Date : 30/03/2021
Spring Boot 9AM
Mr. RAGHU

RestTemplateCode:
https://www.mediafire.com/file/c9lpbmo44nok3e8/SpringBoot9AM_RESTTEMPLATE_30032021.zip/file

All PDFS:
https://www.mediafire.com/file/w5x9w5vcmkwkkdv/RaghuSirNareshITJavaPdfs.zip/file

Log4J:
https://www.youtube.com/watch?v=mblGoKU1aKo
https://www.youtube.com/watch?v=5oLfHiP_iJc

ELK Stack:
https://www.youtube.com/watch?v=uSYExRWbC9Y

JDBC:
https://www.youtube.com/c/NareshIT/search?query=JDBC%20raghu

RestTemplate (Consumer Code)

=> RT has given one global method exchange() used to make any type of HTTP call (GET/POST/PUT/DELETE) that finally returns ResponseEntity<T>

*) exchange(
url,
HttpMethod,
HttpEntity,
ResponseType,
pathVatiables): ResponseEntity<T>

=> HttpMethod it is a enum having values like GET, POST, PUT, DELETE
=> HttpEntity = request Body + request Headers.
It can be null for GET type (No Header and Body case)
=> ResponseType can be String, ClassType and Collection/Array

======Producer code==========

```
package in.nareshit.raghu.rest;

import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import in.nareshit.raghu.model.Student;

@RestController
@RequestMapping("/std")
public class StudentRestController {
```

```
	@PostMapping("/create")
	public ResponseEntity<String> createStudent(
			@RequestBody Student student
			)
	{
		return ResponseEntity.ok("Student data is " +
student);
	}
}
==============Consumer code==========================
package in.nareshit.raghu.runner;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.CommandLineRunner;
import org.springframework.http.HttpEntity;
import org.springframework.http.HttpHeaders;
import org.springframework.http.HttpMethod;
import org.springframework.http.MediaType;
import org.springframework.http.ResponseEntity;
import org.springframework.stereotype.Component;
import org.springframework.web.client.RestTemplate;

@Component
public class RestConsumerExchangePost
		implements CommandLineRunner
{

	private static final Logger LOG =
LoggerFactory.getLogger(RestConsumerExchangePost.class);

	@Autowired
	private RestTemplate rt;

	public void run(String... args) throws Exception {
		//1. Define URL
		String url ="http://localhost:8080/std/create";

		//2. HttpEntity=header+body
		String body ="
{\"stdId\":100,\"stdName\":\"A\",\"stdFee\":300.0}";

		HttpHeaders headers = new HttpHeaders();
		headers.setContentType(MediaType.APPLICATION_JSON);

		HttpEntity<String> requestEntity = new
HttpEntity<String>(body, headers);

		//3. make request and get response
		//URL, HttpMethod, HttpEntity , responseType,
PathVariables
		ResponseEntity<String> resp = rt.exchange(url,
HttpMethod.POST, requestEntity, String.class);


		//4. print details
```

```
                LOG.info("Status ID {}", resp.getStatusCodeValue());
                LOG.info("Status CODE {}",
resp.getStatusCode().name());
                LOG.info("Response Body {}", resp.getBody());
                LOG.info("Response Headers {}", resp.getHeaders());
                //5. Stop server manually
                System.exit(0);
        }
}
```

==========

put() and delete() methods makes HTTP call without any issues, but no response is taken back to consumer app. ie returns void type.

----------Example----------

*) Producer

```
package in.nareshit.raghu.rest;

import org.springframework.http.ResponseEntity;
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.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import in.nareshit.raghu.model.Student;

@RestController
@RequestMapping("/std")
public class StudentRestController {


        @PutMapping("/modify")
        public ResponseEntity<String> updateStudent(
                        @RequestBody Student student
                        )
        {
                System.out.println("EXECUTED.........");
                return ResponseEntity.ok("Student data is " +
student);
        }

        @DeleteMapping("/remove/{id}")
        public ResponseEntity<String> removeStudent(
                        @PathVariable Integer id
                        )
        {
                return ResponseEntity.ok("Student removed " + id);
        }



}
```

*) consumer

```
package in.nareshit.raghu.runner;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.CommandLineRunner;
import org.springframework.http.HttpEntity;
import org.springframework.http.HttpHeaders;
import org.springframework.http.HttpMethod;
import org.springframework.http.MediaType;
import org.springframework.http.ResponseEntity;
import org.springframework.stereotype.Component;
import org.springframework.web.client.RestTemplate;

//@Component
public class RestConsumerPutEx implements CommandLineRunner {

        private static final Logger LOG =
LoggerFactory.getLogger(RestConsumerPutEx.class);

        @Autowired
        private RestTemplate rt;

        public void run(String... args) throws Exception {
                //1. Define URL
                String url ="http://localhost:8080/std/modify";

                //2. creating HttpEntity(requestEntity)
                HttpHeaders headers = new HttpHeaders();
                headers.setContentType(MediaType.APPLICATION_JSON);

                String body ="{}"; //valid JSON

                HttpEntity<String> requestEntity = new
HttpEntity<String>(body, headers);

                //3. make HTTP call
                //rt.put(url, requestEntity);
                ResponseEntity<String> resp =  rt.exchange(url,
HttpMethod.PUT, requestEntity, String.class);

                //4. print details
                LOG.info("Status data
{}",resp.getStatusCode().name());
                LOG.info("Status code {}",resp.getStatusCodeValue());
                LOG.info("Body {}",resp.getBody());

                //5. stop server
                System.exit(0);
        }

}

-----------
package in.nareshit.raghu.runner;

import org.slf4j.Logger;
```

```
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.CommandLineRunner;
import org.springframework.http.HttpMethod;
import org.springframework.http.ResponseEntity;
import org.springframework.stereotype.Component;
import org.springframework.web.client.RestTemplate;

@Component
public class RestConsumerDelete implements CommandLineRunner {

        private static final Logger LOG =
LoggerFactory.getLogger(RestConsumerDelete.class);

        @Autowired
        private RestTemplate rt;

        public void run(String... args) throws Exception {
                //1. Define URL
                String url ="http://localhost:8080/std/remove/{id}";

                //2. creating HttpEntity(requestEntity)
                //no head + body

                //3. make HTTP call
                ResponseEntity<String> resp =  rt.exchange(url,
HttpMethod.DELETE, null, String.class,101);

                //4. print details
                LOG.info("Status data
{}",resp.getStatusCode().name());
                LOG.info("Status code {}",resp.getStatusCodeValue());
                LOG.info("Body {}",resp.getBody());

                //5. stop server
                System.exit(0);
        }

}
==========================================================================
==
        Auto-Type Conversion Support for ResponseType(Class<T>)

*) If Producer returns String type output then consumer can read
   same into String object.

   Else if Producer returns complex Type Class, List/Set
   then even consumer methods getForEntity(), postForEntity(),
   exchange() ..mehthods support auto-type conversion of
   response data ie JSON--> ClassType/Array

--Exmple code -----------

Ex#1
Producer : ResponseEntity<String>
Consumer :  getForEntity(url,String.class)
```

