```xml
=====================pom.xml=========================
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    https://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>2.3.0.RELEASE</version>
        <relativePath/> <!-- lookup parent from repository -->
    </parent>
    <groupId>com.ameya</groupId>
    <artifactId>003-course-api-jdbc</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>003-course-api-jdbc</name>
    <description>Demo project for Spring Boot</description>
    <properties>
        <java.version>1.8</java.version>
    </properties>
    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-jdbc</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>

        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
            <scope>runtime</scope>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
            <exclusions>
                <exclusion>
                    <groupId>org.junit.vintage</groupId>
                    <artifactId>junit-vintage-engine</artifactId>
                </exclusion>
            </exclusions>
        </dependency>
    </dependencies>
```

```
47
48      <build>
49          <plugins>
50              <plugin>
51                  <groupId>org.springframework.boot</groupId>
52                  <artifactId>spring-boot-maven-plugin</artifactId>
53              </plugin>
54          </plugins>
55      </build>
56
57  </project>
58  ==========================application.properties=========================
    ======
59  server.port=9001
60  logging.level.web=trace
61
62  #DB Properties
63  spring.datasource.url=jdbc:mysql://localhost:3306/nineleapsdb
64  spring.datasource.username=root
65  spring.datasource.password=root
66  spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
67  #spring.datasource.initialization-mode=always
68  ===========================schema.sql=================================
    =======
69  CREATE TABLE topic(id INT(5) NOT NULL, topicname VARCHAR(50) NOT NULL,
    topicdesc VARCHAR(100) NOT NULL, PRIMARY KEY (id));
70  ===========================data.sql=================================
    ========
71  INSERT INTO topic VALUES(101,'JavaSE','Introduction To Core Java 3 Days');
72  INSERT INTO topic VALUES(102,'JavaEE','Introduction To JavaEE 3 Days');
73  =============================Topic.java==========================
    =========
74  package com.ameya.models;
75
76  public class Topic {
77
78      private int id;
79      private String topicName;
80      private String topicDesc;
81      public Topic() {}
82      public Topic(int id, String topicName, String topicDesc) {
83          this.id = id;
84          this.topicName = topicName;
85          this.topicDesc = topicDesc;
86      }
87      public int getId() {
88          return id;
89      }
```

```java
 90     public void setId(int id) {
 91         this.id = id;
 92     }
 93     public String getTopicName() {
 94         return topicName;
 95     }
 96     public void setTopicName(String topicName) {
 97         this.topicName = topicName;
 98     }
 99     public String getTopicDesc() {
100         return topicDesc;
101     }
102     public void setTopicDesc(String topicDesc) {
103         this.topicDesc = topicDesc;
104     }
105     @Override
106     public String toString() {
107         return "Topic [id=" + id + ", topicName=" + topicName + ", topicDesc=" +
               topicDesc + "]\n";
108     }
109 }
```

110 ==TopicDAO.java==

```java
111 package com.ameya.daos;
112
113 import java.util.List;
114
115 import com.ameya.models.Topic;
116
117 public interface TopicDAO {
118     List<Topic> getAllTopics();
119     void addTopic(Topic topic);
120     void updateTopic(Topic topic);
121     Topic getTopic(int id);
122     void deleteTopic(int id);
123 }
```

