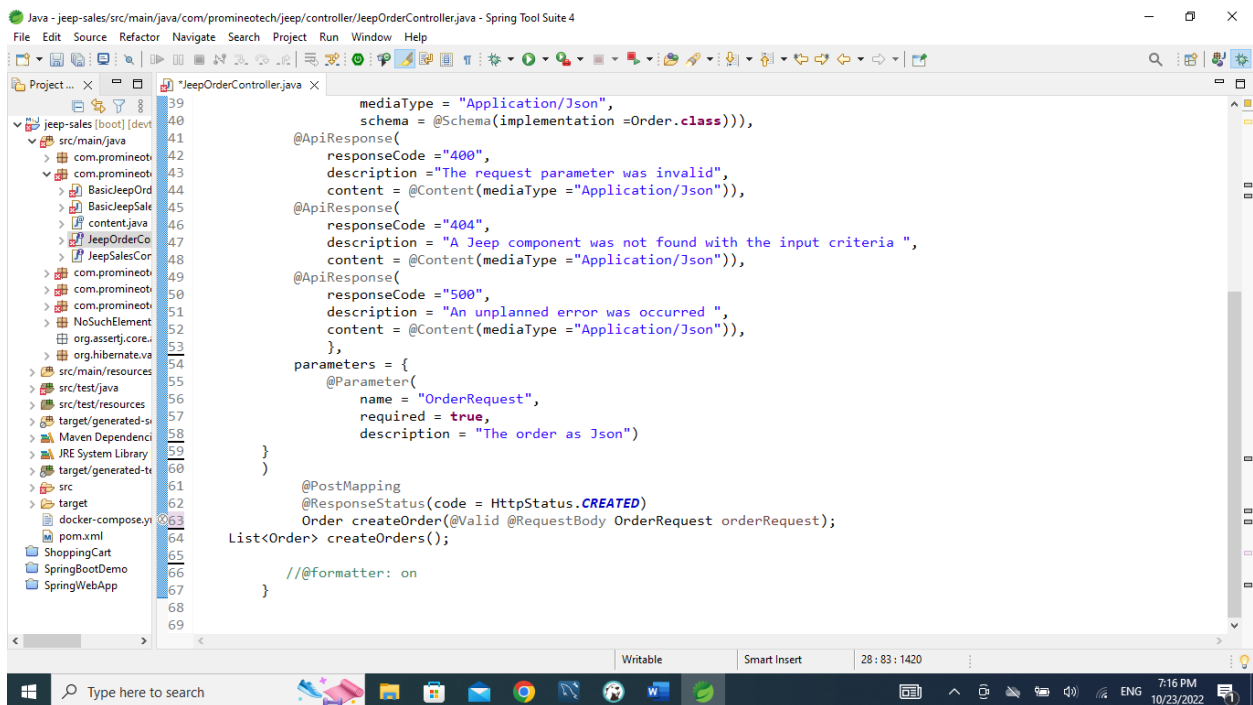
The screenshot shows an IDE window titled "Java - jeep-sales/src/main/java/com/promineotech/jeep/controller/JeepOrderController.java - Spring Tool Suite 4". The editor displays the following code:

```
24  
25 @Validated //turns bean validation on  
26 @RequestMapping("/orders")  
27 @OpenAPIDefinition(info = @Info(title = "Jeep Order Service"), servers = {  
28     @Server(url = "http://localhost:8080", description = "Local server.")})  
29 public interface JeepOrderController {  
30     //formatter:off  
31  
32     @Operation(  
33         summary = " Create an order for Jeep",  
34         description = " Returns the created Jeep",  
35         responses = {  
36             @ApiResponse(  
37                 responseCode = 200,  
38                 description = "Successful response",  
39                 content = {  
40                     @Content(mediaType = "application/json",  
41                         schema = @Schema(implementation = JeepOrder.class))  
42                 }  
43             )  
44         }  
45     )  
46     JeepOrder createJeepOrder(JeepOrder jeepOrder);  
47  
48     @Operation(  
49         summary = " Get an order for Jeep",  
50         description = " Returns the created Jeep",  
51         responses = {  
52             @ApiResponse(  
53                 responseCode = 200,  
54                 description = "Successful response",  
55                 content = {  
56                     @Content(mediaType = "application/json",  
57                         schema = @Schema(implementation = JeepOrder.class))  
58                 }  
59             )  
60         }  
61     )  
62     JeepOrder getJeepOrder(Long id);  
63  
64     @Operation(  
65         summary = " Delete an order for Jeep",  
66         description = " Returns the created Jeep",  
67         responses = {  
68             @ApiResponse(  
69                 responseCode = 200,  
70                 description = "Successful response",  
71                 content = {  
72                     @Content(mediaType = "application/json",  
73                         schema = @Schema(implementation = JeepOrder.class))  
74                 }  
75             )  
76         }  
77     )  
78     void deleteJeepOrder(Long id);  
79  
80     @Operation(  
81         summary = " Update an order for Jeep",  
82         description = " Returns the created Jeep",  
83         responses = {  
84             @ApiResponse(  
85                 responseCode = 200,  
86                 description = "Successful response",  
87                 content = {  
88                     @Content(mediaType = "application/json",  
89                         schema = @Schema(implementation = JeepOrder.class))  
90                 }  
91             )  
92         }  
93     )  
94     JeepOrder updateJeepOrder(Long id, JeepOrder jeepOrder);  
95  
96     @Operation(  
97         summary = " Get all orders for Jeep",  
98         description = " Returns the created Jeep",  
99         responses = {  
100        @ApiResponse(  
101            responseCode = 200,  
102            description = "Successful response",  
103            content = {  
104                @Content(mediaType = "application/json",  
105                    schema = @Schema(implementation = JeepOrder.class))  
106            }  
107        )  
108    }  
109    List<JeepOrder> getAllJeepOrders();  
110  
111     @Operation(  
112         summary = " Get an order for Jeep",  
113         description = " Returns the created Jeep",  
114         responses = {  
115             @ApiResponse(  
116                 responseCode = 200,  
117                 description = "Successful response",  
118                 content = {  
119                     @Content(mediaType = "application/json",  
120                         schema = @Schema(implementation = JeepOrder.class))  
121                 }  
122             )  
123         }  
124     )  
125     JeepOrder getJeepOrder(Long id);  
126  
127     @Operation(  
128         summary = " Delete an order for Jeep",  
129         description = " Returns the created Jeep",  
130         responses = {  
131             @ApiResponse(  
132                 responseCode = 200,  
133                 description = "Successful response",  
134                 content = {  
135                     @Content(mediaType = "application/json",  
136                         schema = @Schema(implementation = JeepOrder.class))  
137                 }  
138             )  
139         }  
140     )  
141     void deleteJeepOrder(Long id);  
142  
143     @Operation(  
144         summary = " Update an order for Jeep",  
145         description = " Returns the created Jeep",  
146         responses = {  
147             @ApiResponse(  
148                 responseCode = 200,  
149                 description = "Successful response",  
150                 content = {  
151                     @Content(mediaType = "application/json",  
152                         schema = @Schema(implementation = JeepOrder.class))  
153                 }  
154             )  
155         }  
156     )  
157     JeepOrder updateJeepOrder(Long id, JeepOrder jeepOrder);  
158  
159     @Operation(  
160         summary = " Get all orders for Jeep",  
161         description = " Returns the created Jeep",  
162         responses = {  
163             @ApiResponse(  
164                 responseCode = 200,  
165                 description = "Successful response",  
166                 content = {  
167                     @Content(mediaType = "application/json",  
168                         schema = @Schema(implementation = JeepOrder.class))  
169                 }  
170             )  
171         }  
172     )  
173     List<JeepOrder> getAllJeepOrders();  
174  
175     @Operation(  
176         summary = " Get an order for Jeep",  
177         description = " Returns the created Jeep",  
178         responses = {  
179             @ApiResponse(  
180                 responseCode = 200,  
181                 description = "Successful response",  
182                 content = {  
183                     @Content(mediaType = "application/json",  
184                         schema = @Schema(implementation = JeepOrder.class))  
185                 }  
186             )  
187         }  
188     )  
189     JeepOrder getJeepOrder(Long id);  
190  
191     @Operation(  
192         summary = " Delete an order for Jeep",  
193         description = " Returns the created Jeep",  
194         responses = {  
195             @ApiResponse(  
196                 responseCode = 200,  
197                 description = "Successful response",  
198                 content = {  
199                     @Content(mediaType = "application/json",  
200                         schema = @Schema(implementation = JeepOrder.class))  
201                 }  
202             )  
203         }  
204     )  
205     void deleteJeepOrder(Long id);  
206  
207     @Operation(  
208         summary = " Update an order for Jeep",  
209         description = " Returns the created Jeep",  
210         responses = {  
211             @ApiResponse(  
212                 responseCode = 200,  
213                 description = "Successful response",  
214                 content = {  
215                     @Content(mediaType = "application/json",  
216                         schema = @Schema(implementation = JeepOrder.class))  
217                 }  
218             )  
219         }  
220     )  
221     JeepOrder updateJeepOrder(Long id, JeepOrder jeepOrder);  
222  
223     @Operation(  
224         summary = " Get all orders for Jeep",  
225         description = " Returns the created Jeep",  
226         responses = {  
227             @ApiResponse(  
228                 responseCode = 200,  
229                 description = "Successful response",  
230                 content = {  
231                     @Content(mediaType = "application/json",  
232                         schema = @Schema(implementation = JeepOrder.class))  
233                 }  
234             )  
235         }  
236     )  
237     List<JeepOrder> getAllJeepOrders();  
238  
239     @Operation(  
240         summary = " Get an order for Jeep",  
241         description = " Returns the created Jeep",  
242         responses = {  
243             @ApiResponse(  
244                 responseCode = 200,  
245                 description = "Successful response",  
246                 content = {  
247                     @Content(mediaType = "application/json",  
248                         schema = @Schema(implementation = JeepOrder.class))  
249                 }  
250             )  
251         }  
252     )  
253     JeepOrder getJeepOrder(Long id);  
254  
255     @Operation(  
256         summary = " Delete an order for Jeep",  
257         description = " Returns the created Jeep",  
258         responses = {  
259             @ApiResponse(  
260                 responseCode = 200,  
261                 description = "Successful response",  
262                 content = {  
263                     @Content(mediaType = "application/json",  
264                         schema = @Schema(implementation = JeepOrder.class))  
265                 }  
266             )  
267         }  
268     )  
269     void deleteJeepOrder(Long id);  
270  
271     @Operation(  
272         summary = " Update an order for Jeep",  
273         description = " Returns the created Jeep",  
274         responses = {  
275             @ApiResponse(  
276                 responseCode = 200,  
277                 description = "Successful response",  
278                 content = {  
279                     @Content(mediaType = "application/json",  
280                         schema = @Schema(implementation = JeepOrder.class))  
281                 }  
282             )  
283         }  
284     )  
285     JeepOrder updateJeepOrder(Long id, JeepOrder jeepOrder);  
286  
287     @Operation(  
288         summary = " Get all orders for Jeep",  
289         description = " Returns the created Jeep",  
290         responses = {  
291             @ApiResponse(  
292                 responseCode = 200,  
