```
===============================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>

    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
            </plugin>
        </plugins>
    </build>

</project>
===============================application.properties============================
server.port=9001
logging.level.web=trace

spring.datasource.url=jdbc:mysql://localhost:3306/nineleapsdb
spring.datasource.username=root
spring.datasource.password=root
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
#spring.datasource.initialization-mode=always
================================schema.sql========================================
CREATE TABLE topic(id INT(11) NOT NULL, topicname VARCHAR(50) NOT NULL, topicdesc
VARCHAR(100) NOT NULL, PRIMARY KEY (id));
===================================data.sql=======================================
insert into topic values(101,'JavaSE','Introduction To Core Java 3 Days');
insert into topic values(102,'JavaEE','Introduction To JavaEE 3 Days');
===============================Topic.java========================================
package com.ameya.models;

public class Topic {

    private int id;
    private String topicName;
    private String topicDesc;
    public Topic() {

    }
    public Topic(int id, String topicName, String topicDesc) {
        super();
        this.id = id;
        this.topicName = topicName;
        this.topicDesc = topicDesc;
    }
    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getTopicName() {
        return topicName;
    }
    public void setTopicName(String topicName) {
        this.topicName = topicName;
    }
    public String getTopicDesc() {
        return topicDesc;
    }
    public void setTopicDesc(String topicDesc) {
        this.topicDesc = topicDesc;
    }

}
=====================================TopicDAO.java================================
package com.ameya.daos;

import java.util.List;

import com.ameya.models.Topic;

public interface TopicDAO {

    List<Topic> getAllTopics();
    void addTopic(Topic topic);
    void updateTopic(Topic topic);
    Topic getTopic(int id);
    void deleteTopic(int id);
}
======================================TopicDAOImpl.java==========================
package com.ameya.daos.impl;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.jdbc.core.JdbcTemplate;
import org.springframework.stereotype.Repository;

import com.ameya.daos.TopicDAO;
import com.ameya.daos.util.TopicRowMapper;
import com.ameya.models.Topic;

@Repository
public class TopicDAOImpl implements TopicDAO {

    @Autowired
    private JdbcTemplate jdbcTemplate;

    @Override
    public List<Topic> getAllTopics() {
        final String SQL="select * from topic";
        List<Topic> topics=jdbcTemplate.query(SQL,new TopicRowMapper());
        return topics;
    }

    @Override
    public void addTopic(Topic topic) {
        final String SQL="insert into topic (id,topicname,topicdesc)"
                + " values (?,?,?)";
        jdbcTemplate.update(SQL,topic.getId(),topic.getTopicName(),
                topic.getTopicDesc());
        System.out.println("++++ Topic Added To DB ++++");

    }

    @Override
    public void updateTopic(Topic topic) {
        final String SQL="update topic set topicname = ? , "
                + "topicdesc = ? where id = ?";
        jdbcTemplate.update(SQL,topic.getTopicName(),
                topic.getTopicDesc(),topic.getId());
        System.out.println("++++ Topic Updated ++++");

    }

    @Override
    public Topic getTopic(int id) {
        final String SQL="select * from topic where id = ?";
        Topic topic=jdbcTemplate.queryForObject(SQL,new Object[] {id},
                new TopicRowMapper());
        return topic;
    }

    @Override
    public void deleteTopic(int id) {
        final String SQL="delete from topic where id = ?";
        jdbcTemplate.update(SQL,id);
        System.out.println("++++ Topic Deleted ++++");

    }

}
======================================TopicService.java======================
package com.ameya.services;

import java.util.List;

import com.ameya.models.Topic;

public interface TopicService {
    List<Topic> getAllTopics();
    void addTopic(Topic topic);
    void updateTopic(Topic topic);
    Topic getTopic(int id);
    void deleteTopic(int id);
}
=====================================TopicServiceImpl.java======================
package com.ameya.services.impl;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

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> getAllTopics() {
        return topicDao.getAllTopics();
    }

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

    }

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

    }

    @Override
    public Topic getTopic(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;
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.getAllTopics();
    }
    @GetMapping(path="/topics/{id}")
    public Topic findById(@PathVariable("id") int id) {
        return topicService.getTopic(id);
    }
    @PostMapping(path="/topics/create")
    public void create(@RequestBody Topic topic) {
        topicService.addTopic(topic);
    }
    @PutMapping(path="/topics/update/{id}")
    public void modify(@RequestBody Topic topic) {
        topicService.updateTopic(topic);
    }
    @DeleteMapping(path="/topics/delete/{id}")
    public void delete(@PathVariable("id") int id) {
        topicService.deleteTopic(id);
    }
```

```
}
======================================================================

http://localhost:9001/topics/
http://localhost:9001/topics/create

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