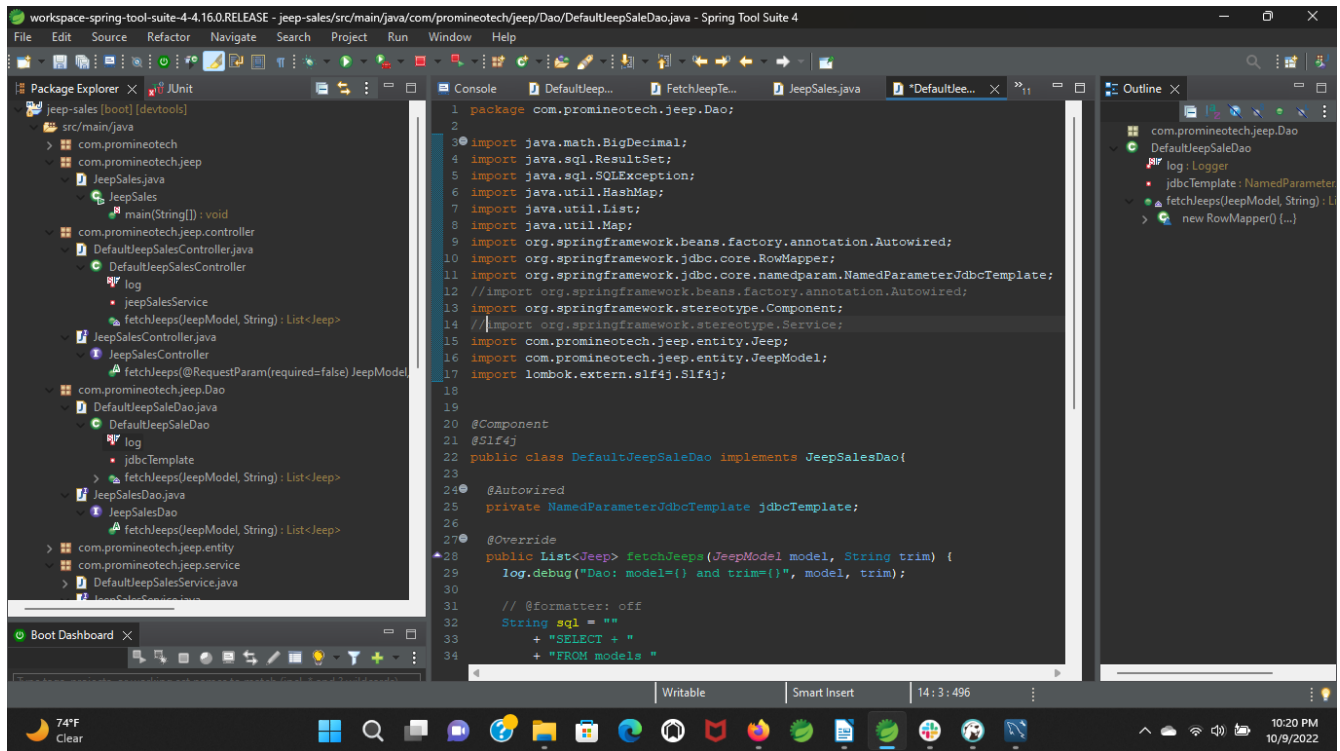
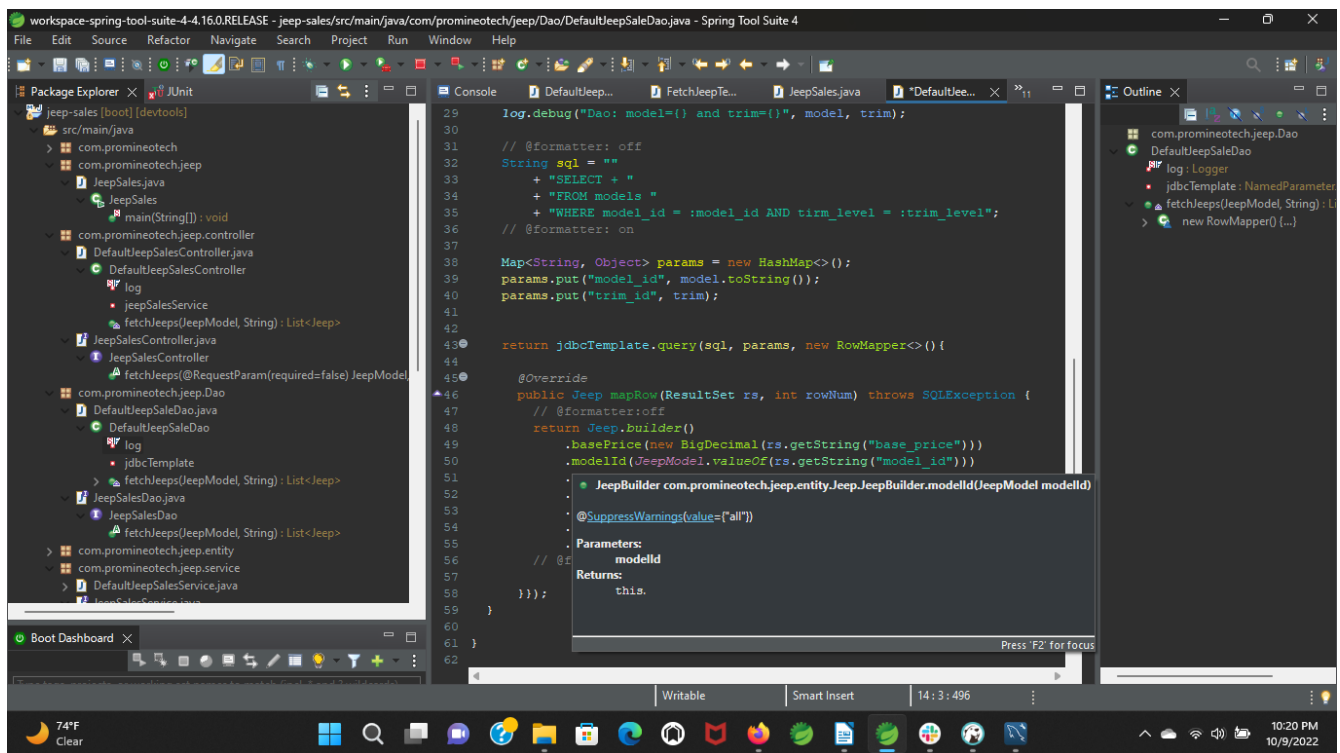