293                 description = "Successful response",  
294                 content = {  
295                     @Content(mediaType = "application/json",  
296                         schema = @Schema(implementation = JeepOrder.class))  
297                 }  
298             )  
299         }  
300     )  
301     List<JeepOrder> getAllJeepOrders();  
302  
303     @Operation(  
304         summary = " Get an order for Jeep",  
305         description = " Returns the created Jeep",  
306         responses = {  
307             @ApiResponse(  
308                 responseCode = 200,  
309                 description = "Successful response",  
310                 content = {  
311                     @Content(mediaType = "application/json",  
312                         schema = @Schema(implementation = JeepOrder.class))  
313                 }  
314             )  
315         }  
316     )  
317     JeepOrder getJeepOrder(Long id);  
318  
319     @Operation(  
320         summary = " Delete an order for Jeep",  
321         description = " Returns the created Jeep",  
322         responses = {  
323             @ApiResponse(  
324                 responseCode = 200,  
325                 description = "Successful response",  
326                 content = {  
327                     @Content(mediaType = "application/json",  
328                         schema = @Schema(implementation = JeepOrder.class))  
329                 }  
330             )  
331         }  
332     )  
333     void deleteJeepOrder(Long id);  
334  
335     @Operation(  
336         summary = " Update an order for Jeep",  
337         description = " Returns the created Jeep",  
338         responses = {  
339             @ApiResponse(  
340                 responseCode = 200,  
341                 description = "Successful response",  
342                 content = {  
343                     @Content(mediaType = "application/json",  
344                         schema = @Schema(implementation = JeepOrder.class))  
345                 }  
346             )  
347         }  
348     )  
349     JeepOrder updateJeepOrder(Long id, JeepOrder jeepOrder);  
350  
351     @Operation(  
352         summary = " Get all orders for Jeep",  
353         description = " Returns the created Jeep",  
354         responses = {  
355             @ApiResponse(  
356                 responseCode = 200,  
357                 description = "Successful response",  
358                 content = {  
359                     @Content(mediaType = "application/json",  
360                         schema = @Schema(implementation = JeepOrder.class))  
361                 }  
362             )  
363         }  
364     )  
365     List<JeepOrder> getAllJeepOrders();  
366  
367     @Operation(  
368         summary = " Get an order for Jeep",  
369         description = " Returns the created Jeep",  
370         responses = {  
371             @ApiResponse(  
372                 responseCode = 200,  
373                 description = "Successful response",  
374                 content = {  
375                     @Content(mediaType = "application/json",  
376                         schema = @Schema(implementation = JeepOrder.class))  
377                 }  
378             )  
379         }  
380     )  
381     JeepOrder getJeepOrder(Long id);  
382  
383     @Operation(  
384         summary = " Delete an order for Jeep",  
385         description = " Returns the created Jeep",  
386         responses = {  
387             @ApiResponse(  
388                 responseCode = 200,  
389                 description = "Successful response",  
390                 content = {  
391                     @Content(mediaType = "application/json",  
392                         schema = @Schema(implementation = JeepOrder.class))  
393                 }  
394             )  
395         }  
396     )  
397     void deleteJeepOrder(Long id);  
398  
399     @Operation(  
400         summary = " Update an order for Jeep",  
401         description = " Returns the created Jeep",  
402         responses = {  
403             @ApiResponse(  
404                 responseCode = 200,  
405                 description = "Successful response",  
406                 content = {  
407                     @Content(mediaType = "application/json",  
408                         schema = @Schema(implementation = JeepOrder.class))  
409                 }  
410             )  
411         }  
412     )  
413     JeepOrder updateJeepOrder(Long id, JeepOrder jeepOrder);  
414  
415     @Operation(  
416         summary = " Get all orders for Jeep",  
417         description = " Returns the created Jeep",  
418         responses = {  
419             @ApiResponse(  
420                 responseCode = 200,  
421                 description = "Successful response",  
422                 content = {  
423                     @Content(mediaType = "application/json",  
424                         schema = @Schema(implementation = JeepOrder.class))  
425                 }  
426             )  
427         }  
428     )  
429     List<JeepOrder> getAllJeepOrders();  
430  
431     @Operation(  
432         summary = " Get an order for Jeep",  
433         description = " Returns the created Jeep",  
434         responses = {  
435             @ApiResponse(  
436                 responseCode = 200,  
437                 description = "Successful response",  
438                 content = {  
439                     @Content(mediaType = "application/json",  
440                         schema = @Schema(implementation = JeepOrder.class))  
441                 }  
442             )  
443         }  
444     )  
445     JeepOrder getJeepOrder(Long id);  
446  
447     @Operation(  
448         summary = " Delete an order for Jeep",  
449         description = " Returns the created Jeep",  
450         responses = {  
451             @ApiResponse(  
452                 responseCode = 200,  
453                 description = "Successful response",  
454                 content = {  
455                     @Content(mediaType = "application/json",  
456                         schema = @Schema(implementation = JeepOrder.class))  
457                 }  
458             )  
459         }  
460     )  
461     void deleteJeepOrder(Long id);  
462  
463     @Operation(  
464         summary = " Update an order for Jeep",  
465         description = " Returns the created Jeep",  
466         responses = {  
467             @ApiResponse(  
468                 responseCode = 200,  
469                 description = "Successful response",  
470                 content = {  
471                     @Content(mediaType = "application/json",  
472                         schema = @Schema(implementation = JeepOrder.class))  
473                 }  
474             )  
475         }  
476     )  
477     JeepOrder updateJeepOrder(Long id, JeepOrder jeepOrder);  
478  
479     @Operation(  
480         summary = " Get all orders for Jeep",  
481         description = " Returns the created Jeep",  
482         responses = {  
483             @ApiResponse(  
484                 responseCode = 200,  
485                 description = "Successful response",  
486                 content = {  
487                     @Content(mediaType = "application/json",  
488                         schema = @Schema(implementation = JeepOrder.class))  
489                 }  
490             )  
491         }  
492     )  
493     List<JeepOrder> getAllJeepOrders();  
494  
495     @Operation(  
496         summary = " Get an order for Jeep",  
497         description = " Returns the created Jeep",  
498         responses = {  
499             @ApiResponse(  
500                 responseCode = 200,  
501                 description = "Successful response",  
502                 content = {  
503                     @Content(mediaType = "application/json",  
504                         schema = @Schema(implementation = JeepOrder.class))  
505                 }  
506             )  
507         }  
508     )  
509     JeepOrder getJeepOrder(Long id);  
510  
511     @Operation(  
512         summary = " Delete an order for Jeep",  
513         description = " Returns the created Jeep",  
514         responses = {  
515             @ApiResponse(  
516                 responseCode = 200,  
517                 description = "Successful response",  
518                 content = {  
519                     @Content(mediaType = "application/json",  
520                         schema = @Schema(implementation = JeepOrder.class))  
521                 }  
522             )  
523         }  
524     )  
525     void deleteJeepOrder(Long id);  
526  
527     @Operation(  
528         summary = " Update an order for Jeep",  
529         description = " Returns the created Jeep",  
530         responses = {  
531             @ApiResponse(  
532                 responseCode = 200,  
533                 description = "Successful response",  
534                 content = {  
535                     @Content(mediaType = "application/json",  
536                         schema = @Schema(implementation = JeepOrder.class))  
537                 }  
538             )  
539         }  
540     )  
541     JeepOrder updateJeepOrder(Long id, JeepOrder jeepOrder);  
542  
543     @Operation(  
544         summary = " Get all orders for Jeep",  
545         description = " Returns the created Jeep",  
546         responses = {  
547             @ApiResponse(  
548                 responseCode = 200,  
549                 description = "Successful response",  
550                 content = {  
551                     @Content(mediaType = "application/json",  
552                         schema = @Schema(implementation = JeepOrder.class))  
553                 }  
554             )  
555         }  
556     )  
557     List<JeepOrder> getAllJeepOrders();  
558  
559     @Operation(  
560         summary = " Get an order for Jeep",  
561         description = " Returns the created Jeep",  
562         responses = {  
563             @ApiResponse(  
564                 responseCode = 200,  
565                 description = "Successful response",  
