RestTemplate

**RestTemplate** là một lớp trung tâm trong Spring dành cho các request đồng bộ *(synchronous calls)* bởi *Client* để truy cập vào *RESTful Web Service*. Khi sử dụng lớp nói trên, người dùng chỉ phải cung cấp URL, các tham số và nhận kết quả trả về

### **Nhận về dữ liệu dưới dạng JSON**

- Sử dụng *getForEntity()* để lấy tài nguyên từ request có phương thức **GET**:

*RestTemplate restTemplate = new RestTemplate();*

*String fooResourceUrl = "http://localhost:8080/spring-rest/foos";*

*ResponseEntity<String> response = restTemplate.getForEntity(fooResourceUrl + "/1", String.class);*

*assertThat(response.getStatusCode(), equalTo(HttpStatus.OK));*

Chú ý rằng chúng ta có quyền truy cập vào *response* để kiểm tra kết quả trả về

*ObjectMapper mapper = new ObjectMapper();*

*JsonNode root = mapper.readTree(response.getBody());*

*JsonNode name = root.path("name");*

*assertThat(name.asText(), notNullValue());*

### **Nhận về dữ liệu dưới dạng Object**

Sử dụng *getForObject()* của RestTemplate để lấy dữ liệu

Cấu trúc phần data kế thừa serializable

*public class* ***Foo*** *implements* ***Serializable*** *{*

*private long id;*

*private String name;*

*// standard getters and setters*

*}*

Sử dụng *getForObject()*

*Foo foo = restTemplate.getForObject(fooResourceUrl + "/1", Foo.class);*

*assertThat(foo.getName(), notNullValue());*

*assertThat(foo.getId(), is(1L));*

## **Sử dụng Head để lấy dữ liệu từ Header**

Lấy ra headers dạng *HttpHeaders*

*HttpHeaders httpHeaders = restTemplate.headForHeaders(fooResourceUrl);*

*assertTrue(httpHeaders.getContentType().includes(MediaType.APPLICATION\_JSON));*

## **Sử dụng Post để tạo tài nguyên**

Để tạo mới một tài nguyên trong API, chúng ta có thể sử dụng phương thức *postForLocation()*, *postForObject()* hoặc là *postForEntity()*.

**Phương thức *postForObject()***

*RestTemplate restTemplate = new RestTemplate();*

*HttpEntity<Foo> request = new HttpEntity<>(new Foo("bar"));*

*Foo foo = restTemplate.postForObject(fooResourceUrl, request, Foo.class);*

*assertThat(foo, notNullValue());*

*assertThat(foo.getName(), is("bar"));*

**Phương thức *postForLocaltion()***

Tương tự như trên, thao tác nay thay vì trả về tài nguyên đầy đủ thì nó chỉ trả về vị trí của tài nguyên đó sau khi được tạo thành công

*RestTemplate restTemplate = new RestTemplate();*

*HttpEntity<Foo> request = new HttpEntity<>(new Foo("bar"));*

*URI location = restTemplate.postForLocation(fooResourceUrl, request);*

*assertThat(location, notNullValue());*

**Phương thức *exchange()***

*RestTemplate restTemplate = new RestTemplate();*

*HttpEntity<Foo> request = new HttpEntity<>(new Foo("bar"));*

*ResponseEntity<Foo> response = restTemplate.exchange(fooResourceUrl, HttpMethod.POST, request, Foo.class);*

*assertThat(response.getStatusCode(), is(HttpStatus.CREATED));*

*Foo foo = response.getBody();*

*assertThat(foo, notNullValue());*

*assertThat(foo.getName(), is("bar"));*

**Submit một form dữ liệu**

Tiếp theo chúng ta hãy cùng xem cách để submit một form sử dụng phương thức POST

Đầu tiên chúng ta cần phải cài *Content-Type header* là

*HttpHeaders headers = new HttpHeaders();*

*headers.setContentType(MediaType.APPLICATION\_FORM\_URLENCODED);*

Tiếp theo, chúng ta xây dựng một request sử dụng đến

*MultiValueMap<String, String> map= new LinkedMultiValueMap<>();*

*map.add("id", "1");*

*HttpEntity<MultiValueMap<String, String>> request = new HttpEntity<>(map, headers);*