## Solution-code



```
1 package com.promineotech.jeep.Dao;
2
3 import java.math.BigDecimal;
4 import java.sql.ResultSet;
5 import java.sql.SQLException;
6 import java.util.HashMap;
7 import java.util.List;
8 import java.util.Map;
9 import org.springframework.beans.factory.annotation.Autowired;
10 import org.springframework.jdbc.core.RowMapper;
11 import org.springframework.jdbc.core.namedparam.NamedParameterJdbcTemplate;
12 //import org.springframework.beans.factory.annotation.Autowired;
13 import org.springframework.stereotype.Component;
14 //import org.springframework.stereotype.Service;
15 import com.promineotech.jeep.entity.Jeep;
16 import com.promineotech.jeep.entity.JeepModel;
17 import lombok.extern.slf4j.Slf4j;
18
19
20 @Component
21 @Slf4j
22 public class DefaultJeepSaleDao implements JeepSalesDao{
23
24     @Autowired
25     private NamedParameterJdbcTemplate jdbcTemplate;
26
27     @Override
28     public List<Jeep> fetchJeeps(JeepModel model, String trim) {
29         log.debug("Dao: model={} and trim={}", model, trim);
30
31         // @formatter: off
32         String sql = ""
33             + "SELECT + "
34             + "FROM models + "
35             + "WHERE model_id = :model_id AND trim_level = :trim_level";
36         // @formatter: on
37
38         Map<String, Object> params = new HashMap<>();
39         params.put("model_id", model.toString());
40         params.put("trim_id", trim);
41
42         return jdbcTemplate.query(sql, params, new RowMapper<>() {
43
44             @Override
45             public Jeep mapRow(ResultSet rs, int rowNum) throws SQLException {
46                 // @formatter:off
47                 return Jeep.builder()
48                     .basePrice(new BigDecimal(rs.getString("base_price")))
49                     .modelId(JeepModel.valueOf(rs.getString("model_id")))
50                     .JeepBuilder com.promineotech.jeep.entity.Jeep.JeepBuilder.modelId(JeepModel modelId)
51                     .SuppressWarnings(value="all")
52                     .Parameters:
53                     modelId
54                     Returns:
55                     this.
56             }
57         });
58     }
59
60 }
61
62 }
```



```
29 log.debug("Dao: model={} and trim={}", model, trim);
30
31 // @formatter: off
32 String sql = ""
33     + "SELECT + "
34     + "FROM models + "
35     + "WHERE model_id = :model_id AND trim_level = :trim_level";
36 // @formatter: on
37
38 Map<String, Object> params = new HashMap<>();
39 params.put("model_id", model.toString());
40 params.put("trim_id", trim);
41
42 return jdbcTemplate.query(sql, params, new RowMapper<>() {
43
44     @Override
45     public Jeep mapRow(ResultSet rs, int rowNum) throws SQLException {
46         // @formatter:off
47         return Jeep.builder()
48             .basePrice(new BigDecimal(rs.getString("base_price")))
49             .modelId(JeepModel.valueOf(rs.getString("model_id")))
50             .JeepBuilder com.promineotech.jeep.entity.Jeep.JeepBuilder.modelId(JeepModel modelId)
51             .SuppressWarnings(value="all")
52             .Parameters:
53             modelId
54             Returns:
55             this.
56     }
57 });
58
59 }
60
61 }
62 }
```

