

```

1  =====Topic.java=====
2  package com.ameya.models;
3
4  public class Topic {
5
6      private int id;
7      private String topicName;
8      private String topicDesc;
9      public Topic() {
10
11      }
12      public Topic(int id, String topicName, String topicDesc) {
13          super();
14          this.id = id;
15          this.topicName = topicName;
16          this.topicDesc = topicDesc;
17      }
18      public int getId() {
19          return id;
20      }
21      public void setId(int id) {
22          this.id = id;
23      }
24      public String getTopicName() {
25          return topicName;
26      }
27      public void setTopicName(String topicName) {
28          this.topicName = topicName;
29      }
30      public String getTopicDesc() {
31          return topicDesc;
32      }
33      public void setTopicDesc(String topicDesc) {
34          this.topicDesc = topicDesc;
35      }
36      @Override
37      public String toString() {
38          return "Topic [id=" + id + ", topicName=" + topicName + ", topicDesc=" +
39              topicDesc + "]";
40
41      }
42  }
43  =====TestRestTemplate.java=====
44  ==
45  package com.ameya.test;
46
47  import java.util.HashMap;
48  import java.util.List;
49  import java.util.Map;
50
51  import org.springframework.core.ParameterizedTypeReference;
52  import org.springframework.http.HttpHeaders;
53  import org.springframework.http.HttpMethod;
54  import org.springframework.http.ResponseEntity;
55  import org.springframework.web.client.RestTemplate;
56
57  import com.ameya.models.Topic;
58
59  public class TestRestTemplate {
60
61      public static void main(String[] args) {
62
63          final String getByIdUri="http://localhost:9001/topics/{id}";
64          final String postUri="http://localhost:9001/topics/create";
65          final String getAllUri="http://localhost:9001/topics";
66
67          RestTemplate restTemplate=new RestTemplate();
68          Map<String, Integer> params=new HashMap<>();
69          params.put("id", 103);
70          String strTopic=restTemplate.getForObject(getByIdUri,String.class,params);
71          System.out.println("String => "+strTopic);
72          System.out.println("-----");

```

```

72 Topic topic=restTemplate.getForObject(getByIdUri,Topic.class,params);
73 System.out.println("Topic => "+topic);
74 System.out.println("-----");
75 Topic newTopic=new Topic(105,"Spring Boot with Spring Cloud",
76     "A Comprehensive 3 Day Course");
77 String postResult=restTemplate.postForObject(postUri, newTopic, String.class);
78 System.out.println("Result => "+postResult);
79 System.out.println("-----");
80 ResponseEntity<List<Topic>> response=restTemplate.exchange(
81     getAllUri,
82     HttpMethod.GET,
83     null,
84     new ParameterizedTypeReference<List<Topic>>() {}
85 );
86 List<Topic>topics=response.getBody();
87 for(Topic t : topics) {
88     System.out.println(t);
89 }
90 System.out.println("-----");
91 ResponseEntity<Topic> topicResponse=restTemplate.getForEntity(getByIdUri,
92     Topic.class,params);
93 Topic topic1=topicResponse.getBody();
94 System.out.println(topic1);
95
96 HttpHeaders headers1=response.getHeaders();
97 System.out.println("headers1 =>"+headers1);
98
99 HttpHeaders headers2=topicResponse.getHeaders();
100 System.out.println("headers2 =>"+headers2);
101 }
102 }
103
104

```