566                 content = {  
567                     @Content(mediaType = "application/json",  
568                         schema = @Schema(implementation = JeepOrder.class))  
569                 }  
570             )  
571         }  
572     )  
573     JeepOrder getJeepOrder(Long id);  
574  
575     @Operation(  
576         summary = " Delete an order for Jeep",  
577         description = " Returns the created Jeep",  
578         responses = {  
579             @ApiResponse(  
580                 responseCode = 200,  
581                 description = "Successful response",  
582                 content = {  
583                     @Content(mediaType = "application/json",  
584                         schema = @Schema(implementation = JeepOrder.class))  
585                 }  
586             )  
587         }  
588     )  
589     void deleteJeepOrder(Long id);  
590  
591     @Operation(  
592         summary = " Update an order for Jeep",  
593         description = " Returns the created Jeep",  
594         responses = {  
595             @ApiResponse(  
596                 responseCode = 200,  
597                 description = "Successful response",  
598                 content = {  
599                     @Content(mediaType = "application/json",  
600                         schema = @Schema(implementation = JeepOrder.class))  
601                 }  
602             )  
603         }  
604     )  
605     JeepOrder updateJeepOrder(Long id, JeepOrder jeepOrder);  
606  
607     @Operation(  
608         summary = " Get all orders for Jeep",  
609         description = " Returns the created Jeep",  
610         responses = {  
611             @ApiResponse(  
612                 responseCode = 200,  
613                 description = "Successful response",  
614                 content = {  
615                     @Content(mediaType = "application/json",  
616                         schema = @Schema(implementation = JeepOrder.class))  
617                 }  
618             )  
619         }  
620     )  
621     List<JeepOrder> getAllJeepOrders();  
622  
623     @Operation(  
624         summary = " Get an order for Jeep",  
625         description = " Returns the created Jeep",  
626         responses = {  
627             @ApiResponse(  
628                 responseCode = 200,  
629                 description = "Successful response",  
630                 content = {  
631                     @Content(mediaType = "application/json",  
632                         schema = @Schema(implementation = JeepOrder.class))  
633                 }  
634             )  
635         }  
636     )  
637     JeepOrder getJeepOrder(Long id);  
638  
639     @Operation(  
640         summary = " Delete an order for Jeep",  
641         description = " Returns the created Jeep",  
642         responses = {  
643             @ApiResponse(  
644                 responseCode = 200,  
645                 description = "Successful response",  
646                 content = {  
647                     @Content(mediaType = "application/json",  
648                         schema = @Schema(implementation = JeepOrder.class))  
649                 }  
650             )  
651         }  
652     )  
653     void deleteJeepOrder(Long id);  
654  
655     @Operation(  
656         summary = " Update an order for Jeep",  
657         description = " Returns the created Jeep",  
658         responses = {  
659             @ApiResponse(  
660                 responseCode = 200,  
661                 description = "Successful response",  
662                 content = {  
663                     @Content(mediaType = "application/json",  
664                         schema = @Schema(implementation = JeepOrder.class))  
665                 }  
666             )  
667         }  
668     )  
669     JeepOrder updateJeepOrder(Long id, JeepOrder jeepOrder);  
670  
671     @Operation(  
672         summary = " Get all orders for Jeep",  
673         description = " Returns the created Jeep",  
674         responses = {  
675             @ApiResponse(  
676                 responseCode = 200,  
677                 description = "Successful response",  
678                 content = {  
679                     @Content(mediaType = "application/json",  
680                         schema = @Schema(implementation = JeepOrder.class))  
681                 }  
682             )  
683         }  
684     )  
685     List<JeepOrder> getAllJeepOrders();  
686  
687     @Operation(  
688         summary = " Get an order for Jeep",  
689         description = " Returns the created Jeep",  
690         responses = {  
691             @ApiResponse(  
692                 responseCode = 200,  
693                 description = "Successful response",  
694                 content = {  
695                     @Content(mediaType = "application/json",  
696                         schema = @Schema(implementation = JeepOrder.class))  
697                 }  
698             )  
699         }  
700     )  
701     JeepOrder getJeepOrder(Long id);  
702  
703     @Operation(  
704         summary = " Delete an order for Jeep",  
705         description = " Returns the created Jeep",  
706         responses = {  
707             @ApiResponse(  
708                 responseCode = 200,  
709                 description = "Successful response",  
710                 content = {  
711                     @Content(mediaType = "application/json",  
712                         schema = @Schema(implementation = JeepOrder.class))  
713                 }  
714             )  
715         }  
716     )  
717     void deleteJeepOrder(Long id);  
718  
719     @Operation(  
720         summary = " Update an order for Jeep",  
721         description = " Returns the created Jeep",  
722         responses = {  
723             @ApiResponse(  
724                 responseCode = 200,  
725                 description = "Successful response",  
726                 content = {  
727                     @Content(mediaType = "application/json",  
728                         schema = @Schema(implementation = JeepOrder.class))  
729                 }  
730             )  
731         }  
732     )  
733     JeepOrder updateJeepOrder(Long id, JeepOrder jeepOrder);  
734  
735     @Operation(  
736         summary = " Get all orders for Jeep",  
737         description = " Returns the created Jeep",  
738         responses = {  
739             @ApiResponse(  
740                 responseCode = 200,  
741                 description = "Successful response",  
742                 content = {  
743                     @Content(mediaType = "application/json",  
744                         schema = @Schema(implementation = JeepOrder.class))  
745                 }  
746             )  
747         }  
748     )  
749     List<JeepOrder> getAllJeepOrders();  
750  
751     @Operation(  
752         summary = " Get an order for Jeep",  
753         description = " Returns the created Jeep",  
754         responses = {  
755             @ApiResponse(  
756                 responseCode = 200,  
757                 description = "Successful response",  
758                 content = {  
759                     @Content(mediaType = "application/json",  
760                         schema = @Schema(implementation = JeepOrder.class))  
761                 }  
762             )  
763         }  
764     )  
765     JeepOrder getJeepOrder(Long id);  
766  
767     @Operation(  
768         summary = " Delete an order for Jeep",  
769         description = " Returns the created Jeep",  
770         responses = {  
771             @ApiResponse(  
772                 responseCode = 200,  
773                 description = "Successful response",  
774                 content = {  
775                     @Content(mediaType = "application/json",  
776                         schema = @Schema(implementation = JeepOrder.class))  
777                 }  
778             )  
779         }  
780     )  
781     void deleteJeepOrder(Long id);  
782  
783     @Operation(  
784         summary = " Update an order for Jeep",  
785         description = " Returns the created Jeep",  
786         responses = {  
787             @ApiResponse(  
788                 responseCode = 200,  
789                 description = "Successful response",  
790                 content = {  
791                     @Content(mediaType = "application/json",  
792                         schema = @Schema(implementation = JeepOrder.class))  
793                 }  
794             )  
795         }  
796     )  
797     JeepOrder updateJeepOrder(Long id, JeepOrder jeepOrder);  
798  
799     @Operation(  
800         summary = " Get all orders for Jeep",  
801         description = " Returns the created Jeep",  
802         responses = {  
803             @ApiResponse(  
804                 responseCode = 200,  
805                 description = "Successful response",  
806                 content = {  
807                     @Content(mediaType = "application/json",  
808                         schema = @Schema(implementation = JeepOrder.class))  
809                 }  
810             )  
811         }  
812     )  
813     List<JeepOrder> getAllJeepOrders();  
814  
815     @Operation(  
816         summary = " Get an order for Jeep",  
817         description = " Returns the created Jeep",  
818         responses = {  
819             @ApiResponse(  
820                 responseCode = 200,  
821                 description = "Successful response",  
822                 content = {  
823                     @Content(mediaType = "application/json",  
824                         schema = @Schema(implementation = JeepOrder.class))  
825                 }  
826             )  
827         }  
828     )  
829     JeepOrder getJeepOrder(Long id);  
830  
831     @Operation(  
832         summary = " Delete an order for Jeep",  
833         description = " Returns the created Jeep",  
834         responses = {  
835             @ApiResponse(  
836                 responseCode = 200,  
837                 description = "Successful response",  
838                 content = {  
839                     @Content(mediaType = "application/json",  