124 ==TopicRowMapper.java==

```java
125 package com.ameya.daos.util;
126
127 import java.sql.ResultSet;
128 import java.sql.SQLException;
129
130 import org.springframework.jdbc.core.RowMapper;
131
132 import com.ameya.models.Topic;
133
134 public class TopicRowMapper implements RowMapper<Topic> {
```

```java
135
136        @Override
137        public Topic mapRow(ResultSet rs, int rowNum) throws SQLException {
138            Topic topic=new Topic();
139            topic.setId(rs.getInt("id"));
140            topic.setTopicName(rs.getString("topicname"));
141            topic.setTopicDesc(rs.getString("topicdesc"));
142            return topic;
143        }
144
145    }
146    ======================================TopicDAOImpl.java===========
       ================
147    package com.ameya.daos.impl;
148
149    import java.util.List;
150
151    import org.springframework.beans.factory.annotation.Autowired;
152    import org.springframework.jdbc.core.JdbcTemplate;
153    import org.springframework.stereotype.Repository;
154
155    import com.ameya.daos.TopicDAO;
156    import com.ameya.daos.util.TopicRowMapper;
157    import com.ameya.models.Topic;
158
159    @Repository
160    public class TopicDAOImpl implements TopicDAO {
161
162        @Autowired
163        private JdbcTemplate jdbcTemplate;
164
165        @Override
166        public List<Topic> getAllTopics() {
167            final String SQL="select * from topic";
168            List<Topic> topics=jdbcTemplate.query(SQL,new TopicRowMapper());
169            return topics;
170        }
171
172        @Override
173        public void addTopic(Topic topic) {
174            final String SQL="insert into topic (id,topicname,topicdesc) values(?,?,?)";
175
               jdbcTemplate.update(SQL,topic.getId(),topic.getTopicName(),topic.getTopicDesc
               ());
176            System.out.println("+++++ Added To DB - Topic +++++");
177        }
178
179        @Override
```

```java
180    public void updateTopic(Topic topic) {
181        final String SQL="update topic set topicname = ? , topicdesc = ? where id = ?";
182
           jdbcTemplate.update(SQL,topic.getTopicName(),topic.getTopicDesc(),topic.getId
           ());
183        System.out.println("+++++ Updated The DB - Topic +++++");
184    }
185
186    @Override
187    public Topic getTopic(int id) {
188        final String SQL="select * from topic where id = ?";
189        Topic topic=jdbcTemplate.queryForObject(SQL,new Object[] {id},new
           TopicRowMapper());
190        return topic;
191    }
192
193    @Override
194    public void deleteTopic(int id) {
195        final String SQL="delete from topic where id = ?";
196        jdbcTemplate.update(SQL,id);
197        System.out.println("+++++ Deleted From DB - Topic +++++");
198    }
199
200 }
201 ==================================TopicService.java====================
    ========
202 package com.ameya.services;
203
204 import java.util.List;
205
206 import com.ameya.models.Topic;
207
208 public interface TopicService {
209
210    List<Topic> findAll();
211    void createTopic(Topic topic);
212    void updateTopic(Topic topic);
213    Topic findById(int id);
214    void deleteTopic(int id);
215 }
216 ==================================TopicServiceImpl.java==============
    ==========
217 package com.ameya.services.impl;
218
219 import java.util.List;
220
221 import org.springframework.beans.factory.annotation.Autowired;
222 import org.springframework.stereotype.Service;
```

```java
import com.ameya.daos.TopicDAO;
import com.ameya.models.Topic;
import com.ameya.services.TopicService;

@Service
public class TopicServiceImpl implements TopicService {

    @Autowired
    private TopicDAO topicDao;

    @Override
    public List<Topic> findAll() {
        return topicDao.getAllTopics();
    }

    @Override
    public void createTopic(Topic topic) {
        topicDao.addTopic(topic);

    }

    @Override
    public void updateTopic(Topic topic) {
    topicDao.updateTopic(topic);

    }

    @Override
    public Topic findById(int id) {
        return topicDao.getTopic(id);
    }

    @Override
    public void deleteTopic(int id) {
        topicDao.deleteTopic(id);

    }

}
==================================TopicController.java==================================
package com.ameya.controllers;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.DeleteMapping;
```

```java
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.PutMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RestController;

import com.ameya.models.Topic;
import com.ameya.services.TopicService;

@RestController
public class TopicController {

    @Autowired
    private TopicService topicService;

    @GetMapping(path="/topics")
    public List<Topic> findAll(){
        return topicService.findAll();
    }
    @GetMapping(path="/topics/{id}")
    public Topic findById(@PathVariable("id") int id) {
        return topicService.findById(id);
    }
    @PostMapping(path="/topics/create")
    public void create(@RequestBody Topic topic) {
        topicService.createTopic(topic);
    }
    @PutMapping(path="/topics/update")
    public void modify(@RequestBody Topic topic) {
        topicService.updateTopic(topic);
    }
    @DeleteMapping(path="/topics/delete/{id}")
    public void delete(@PathVariable("id") int id) {
        topicService.deleteTopic(id);
    }
}
```
===
Sample URLs

GET : http://localhost:9001/topics/
POST : http://localhost:9001/topics/create

{
     "id": 103,
     "topicName": "Spring",
     "topicDesc": "Introduction To Spring 3 Days"

```
318  }
319  PUT : http://localhost:9001/topics/update
320
321  {
322          "id": 102,
323          "topicName": "EJB",
324          "topicDesc": "Introduction To EJB 3 Days"
325  }
326  GET : http://localhost:9001/topics/101
327  Delete : http://localhost:9001/topics/102
328
```