Cuối cùng chúng ta kết nối đến REST service bằng cách gọi phương thức restTemplate.postForEntity()

*ResponseEntity<String> response = restTemplate.postForEntity(*

*fooResourceUrl+"/form", request , String.class);*

*assertThat(response.getStatusCode(), is(HttpStatus.CREATED));*

## **Sử dụng PUT để cập nhật các tài nguyên**

### Ví dụ **PUT** đơn giản với phương thức *exchange()*

*Foo updatedInstance = new Foo("newName");*

*updatedInstance.setId(createResponse.getBody().getId());*

*String resourceUrl = fooResourceUrl + '/' + createResponse.getBody().getId();*

*HttpEntity<Foo> requestUpdate = new HttpEntity<>(updatedInstance, headers);*

*template.exchange(resourceUrl, HttpMethod.PUT, requestUpdate, Void.class);*

### Phương thức **PUT** với *Request Callback*

Hãy đảm bảo rằng chúng ta chuẩn bị *callback*, nơi mà có thể đặt tất cả các *header* của chúng ta cần cũng như nội dung *request*:

*RequestCallback* ***requestCallback****(final Foo updatedInstance) {*

*return clientHttpRequest -> {*

*ObjectMapper mapper = new ObjectMapper();*

*mapper.writeValue(clientHttpRequest.getBody(), updatedInstance);*

*clientHttpRequest.getHeaders().add(HttpHeaders.CONTENT\_TYPE, MediaType.APPLICATION\_JSON\_VALUE);*

*clientHttpRequest.getHeaders().add(HttpHeaders.AUTHORIZATION, "Basic " + getBase64EncodedLogPass());*

*};*

*}*

Tạo một tài nguyên mới với phương thức POST

*ResponseEntity<Foo> response = restTemplate.exchange(fooResourceUrl, HttpMethod.POST, request, Foo.class);*

*assertThat(response.getStatusCode(), is(HttpStatus.CREATED));*

Và sau đó chúng ta cập nhật lại tài nguyên như sau:

*Foo updatedInstance = new Foo("newName");*

*updatedInstance.setId(response.getBody().getId());*

*String resourceUrl = fooResourceUrl + '/' + response.getBody().getId();*

*restTemplate.execute(*

*resourceUrl,*

*HttpMethod.PUT,*

*requestCallback(updatedInstance),*

*clientHttpResponse -> null);*

## **Sử dụng phương thức DELETE để xóa một tài nguyên**

Để xóa một tài nguyên có sẵn chúng ta sử dụng:

*String entityUrl = fooResourceUrl + "/" + existingResource.getId();*

*restTemplate.delete(entityUrl);*

## **Configure Timeout**

Bằng cách sử dụng ClientHttpRequestFactory

*RestTemplate restTemplate = new RestTemplate(getClientHttpRequestFactory());*

*private ClientHttpRequestFactory* ***getClientHttpRequestFactory****() {*

*int timeout = 5000;*

*HttpComponentsClientHttpRequestFactory clientHttpRequestFactory*

*= new HttpComponentsClientHttpRequestFactory();*

*clientHttpRequestFactory.setConnectTimeout(timeout);*

*return clientHttpRequestFactory;*

*}*

Hoặc sử dụng RequestConfig

*private ClientHttpRequestFactory* ***getClientHttpRequestFactory****() {*

*int timeout = 5000;*

*RequestConfig config = RequestConfig.custom()*

*.setConnectTimeout(timeout)*

*.setConnectionRequestTimeout(timeout)*

*.setSocketTimeout(timeout)*

*.build();*

*CloseableHttpClient client = HttpClientBuilder*

*.create()*

*.setDefaultRequestConfig(config)*

*.build();*

*return new HttpComponentsClientHttpRequestFactory(client);*

*}*

*Ref:* [*https://www.baeldung.com/rest-template*](https://www.baeldung.com/rest-template)