840                         schema = @Schema(implementation = JeepOrder.class))  
841                 }  
842             )  
843         }  
844     )  
845     void deleteJeepOrder(Long id);  
846  
847     @Operation(  
848         summary = " Update an order for Jeep",  
849         description = " Returns the created Jeep",  
850         responses = {  
851             @ApiResponse(  
852                 responseCode = 200,  
853                 description = "Successful response",  
854                 content = {  
855                     @Content(mediaType = "application/json",  
856                         schema = @Schema(implementation = JeepOrder.class))  
857                 }  
858             )  
859         }  
860     )  
861     JeepOrder updateJeepOrder(Long id, JeepOrder jeepOrder);  
862  
863     @Operation(  
864         summary = " Get all orders for Jeep",  
865         description = " Returns the created Jeep",  
866         responses = {  
867             @ApiResponse(  
868                 responseCode = 200,  
869                 description = "Successful response",  
870                 content = {  
871                     @Content(mediaType = "application/json",  
872                         schema = @Schema(implementation = JeepOrder.class))  
873                 }  
874             )  
875         }  
876     )  
877     List<JeepOrder> getAllJeepOrders();  
878  
879     @Operation(  
880         summary = " Get an order for Jeep",  
881         description = " Returns the created Jeep",  
882         responses = {  
883             @ApiResponse(  
884                 responseCode = 200,  
885                 description = "Successful response",  
886                 content = {  
887                     @Content(mediaType = "application/json",  
888                         schema = @Schema(implementation = JeepOrder.class))  
889                 }  
890             )  
891         }  
892     )  
893     JeepOrder getJeepOrder(Long id);  
894  
895     @Operation(  
896         summary = " Delete an order for Jeep",  
897         description = " Returns the created Jeep",  
898         responses = {  
899             @ApiResponse(  
900                 responseCode = 200,  
901                 description = "Successful response",  
902                 content = {  
903                     @Content(mediaType = "application/json",  
904                         schema = @Schema(implementation = JeepOrder.class))  
905                 }  
906             )  
907         }  
908     )  
909     void deleteJeepOrder(Long id);  
910  
911     @Operation(  
912         summary = " Update an order for Jeep",  
913         description = " Returns the created Jeep",  
914         responses = {  
915             @ApiResponse(  
916                 responseCode = 200,  
917                 description = "Successful response",  
918                 content = {  
919                     @Content(mediaType = "application/json",  
920                         schema = @Schema(implementation = JeepOrder.class))  
921                 }  
922             )  
923         }  
924     )  
925     JeepOrder updateJeepOrder(Long id, JeepOrder jeepOrder);  
926  
927     @Operation(  
928         summary = " Get all orders for Jeep",  
929         description = " Returns the created Jeep",  
930         responses = {  
931             @ApiResponse(  
932                 responseCode = 200,  
933                 description = "Successful response",  
934                 content = {  
935                     @Content(mediaType = "application/json",  
936                         schema = @Schema(implementation = JeepOrder.class))  
937                 }  
938             )  
939         }  
940     )  
941     List<JeepOrder> getAllJeepOrders();  
942  
943     @Operation(  
944         summary = " Get an order for Jeep",  
945         description = " Returns the created Jeep",  
946         responses = {  
947             @ApiResponse(  
948                 responseCode = 200,  
949                 description = "Successful response",  
950                 content = {  
951                     @Content(mediaType = "application/json",  
952                         schema = @Schema(implementation = JeepOrder.class))  
953                 }  
954             )  
955         }  
956     )  
957     JeepOrder getJeepOrder(Long id);  
958  
959     @Operation(  
960         summary = " Delete an order for Jeep",  
961         description = " Returns the created Jeep",  
962         responses = {  
963             @ApiResponse(  
964                 responseCode = 200,  
965                 description = "Successful response",  
966                 content = {  
967                     @Content(mediaType = "application/json",  
968                         schema = @Schema(implementation = JeepOrder.class))  
969                 }  
970             )  
971         }  
972     )  
973     void deleteJeepOrder(Long id);  
974  
975     @Operation(  
976         summary = " Update an order for Jeep",  
977         description = " Returns the created Jeep",  
978         responses = {  
979             @ApiResponse(  
980                 responseCode = 200,  
981                 description = "Successful response",  
982                 content = {  
983                     @Content(mediaType = "application/json",  
984                         schema = @Schema(implementation = JeepOrder.class))  
985                 }  
986             )  
987         }  
988     )  
989     JeepOrder updateJeepOrder(Long id, JeepOrder jeepOrder);  
990  
991     @Operation(  
992         summary = " Get all orders for Jeep",  
993         description = " Returns the created Jeep",  
994         responses = {  
995             @ApiResponse(  
996                 responseCode = 200,  
997                 description = "Successful response",  
998                 content = {  
999                     @Content(mediaType = "application/json",  
1000                        schema = @Schema(implementation = JeepOrder.class))  
1001                    }  
1002                }<
```

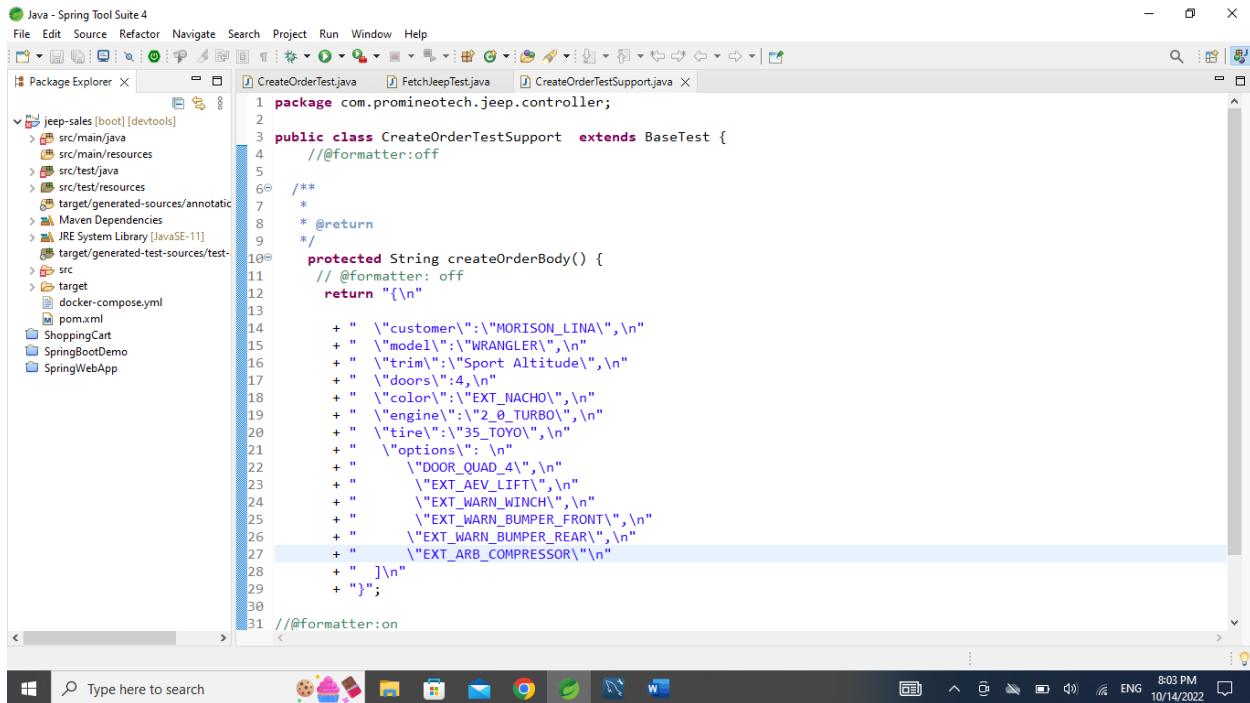


```
Java - jeep-sales/src/main/java/com/promineotech/jeep/controller/JeepOrderController.java - Spring Tool Suite 4
File Edit Source Refactor Navigate Search Project Run Window Help
Project ... X *JeepOrderController.java X
jeep-sales [boot] [dev]
  src/main/java
    com.promineotech
      JeepOrderController.java
        package com.promineotech.jeepp.controller;
        import java.util.List;

        @Validated //turns bean validation on
        @RequestMapping("/orders")
        @OpenAPIDefinition(info = @Info(title = "Jeep Order Service"), servers = {
            @Server(url = "http://localhost:8080", description = "Local server.")})
        public interface JeepOrderController {
            //formatter:off
            @Operation(
                summary = " Create an order for Jeep",
                description = " Returns the created Jeep",
                responses = {
                    @ApiResponse(
                        responseCode = "201",
                        description = "The created Jeep is returned",
                        content = @Content(
                            mediaType = "Application/Json",
                            schema = @Schema(implementation = Order.class)),
                    @ApiResponse(
                        responseCode = "400",
                        description = "The request parameter was invalid",
                        content = @Content(mediaType = "Application/Json")),
                    @ApiResponse(
                        responseCode = "404",
                        description = "A Jeep component was not found with the input criteria ",
                        content = @Content(mediaType = "Application/Json")),
                    @ApiResponse(
                        responseCode = "500",
                        description = "An unplanned error was occurred ",
                        content = @Content(mediaType = "Application/Json")),
                }
            )
        }
        Writeable Smart Insert 28 : 83 : 1420 7:15 PM 10/23/2022
```

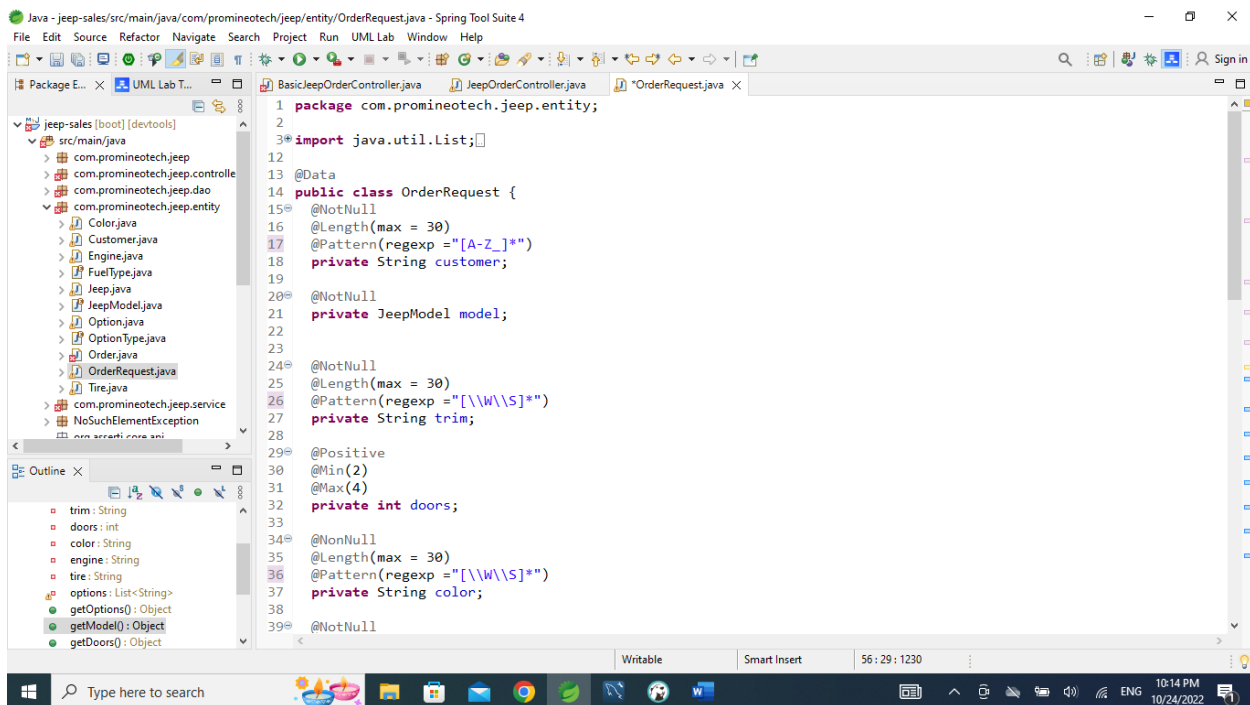


```
        mediaType = "Application/Json",
        schema = @Schema(implementation = Order.class)),
        @ApiResponse(
            responseCode = "400",
            description = "The request parameter was invalid",
            content = @Content(mediaType = "Application/Json")),
        @ApiResponse(
            responseCode = "404",
            description = "A Jeep component was not found with the input criteria ",
            content = @Content(mediaType = "Application/Json")),
        @ApiResponse(
            responseCode = "500",
            description = "An unplanned error was occurred ",
            content = @Content(mediaType = "Application/Json")),
        },
        parameters = {
            @Parameter(
                name = "OrderRequest",
                required = true,
                description = "The order as Json")
        }
    )
    @PostMapping
    @ResponseStatus(code = HttpStatus.CREATED)
    Order createOrder(@Valid @RequestBody OrderRequest orderRequest);
    List<Order> createOrders();
    //formatter: on
}
Writeable Smart Insert 28 : 83 : 1420 7:16 PM 10/23/2022
```



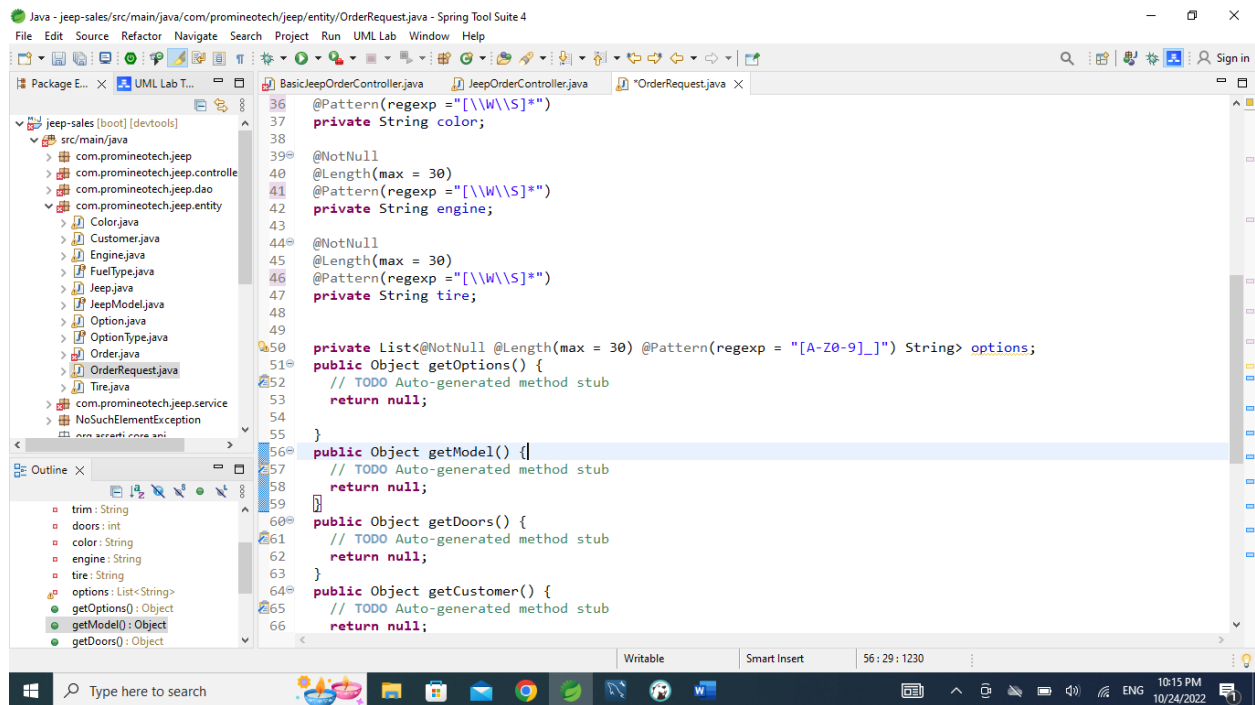
The screenshot shows the Spring Tool Suite 4 IDE with the 'CreateOrderTestSupport.java' file open. The file is located in the package 'com.promineotech.jeeep.controller;'. It is a public class that extends 'BaseTest'. The code includes a JUnit 5 test method 'createOrderBody()' which uses the '@formatter:off' annotation to format a JSON string. The JSON string represents a vehicle with the following details: customer 'MORISON_LINA', model 'WRANGLER', trim 'Sport Altitude', 4 doors, color 'EXT_NACHO', engine '2_0_TURBO', tire '35_TOYO', and various options including 'DOOR_QUAD_4', 'EXT_AEV_LIFT', 'EXT_WARN_WINGCH', 'EXT_WARN BUMPER_FRONT', 'EXT_WARN BUMPER_REAR', and 'EXT_ARB_COMPRESSOR'. The test method returns the formatted JSON string and is annotated with '@return' and '@formatter:on'.