## Running -App

The screenshot shows the Spring Tool Suite 4 IDE with the file `DefaultJeepSaleDao.java` open. The JUnit test runner on the left indicates a successful run: "Finished after 5.38 seconds", "Runs: 1/1", "Errors: 0", and "Failures: 0". The code in the editor is as follows:

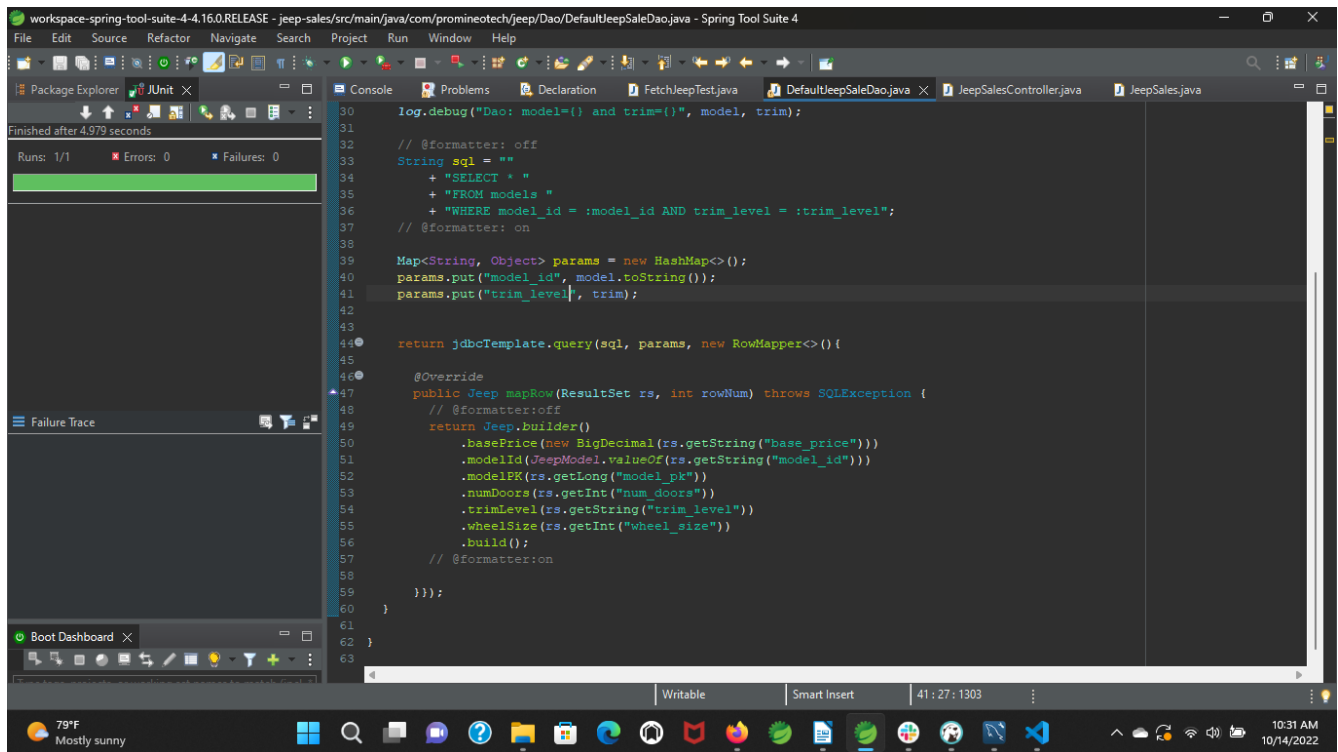
```
1 package com.promineotech.jeep.Dao;
2
3 import java.math.BigDecimal;
4
5 @Service
6 @Slf4j
7 public class DefaultJeepSaleDao implements JeepSalesDao {
8
9     @Autowired
10     private NamedParameterJdbcTemplate jdbcTemplate;
11
12     @Override
13     public List<Jeep> fetchJeeps(JeepModel model, String trim) {
14         log.debug("Dao: model={} and trim={}", model, trim);
15
16         // @formatter: off
17         String sql = "
18             + "SELECT * "
19             + "FROM models "
20             + "WHERE model_id = :model_id AND trim_level = :trim_level";
21         // @formatter: on
22
23         Map<String, Object> params = new HashMap<>();
24         params.put("model_id", model.toString());
25         params.put("trim_level", trim);
26
27         return jdbcTemplate.query(sql, params, new RowMapper<>() {
28
29             @Override
30             public Jeep mapRow(ResultSet rs, int rowNum) throws SQLException {
31                 // @formatter:off
32                 return Jeep.builder()
33                     .model_id(rs.getInt("model_id"))
34                     .trim_level(rs.getInt("trim_level"))
35                     .price(rs.getBigDecimal("price"))
36                     .build();
37             }
38         });
39     }
40 }
```

The bottom status bar shows the system temperature as 75°F Sunny and the time as 9:38 AM on 10/14/2022.