Ex#2
Producer : ResponseEntity<Employee>
Consumer : getForEntity(url,Employee.class) //JSON-->Employee
 ** we can read plain JSON
    getForEntity(url,String.class) //print as it is JSON

Ex#3
Producer : ResponseEntity<List<Employee>>
Consumer : getForEntity(url,Employee[].class) //JSON->Employee Array
 ** we can read plain JSON
    getForEntity(url,String.class) //print as it is JSON

========code================
1. Producer
```
package in.nareshit.raghu.rest;

@RestController
@RequestMapping("/std")
public class StudentRestController {


        @GetMapping("/one/{id}")
        public ResponseEntity<Student> getOneStudent(
                        @PathVariable Integer id
                        )
        {
                return ResponseEntity.ok(new Student(id, "TEST",
200.0));
        }

        @GetMapping("/all")
        public ResponseEntity<List<Student>> getAllStudent()
        {
                return ResponseEntity.ok(
                                Arrays.asList(
                                                new
Student(110, "SAM", 250.0),
                                                new
Student(112, "SYED", 252.0),
                                                new
Student(113, "RAM", 251.0)
                                                )

                                );
        }


}
```
2. Consumer
```
package in.nareshit.raghu.runner;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.CommandLineRunner;
import org.springframework.http.ResponseEntity;
import org.springframework.web.client.RestTemplate;
```

```
import in.nareshit.raghu.model.Student;

//@Component
public class RestConsumerGetClassType implements CommandLineRunner {

	private static final Logger LOG =
LoggerFactory.getLogger(RestConsumerGetClassType.class);

	@Autowired
	private RestTemplate rt;

	public void run(String... args) throws Exception {
		//1. Define URL of Provider
		String url = "http://localhost:8080/std/one/{id}";

		//2. Make call and get Response
		//ResponseEntity<String>  resp = rt.getForEntity(url,
String.class,550);
		ResponseEntity<Student>  resp = rt.getForEntity(url,
Student.class,550);

		//3. print details

		LOG.info("Status ID {}", resp.getStatusCodeValue());
		LOG.info("Status CODE {}",
resp.getStatusCode().name());
		LOG.info("Response Body {}", resp.getBody());
		LOG.info("Response Headers {}", resp.getHeaders());
		//4. Stop server manually
		System.exit(0);
	}

}
---------------------------
package in.nareshit.raghu.runner;

import java.util.Arrays;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.CommandLineRunner;
import org.springframework.http.ResponseEntity;
import org.springframework.stereotype.Component;
import org.springframework.web.client.RestTemplate;

import in.nareshit.raghu.model.Student;

@Component
public class RestConsumerGetClassTypeAll implements CommandLineRunner
{

	private static final Logger LOG =
LoggerFactory.getLogger(RestConsumerGetClassTypeAll.class);

	@Autowired
```

```
        private RestTemplate rt;

        public void run(String... args) throws Exception {
                //1. Define URL of Provider
                String url = "http://localhost:8080/std/all";

                //2. Make call and get Response
                //ResponseEntity<String>  resp = rt.getForEntity(url,
String.class);
                ResponseEntity<Student[]>  resp = rt.getForEntity(url,
Student[].class);

                //3. print details

                LOG.info("Status ID {}", resp.getStatusCodeValue());
                LOG.info("Status CODE {}",
resp.getStatusCode().name());
                LOG.info("Response Body {}",
Arrays.asList(resp.getBody()));
                LOG.info("Response Headers {}", resp.getHeaders());
                //4. Stop server manually
                System.exit(0);
        }

}
=========================================================
Download MongoDB:
1. https://www.mongodb.com/try/download/community
2. Details
  Version : 3.6
  OS      : Windows
  Package : MSI
3. Click on Download
 (mongodb-win32-x86_64-2008plus-ssl-3.6.23-signed.msi)
4. Double click > next > next > Finish
  (uncheck > MongoDB Compass option while installing)

*) SET PATH in ENVIRONMENT VARIABLE:
    C:\Program Files\MongoDB\Server\3.6\bin

*) create folder system in drive (C:/data/db create this folder)
   C:\
      data\
         db\

*) Execute commands
 > mongod  (server)
 > mongo   (client)
 > show dbs;

=======================================================
5. Robo 3T (UI) : https://robomongo.org/
   Downloa and install
```