```
1 package com.promineotech.jeeep.controller;
2
3 public class CreateOrderTestSupport extends BaseTest {
4     // @formatter:off
5
6     /**
7      *
8      * @return
9      */
10    protected String createOrderBody() {
11        // @formatter: off
12        return "{\n"
13
14            + "  \"customer\": \"MORISON_LINA\", \n"
15            + "  \"model\": \"WRANGLER\", \n"
16            + "  \"trim\": \"Sport Altitude\", \n"
17            + "  \"doors\": 4, \n"
18            + "  \"color\": \"EXT_NACHO\", \n"
19            + "  \"engine\": \"2_0_TURBO\", \n"
20            + "  \"tire\": \"35_TOYO\", \n"
21            + "  \"options\": [\n"
22            + "    \"DOOR_QUAD_4\", \n"
23            + "    \"EXT_AEV_LIFT\", \n"
24            + "    \"EXT_WARN_WINGCH\", \n"
25            + "    \"EXT_WARN BUMPER_FRONT\", \n"
26            + "    \"EXT_WARN BUMPER_REAR\", \n"
27            + "    \"EXT_ARB_COMPRESSOR\" \n"
28            + "  ] \n"
29            + "}";
30
31    // @formatter:on
```

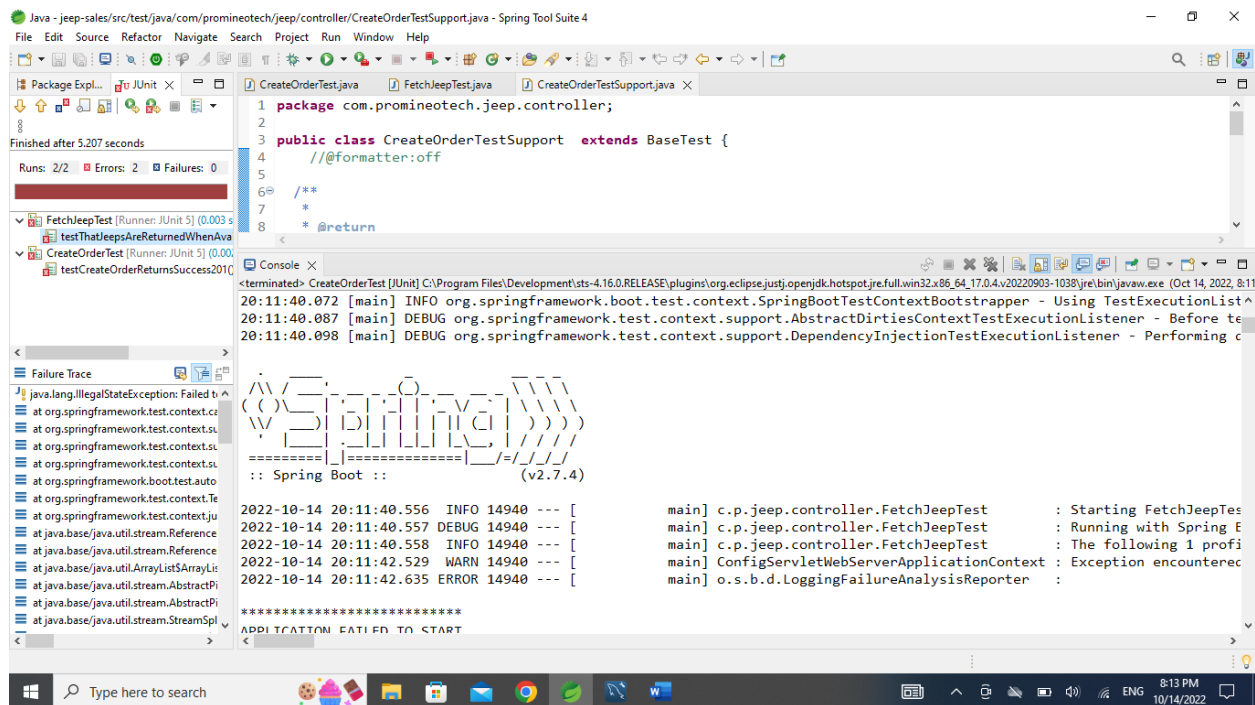


The screenshot shows the Spring Tool Suite 4 IDE with the 'OrderRequest.java' file open. The file is located in the package 'com.promineotech.jeeep.entity;'. It is a public class that represents an order request. The class has several attributes: 'customer' (String, not null, length max 30, pattern '[A-Z]*'), 'model' (JeepModel, not null), 'trim' (String, not null, length max 30, pattern '[\\W\\S]*'), 'doors' (int, positive, min 2, max 4), and 'color' (String, not null, length max 30, pattern '[\\W\\S]*'). The class also has methods 'getOptions()' and 'getDoors()' which return a list of strings and an integer respectively. The class is annotated with '@Data' and '@Pattern' for validation.

```
1 package com.promineotech.jeeep.entity;
2
3 import java.util.List;
4
5 @Data
6 public class OrderRequest {
7     @NotNull
8     @Length(max = 30)
9     @Pattern(regexp = "[A-Z]*")
10    private String customer;
11
12    @NotNull
13    private JeepModel model;
14
15    @NotNull
16    @Length(max = 30)
17    @Pattern(regexp = "[\\W\\S]*")
18    private String trim;
19
20    @Positive
21    @Min(2)
22    @Max(4)
23    private int doors;
24
25    @NotNull
26    @Length(max = 30)
27    @Pattern(regexp = "[\\W\\S]*")
28    private String color;
29
30    @NotNull
31    private List<String> options;
32
33    @NotNull
34    private Integer doors;
35
36    @NotNull
37    private String color;
38
39    @NotNull
40    private Integer doors;
41
42    @NotNull
43    private String color;
44
45    @NotNull
46    private Integer doors;
47
48    @NotNull
49    private String color;
50
51    @NotNull
52    private Integer doors;
53
54    @NotNull
55    private String color;
56
57    @NotNull
58    private Integer doors;
59
60    @NotNull
61    private String color;
62
63    @NotNull
64    private Integer doors;
65
66    @NotNull
67    private String color;
68
69    @NotNull
70    private Integer doors;
71
72    @NotNull
73    private String color;
74
75    @NotNull
76    private Integer doors;
77
78    @NotNull
79    private String color;
80
81    @NotNull
82    private Integer doors;
83
84    @NotNull
85    private String color;
86
87    @NotNull
88    private Integer doors;
89
90    @NotNull
91    private String color;
92
93    @NotNull
94    private Integer doors;
95
96    @NotNull
97    private String color;
98
99    @NotNull
100   private Integer doors;
101
102   @NotNull
103   private String color;
104
105   @NotNull
106   private Integer doors;
107
108   @NotNull
109   private String color;
110
111   @NotNull
112   private Integer doors;
113
114   @NotNull
115   private String color;
116
117   @NotNull
118   private Integer doors;
119
120   @NotNull
121   private String color;
122
123   @NotNull
124   private Integer doors;
125
126   @NotNull
127   private String color;
128
129   @NotNull
130   private Integer doors;
131
132   @NotNull
133   private String color;
134
135   @NotNull
136   private Integer doors;
137
138   @NotNull
139   private String color;
140
141   @NotNull
142   private Integer doors;
143
144   @NotNull
145   private String color;
146
147   @NotNull
148   private Integer doors;
149
150   @NotNull
151   private String color;
152
153   @NotNull
154   private Integer doors;
155
156   @NotNull
157   private String color;
158
159   @NotNull
160   private Integer doors;
161
162   @NotNull
163   private String color;
164
165   @NotNull
166   private Integer doors;
167
168   @NotNull
169   private String color;
170
171   @NotNull
172   private Integer doors;
173
174   @NotNull
175   private String color;
176
177   @NotNull
178   private Integer doors;
179
180   @NotNull
181   private String color;
182
183   @NotNull
184   private Integer doors;
185
186   @NotNull
187   private String color;
188
189   @NotNull
190   private Integer doors;
191
192   @NotNull
193   private String color;
194
195   @NotNull
196   private Integer doors;
197
198   @NotNull
199   private String color;
200
201   @NotNull
202   private Integer doors;
203
204   @NotNull
205   private String color;
206
207   @NotNull
208   private Integer doors;
209
210   @NotNull
211   private String color;
212
213   @NotNull
214   private Integer doors;
215
216   @NotNull
217   private String color;
218
219   @NotNull
220   private Integer doors;
221
222   @NotNull
223   private String color;
224
225   @NotNull
226   private Integer doors;
227
228   @NotNull
229   private String color;
230
231   @NotNull
232   private Integer doors;
233
234   @NotNull
235   private String color;
236
237   @NotNull
238   private Integer doors;
239
240   @NotNull
241   private String color;
242
243   @NotNull
244   private Integer doors;
245
246   @NotNull
247   private String color;
248
249   @NotNull
250   private Integer doors;
251
252   @NotNull
253   private String color;
254
255   @NotNull
256   private Integer doors;
257
258   @NotNull
259   private String color;
260
261   @NotNull
262   private Integer doors;
263
264   @NotNull
265   private String color;
266
267   @NotNull
268   private Integer doors;
269
270   @NotNull
271   private String color;
272
273   @NotNull
274   private Integer doors;
275
276   @NotNull
277   private String color;
278
279   @NotNull
280   private Integer doors;
281
282   @NotNull
283   private String color;
284
285   @NotNull
286   private Integer doors;
287
288   @NotNull
289   private String color;
290
291   @NotNull
292   private Integer doors;
293
294   @NotNull
295   private String color;
296
297   @NotNull
298   private Integer doors;
299
300   @NotNull
301   private String color;
302
303   @NotNull
304   private Integer doors;
305
306   @NotNull
307   private String color;
308
309   @NotNull
310   private Integer doors;
311
312   @NotNull
313   private String color;
314
315   @NotNull
316   private Integer doors;
317
318   @NotNull
319   private String color;
320
321   @NotNull
322   private Integer doors;
323
324   @NotNull
325   private String color;
326
327   @NotNull
328   private Integer doors;
329
330   @NotNull
331   private String color;
332
333   @NotNull
334   private Integer doors;
335
336   @NotNull
337   private String color;
338
339   @NotNull
340   private Integer doors;
341
342   @NotNull
343   private String color;
344
345   @NotNull
346   private Integer doors;
347
348   @NotNull
349   private String color;
350
351   @NotNull
352   private Integer doors;
353
354   @NotNull
355   private String color;
356
357   @NotNull
358   private Integer doors;
359
360   @NotNull
361   private String color;
362
363   @NotNull
364   private Integer doors;
365
366   @NotNull
367   private String color;
368
369   @NotNull
370   private Integer doors;
371
372   @NotNull
373   private String color;
374
375   @NotNull
376   private Integer doors;
377
378   @NotNull
379   private String color;
380
381   @NotNull
382   private Integer doors;
383
384   @NotNull
385   private String color;
386
387   @NotNull
388   private Integer doors;
389
390   @NotNull
391   private String color;
392
393   @NotNull
394   private Integer doors;
395
396   @NotNull
397   private String color;
398
399   @NotNull
400   private Integer doors;
401
402   @NotNull
403   private String color;
404
405   @NotNull
406   private Integer doors;
407
408   @NotNull
409   private String color;
410
411   @NotNull
412   private Integer doors;
413
414   @NotNull
415   private String color;
416
417   @NotNull
418   private Integer doors;
419
420   @NotNull
421   private String color;
422
423   @NotNull
424   private Integer doors;
425
426   @NotNull
427   private String color;
428
429   @NotNull
430   private Integer doors;
431
432   @NotNull
433   private String color;
434
435   @NotNull
436   private Integer doors;
437
438   @NotNull
439   private String color;
440
441   @NotNull
442   private Integer doors;
443
444   @NotNull
445   private String color;
446
447   @NotNull
448   private Integer doors;
449
450   @NotNull
451   private String color;
452
453   @NotNull
454   private Integer doors;
455
456   @NotNull
457   private String color;
458
459   @NotNull
460   private Integer doors;
461
462   @NotNull
463   private String color;
464
465   @NotNull
466   private Integer doors;
467
468   @NotNull
469   private String color;
470
471   @NotNull
472   private Integer doors;
473
474   @NotNull