The screenshot shows the Spring Tool Suite 4 IDE with the file `DefaultJeepSaleDao.java` open. The JUnit test runner on the left indicates a successful run: "Finished after 4.979 seconds", "Runs: 1/1", "Errors: 0", and "Failures: 0". The code in the editor is as follows:

```
1 package com.promineotech.jeep.Dao;
2
3 import java.math.BigDecimal;
4
5 @Service
6 @Slf4j
7 public class DefaultJeepSaleDao implements JeepSalesDao {
8
9     @Autowired
10     private NamedParameterJdbcTemplate jdbcTemplate;
11
12     @Override
13     public List<Jeep> fetchJeeps(JeepModel model, String trim) {
14         log.debug("Dao: model={} and trim={}", model, trim);
15
16         // @formatter: off
17         String sql = "
18             + "SELECT * "
19             + "FROM models "
20             + "WHERE model_id = :model_id AND trim_level = :trim_level";
21         // @formatter: on
22
23         Map<String, Object> params = new HashMap<>();
24         params.put("model_id", model.toString());
25         params.put("trim_level", trim);
26
27         return jdbcTemplate.query(sql, params, new RowMapper<>() {
28
29             @Override
30             public Jeep mapRow(ResultSet rs, int rowNum) throws SQLException {
31                 // @formatter:off
32                 return Jeep.builder()
33                     .model_id(rs.getInt("model_id"))
34                     .trim_level(rs.getInt("trim_level"))
35                     .price(rs.getBigDecimal("price"))
36                     .build();
37             }
38         });
39     }
40 }
```

The bottom status bar shows the system temperature as 79°F Mostly sunny and the time as 10:31 AM on 10/14/2022.



Github-Address

<https://github.com/kevinkhalili/Demo>