475   private String color;
476
477   @NotNull
478   private Integer doors;
479
480   @NotNull
481   private String color;
482
483   @NotNull
484   private Integer doors;
485
486   @NotNull
487   private String color;
488
489   @NotNull
490   private Integer doors;
491
492   @NotNull
493   private String color;
494
495   @NotNull
496   private Integer doors;
497
498   @NotNull
499   private String color;
500
501   @NotNull
502   private Integer doors;
503
504   @NotNull
505   private String color;
506
507   @NotNull
508   private Integer doors;
509
510   @NotNull
511   private String color;
512
513   @NotNull
514   private Integer doors;
515
516   @NotNull
517   private String color;
518
519   @NotNull
520   private Integer doors;
521
522   @NotNull
523   private String color;
524
525   @NotNull
526   private Integer doors;
527
528   @NotNull
529   private String color;
530
531   @NotNull
532   private Integer doors;
533
534   @NotNull
535   private String color;
536
537   @NotNull
538   private Integer doors;
539
540   @NotNull
541   private String color;
542
543   @NotNull
544   private Integer doors;
545
546   @NotNull
547   private String color;
548
549   @NotNull
550   private Integer doors;
551
552   @NotNull
553   private String color;
554
555   @NotNull
556   private Integer doors;
557
558   @NotNull
559   private String color;
560
561   @NotNull
562   private Integer doors;
563
564   @NotNull
565   private String color;
566
567   @NotNull
568   private Integer doors;
569
570   @NotNull
571   private String color;
572
573   @NotNull
574   private Integer doors;
575
576   @NotNull
577   private String color;
578
579   @NotNull
580   private Integer doors;
581
582   @NotNull
583   private String color;
584
585   @NotNull
586   private Integer doors;
587
588   @NotNull
589   private String color;
590
591   @NotNull
592   private Integer doors;
593
594   @NotNull
595   private String color;
596
597   @NotNull
598   private Integer doors;
599
600   @NotNull
601   private String color;
602
603   @NotNull
604   private Integer doors;
605
606   @NotNull
607   private String color;
608
609   @NotNull
610   private Integer doors;
611
612   @NotNull
613   private String color;
614
615   @NotNull
616   private Integer doors;
617
618   @NotNull
619   private String color;
620
621   @NotNull
622   private Integer doors;
623
624   @NotNull
625   private String color;
626
627   @NotNull
628   private Integer doors;
629
630   @NotNull
631   private String color;
632
633   @NotNull
634   private Integer doors;
635
636   @NotNull
637   private String color;
638
639   @NotNull
640   private Integer doors;
641
642   @NotNull
643   private String color;
644
645   @NotNull
646   private Integer doors;
647
648   @NotNull
649   private String color;
650
651   @NotNull
652   private Integer doors;
653
654   @NotNull
655   private String color;
656
657   @NotNull
658   private Integer doors;
659
660   @NotNull
661   private String color;
662
663   @NotNull
664   private Integer doors;
665
666   @NotNull
667   private String color;
668
669   @NotNull
670   private Integer doors;
671
672   @NotNull
673   private String color;
674
675   @NotNull
676   private Integer doors;
677
678   @NotNull
679   private String color;
680
681   @NotNull
682   private Integer doors;
683
684   @NotNull
685   private String color;
686
687   @NotNull
688   private Integer doors;
689
690   @NotNull
691   private String color;
692
693   @NotNull
694   private Integer doors;
695
696   @NotNull
697   private String color;
698
699   @NotNull
700   private Integer doors;
701
702   @NotNull
703   private String color;
704
705   @NotNull
706   private Integer doors;
707
708   @NotNull
709   private String color;
710
711   @NotNull
712   private Integer doors;
713
714   @NotNull
715   private String color;
716
717   @NotNull
718   private Integer doors;
719
720   @NotNull
721   private String color;
722
723   @NotNull
724   private Integer doors;
725
726   @NotNull
727   private String color;
728
729   @NotNull
730   private Integer doors;
731
732   @NotNull
733   private String color;
734
735   @NotNull
736   private Integer doors;
737
738   @NotNull
739   private String color;
740
741   @NotNull
742   private Integer doors;
743
744   @NotNull
745   private String color;
746
747   @NotNull
748   private Integer doors;
749
750   @NotNull
751   private String color;
752
753   @NotNull
754   private Integer doors;
755
756   @NotNull
757   private String color;
758
759   @NotNull
760   private Integer doors;
761
762   @NotNull
763   private String color;
764
765   @NotNull
766   private Integer doors;
767
768   @NotNull
769   private String color;
770
771   @NotNull
772   private Integer doors;
773
774   @NotNull
775   private String color;
776
777   @NotNull
778   private Integer doors;
779
780   @NotNull
781   private String color;
782
783   @NotNull
784   private Integer doors;
785
786   @NotNull
787   private String color;
788
789   @NotNull
790   private Integer doors;
791
792   @NotNull
793   private String color;
794
795   @NotNull
796   private Integer doors;
797
798   @NotNull
799   private String color;
800
801   @NotNull
802   private Integer doors;
803
804   @NotNull
805   private String color;
806
807   @NotNull
808   private Integer doors;
809
810   @NotNull
811   private String color;
812
813   @NotNull
814   private Integer doors;
815
816   @NotNull
817   private String color;
818
819   @NotNull
820   private Integer doors;
821
822   @NotNull
823   private String color;
824
825   @NotNull
826   private Integer doors;
827
828   @NotNull
829   private String color;
830
831   @NotNull
832   private Integer doors;
833
834   @NotNull
835   private String color;
836
837   @NotNull
838   private Integer doors;
839
840   @NotNull
841   private String color;
842
843   @NotNull
844   private Integer doors;
845
846   @NotNull
847   private String color;
848
849   @NotNull
850   private Integer doors;
851
852   @NotNull
853   private String color;
854
855   @NotNull
856   private Integer doors;
857
858   @NotNull
859   private String color;
860
861   @NotNull
862   private Integer doors;
863
864   @NotNull
865   private String color;
866
867   @NotNull
868   private Integer doors;
869
870   @NotNull
871   private String color;
872
873   @NotNull
874   private Integer doors;
875
876   @NotNull
877   private String color;
878
879   @NotNull
880   private Integer doors;
881
882   @NotNull
883   private String color;
884
885   @NotNull
886   private Integer doors;
887
888   @NotNull
889   private String color;
890
891   @NotNull
892   private Integer doors;
893
894   @NotNull
895   private String color;
896
897   @NotNull
898   private Integer doors;
899
900   @NotNull
901   private String color;
902
903   @NotNull
904   private Integer doors;
905
906   @NotNull
907   private String color;
908
909   @NotNull
910   private Integer doors;
911
912   @NotNull
913   private String color;
914
915   @NotNull
916   private Integer doors;
917
918   @NotNull
919   private String color;
920
921   @NotNull
922   private Integer doors;
923
924   @NotNull
925   private String color;
926
927   @NotNull
928   private Integer doors;
929
930   @NotNull
931   private String color;
932
933   @NotNull
934   private Integer doors;
935
936   @NotNull
937   private String color;
938
939   @NotNull
940   private Integer doors;
941
942   @NotNull
943   private String color;
944
945   @NotNull
946   private Integer doors;
947
948   @NotNull
949   private String color;
950
951   @NotNull
952   private Integer doors;
953
954   @NotNull
955   private String color;
956
957   @NotNull
958   private Integer doors;
959
960   @NotNull
961   private String color;
962
963   @NotNull
964   private Integer doors;
965
966   @NotNull
967   private String color;
968
969   @NotNull
970   private Integer doors;
971
972   @NotNull
973   private String color;
974
975   @NotNull
976   private Integer doors;
977
978   @NotNull
979   private String color;
980
981   @NotNull
982   private Integer doors;
983
984   @NotNull
985   private String color;
986
987   @NotNull
988   private Integer doors;
989
990   @NotNull
991   private String color;
992
993   @NotNull
994   private Integer doors;
995
996   @NotNull
997   private String color;
998
999   @NotNull
1000  private Integer doors;
1001
1002  @NotNull
1003  private String color;
1004
1005  @NotNull
1006  private Integer doors;
1007
1008  @NotNull
1009  private String color;
1010
1011  @NotNull
1012  private Integer doors;
1013
1014  @NotNull
1015  private String color;
1016
1017  @NotNull
1018  private Integer doors;
1019
1020  @NotNull
1021  private String color;
1022
1023  @NotNull
1024  private Integer doors;
1025
1026  @NotNull
1027  private String color;
1028
1029  @NotNull
1030  private Integer doors;
1031
1032  @NotNull
1033  private String color;
1034
1035  @NotNull
1036  private Integer doors;
1037
1038  @NotNull
1039  private String color;
1040
1041  @NotNull
1042  private Integer doors;
1043
1044  @NotNull
1045  private String color;
1046
1047  @NotNull
1048  private Integer doors;
1049
1050  @NotNull
1051  private String color;
1052
1053  @NotNull
1054  private Integer doors;
1055
1056  @NotNull
1057  private String color;
1058
1059  @NotNull
1060  private Integer doors;
1061
1062  @NotNull
1063  private String color;
1064
1065  @NotNull
1066  private Integer doors;
1067
1068  @NotNull
1069  private String color;
1070
1071  @NotNull
1072  private Integer doors;
1073
1074  @NotNull
1075  private String color;
1076
1077  @NotNull
1078  private Integer doors;
1079
1080  @NotNull
1081  private String color;
1082
1083  @NotNull
1084  private Integer doors;
1085
1086  @NotNull
1087  private String color;
1088
1089  @NotNull
1090  private Integer doors;
1091
1092  @NotNull
1093  private String color;
1094
1095  @NotNull
1096  private Integer doors;
1097
1098  @NotNull
1099  private String color;
1100
1101  @NotNull
1102  private Integer doors;
1103
1104  @NotNull
1105  private String color;
1106
1107  @NotNull
1108  private Integer doors;
1109
1110  @NotNull
1111  private String color;
1112
1113  @NotNull
1114  private Integer doors;
1115
1116  @NotNull
1117  private String color;
1118
1119  @NotNull
1120  private Integer doors;
1121
1122  @NotNull
1123  private String color;
1124
1125  @NotNull
1126  private Integer doors;
1127
1128  @NotNull
1129  private String color;
1130
1131  @NotNull
1132  private Integer doors;
1133
1134  @NotNull
1135  private String color;
1136
1137  @NotNull
1138  private Integer doors;
1139
1140  @NotNull
1141  private String color;
1142
1143  @NotNull
1144  private Integer doors;
1145
1146  @NotNull
1147  private String color;
1148
1149  @NotNull
1150  private Integer doors;
1151
1152  @NotNull
1153  private String color;
1154
1155  @NotNull
1156  private Integer doors;
1157
1158  @NotNull
1159  private String color;
1160
1161  @NotNull
1162  private Integer doors;
1163
1164  @NotNull
1165  private String color;
1166
1167  @NotNull
1168  private Integer doors;
1169
1170  @NotNull
1171  private String color;
1172
1173  @NotNull
1174  private Integer doors;
1175
1176  @NotNull
1177  private String color;
1178
1179  @NotNull
1180  private Integer doors;
1181
1182  @NotNull
1183  private String color;
1184
1185  @NotNull
1186  private Integer doors;
1187
1188  @NotNull
1189  private String color;
1190
1191  @NotNull
1192  private Integer doors;
1193
1194  @NotNull
1195  private String color;
1196
1197  @NotNull
1198  private Integer doors;
1199
1200  @NotNull
1201  private String color;
1202
1203  @NotNull
1204  private Integer doors;
1205
1206  @NotNull
1207  private String color;
1208
1209  @NotNull
1210  private Integer doors;
1211
1212  @NotNull
1213  private String color;
1214
1215  @NotNull
1216  private Integer doors;
1217
1218  @NotNull
1219  private String color;
1220
1221  @NotNull
1222  private Integer doors;
1223
1224  @NotNull
1225  private String color;
1226
1227  @NotNull
1228  private Integer doors;
1229
1230  @NotNull
1231  private String color;
1232
1233  @NotNull
1234  private Integer doors;
1235
1236  @NotNull
1237  private String color;
1238
1239  @NotNull
1240  private Integer doors;
1241
1242  @NotNull
1243  private String color;
1244
1245  @NotNull
1246  private Integer doors;
1247
1248  @NotNull
1249  private String color;
1250
1251  @NotNull
1252  private Integer doors;
1253
1254  @NotNull
1255  private String color;
1256
1257  @NotNull
1258  private Integer doors;
1259
1260  @NotNull
1261  private String color;
1262
1263  @NotNull
1264  private Integer doors;
1265
1266  @NotNull
1267  private String color;
1268
1269  @NotNull
1270  private Integer doors;
1271
1272  @NotNull
1273  private String color;
1274
1275  @NotNull
1276  private Integer doors;
1277
1278  @NotNull
1279  private String color;
1280
1281  @NotNull
1282  private Integer doors;
1283
1284  @NotNull
1285  private String color;
1286
1287  @NotNull
1288  private Integer doors;
1289
1290  @NotNull
1291  private String color;
1292
1293  @NotNull
1294  private Integer doors;
1295
1296  @NotNull
1297  private String color;
1298
1299  @NotNull
1300  private Integer doors;
1301
1302  @NotNull
1303  private String color;
1304
1305  @NotNull
1306  private Integer doors;
1307
1308  @NotNull
1309  private String color;
1310
1311  @NotNull
1312  private Integer doors;
1313
1314  @NotNull
1315  private String color;
1316
1317  @NotNull
1318  private Integer doors;
1319
1320  @NotNull
1321  private String color;
1322
1323  @NotNull
1324  private Integer doors;
1325
1326  @NotNull
1327  private String color;
1328
1329  @NotNull
1330  private Integer doors;
1331
1332  @NotNull
1333  private String color;
1334
1335  @NotNull
1336  private Integer doors;
1337
1338  @NotNull
1339  private String color;
1340
1341  @NotNull
1342  private Integer doors;
1343
1344  @NotNull
1345  private String color;
1346
1347  @NotNull
1348  private Integer doors;
1349
1350  @NotNull
1351  private String color;
1352
1353  @NotNull
1354  private Integer doors;
1355
1356  @NotNull
1357  private String color;
1358
1359  @NotNull
1360  private Integer doors;
1361
1362  @NotNull
1363  private String color;
1364
1365  @NotNull
1366  private Integer doors;
1367
1368  @NotNull
1369  private String color;
1370
1371  @NotNull
1372  private Integer doors;
1373
1374  @NotNull
1375  private String color;
1376
1377  @NotNull
1378  private Integer doors;
1379
1380  @NotNull
1381  private String color;
1382
1383  @NotNull
1384  private Integer doors;
1385
1386  @NotNull
1387  private String color;
1388
1389  @NotNull
1390  private Integer doors;
1391
1392  @NotNull
1393  private String color;
1394
1395  @NotNull
1396  private Integer doors;
1397
1398  @NotNull
1399  private String color;
1400
1401  @NotNull
1402  private Integer doors;
1403
1404  @NotNull
1405  private String color;
1406
1407  @NotNull
1408  private Integer doors;
1409
1410  @NotNull
1411  private String color;
1412
1413  @NotNull
1414  private Integer doors;
1415
1416  @NotNull
1417  private String color;
1418
1419  @NotNull
1420  private Integer doors;
1421
1422  @NotNull
1423  private String color;
1424
1425  @NotNull
1426  private Integer doors;
1427
1428  @NotNull
1429  private String color;
1430
1431  @NotNull
1432  private Integer doors;
1433
1434  @NotNull
1435  private String color;
1436
1437  @NotNull
1438  private Integer doors;
1439
1440  @NotNull
1441  private String color;
1442
1443  @NotNull
1444  private Integer doors;
1445
1446  @NotNull
1447  private String color;
1448
1449  @NotNull
1450  private Integer doors;
1451
1452  @NotNull
1453  private String color;
1454
1455  @NotNull
1456  private Integer doors;
1457
1458  @NotNull
1459  private String color;
1460
1461  @NotNull
1462  private Integer doors;
1463
1464  @NotNull
1465  private String color;
1466
1467  @NotNull
1468  private Integer doors;
1469
1470  @NotNull
1471  private String color;
1472
1473  @NotNull
1474  private Integer doors;
1475
1476  @NotNull
1477  private String color;
1478
1479  @NotNull
1480  private Integer doors;
1481
1482  @NotNull
1483  private String color;
1484
1485  @NotNull
1486  private Integer doors;
1487
1488  @NotNull
1489  private String color;
1490
1491  @NotNull
1492  private Integer doors;
1493
1494  @NotNull
1495  private String color;
1496
1497  @NotNull
1498  private Integer doors;
1499
1500  @NotNull
1501  private String color;
1502
1503  @NotNull
1504  private Integer doors;
1505
1506  @NotNull
1507  private String color;
1508
1509  @NotNull
1510  private Integer doors;
1511
1512  @NotNull
1513  private String color;
1514
1515  @NotNull
1516  private Integer doors;
1517
1518  @NotNull
1519  private String color;
1520
1521  @NotNull
1522  private Integer doors;
1523
1524  @NotNull
1525  private String color;
1526
1527  @NotNull
1528  private Integer doors;
1529
1530  @NotNull
1531  private String color;
1532
1533  @NotNull
1534  private Integer doors;
1535
1536  @NotNull
1537  private String color;
1538
1539  @NotNull
1540  private Integer doors;
1541
1542  @NotNull
1543  private String color;
1544
1545  @NotNull
1546  private Integer doors;
1547
1548  @NotNull
1549  private String color;
1550
1551  @NotNull
1552  private Integer doors;
1553
1554  @NotNull
1555  private String color;
1556
1557  @NotNull
1558  private Integer doors;
1559
1560  @NotNull
1561  private String color;
1562
1563  @NotNull
1564  private Integer doors;
1565
1566  @NotNull
1567  private String color;
1568
1569  @NotNull
1570  private Integer doors;
1571
1572  @NotNull
1573  private String color;
1574
1575  @NotNull
1576  private Integer doors;
1577
1578  @NotNull
1579  private String color;
1580
1581  @NotNull
1582  private Integer doors;
1583
1584  @NotNull
1585  private String color;
1586
1587  @NotNull
1588  private Integer doors;
1589
1590  @NotNull
1591  private String color;
1592
1593  @NotNull
1594  private Integer doors;
1595
1596  @NotNull
1597  private String color;
1598
1599  @NotNull
1600  private Integer doors;
1601
1602  @NotNull
1603  private String color;
1604
1605  @NotNull
1606  private Integer doors;
1607
1608  @NotNull
1609  private String color;
1610
1611  @NotNull
1612  private Integer doors;
1613
1614  @NotNull
1615  private String color;
1616
1617  @NotNull
1618  private Integer doors;
```



Screenshots of Running Application:



The screenshot shows an IDE window titled "Java - jeep-sales/src/main/java/com/promineotech/jeep/controller/JeepOrderController.java - Spring Tool Suite 4". The interface includes a Project Explorer on the left, a central editor showing the `JeepOrderController.java` file, and a Console window at the bottom.

Project Explorer: Shows a test run summary for `FetchJeepTest` (JUnit 5) with 2 runs, 2 errors, and 0 failures. The test `testThatJeepsAreReturnedWhenAvalidModelAndTrimAreSuppli` failed.

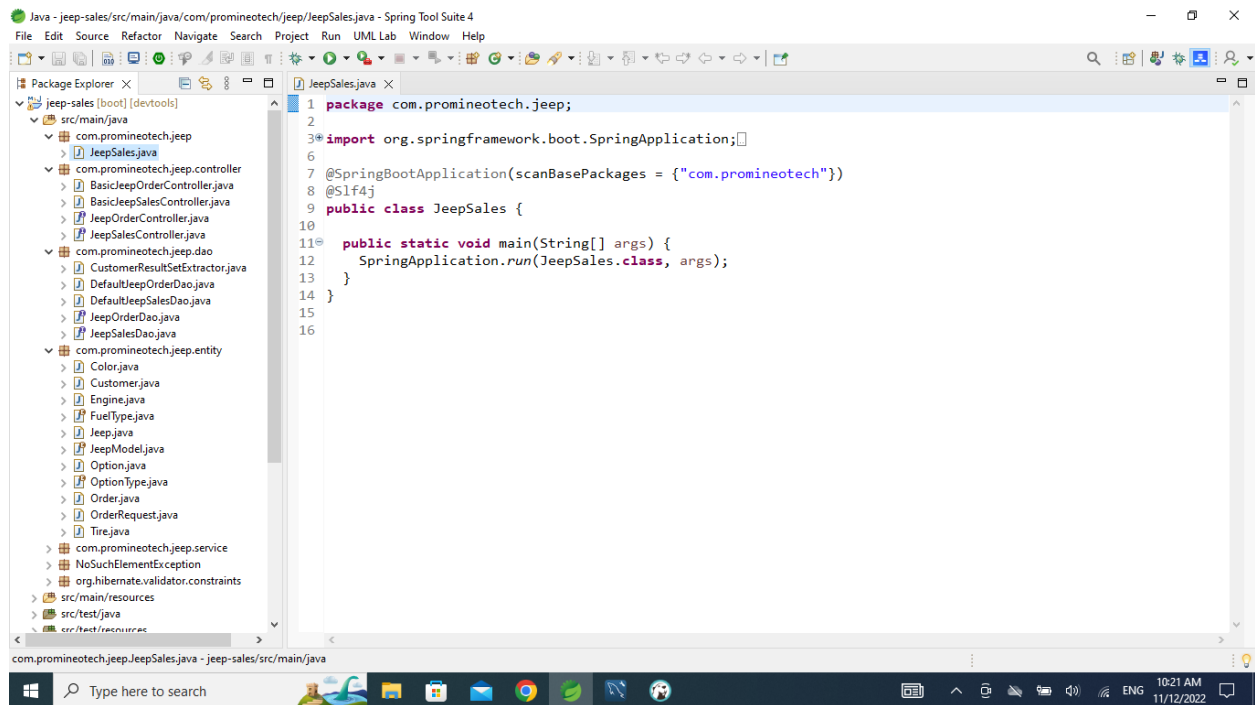
JeepOrderController.java: The code defines an OpenAPI endpoint for "/orders". It includes annotations for validation, request mapping, and server information. The `@Operation` annotation describes the endpoint's purpose and response.

```
24
25 @Validated //turns bean validation on
26 @RequestMapping("/orders")
27 @OpenAPIDefinition(info = @Info(title = "Jeep Order Service"), servers = {
28     @Server(url = "http://localhost:8080", description = "Local server.")})
29 public interface JeepOrderController {
30     //formatter:off
31     @Operation(
32         summary = "Create an order for Jeep",
33         description = "Returns the created Jeep",
34         responses = {
35             @ApiResponse(
36                 responseCode = "201",
37                 description = "The created Jeep is returned",
38                 content = @Content(
39                     mediaType = "Application/Json",
40                     schema = @Schema(implementation = Order.class))),
41         }
```

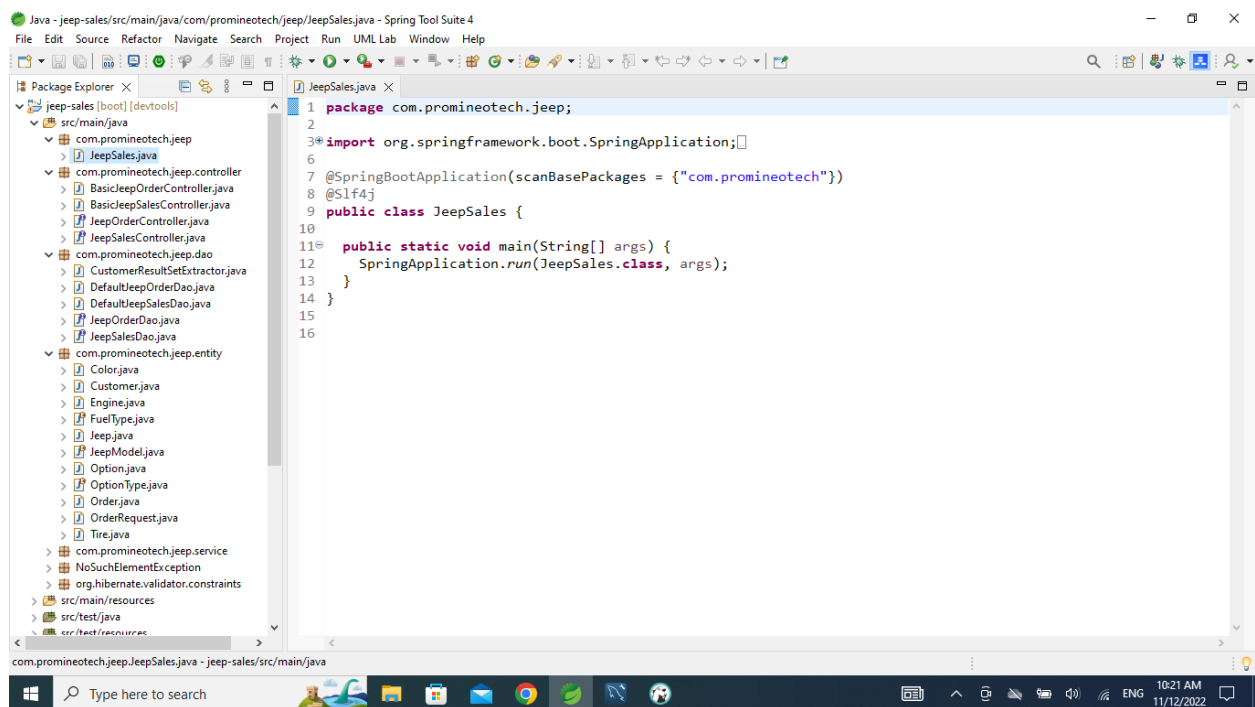
Failure Trace: The error message is `java.lang.IllegalStateException: Failed to load ApplicationContext`. The stack trace lists various Spring Framework classes involved in the context loading process.

Console: The console shows the output of the test run, including the Spring Boot logo and version (v2.7.4). The output indicates that the test `CreateOrderTest` failed.

```
<terminated> CreateOrderTest [JUnit] C:\Program Files\Development\sts-4.16.0.RELEASE\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64.17.0.4.v20220903-1038\jre\bin\ja
:: Spring Boot ::
(v2.7.4)
2022-10-23 19:21:26.354 INFO 11928 --- [main] c.p.jeep.controller.CreateOrderTest : Start
2022-10-23 19:21:26.354 INFO 11928 --- [main] c.p.jeep.controller.CreateOrderTest : The f
2022-10-23 19:21:26.643 INFO 11928 --- [main] .s.d.r.c.RepositoryConfigurationDelegate : Multi
```



```
1 package com.promineotech.jeepp;
2
3 import org.springframework.boot.SpringApplication;
4
5
6
7 @SpringBootApplication(scanBasePackages = {"com.promineotech"})
8 @Slf4j
9 public class JeepSales {
10
11     public static void main(String[] args) {
12         SpringApplication.run(JeepSales.class, args);
13     }
14 }
15
16
```



```
1 package com.promineotech.jeepp;
2
3 import org.springframework.boot.SpringApplication;
4
5
6
7 @SpringBootApplication(scanBasePackages = {"com.promineotech"})
8 @Slf4j
9 public class JeepSales {
10
11     public static void main(String[] args) {
12         SpringApplication.run(JeepSales.class, args);
13     }
14 }
15
16
```

URL to GitHub Repository:

[Zbekele2022/Week-16-Coding-Assignment \(github.com\)](https://github.com/Zbekele2022/Week-16-Coding-Assignment)

