[Bài đọc] RestTemplate

Tương tự như nhiều lớp template khác, chẳng hạn như JdbcTemplate và HibernateTemplate, lớp RestTemplate cũng được thiết kế để thực hiện các chức năng phức tạp gọi đến các REST services. Bảng sau tóm tắt các phương thức RestTemplate ánh xạ các phương thức HTTP:

|  |  |  |
| --- | --- | --- |
| **RestTemplate method** | **Phương thức HTTP** | **Description** |
| getForEntity and getForObject | GET | Phương thức này lấy ra các đại diện trên URI được chỉ định |
| postForLocation and postForObject | POST | Phương thức này tạo ra một tài nguyên mới bằng cách đăng đối tượng mới trên vị trí URI được chỉ định và trả về tiêu đề có giá trị như Location |
| put | PUT | Phương thức này tạo hoặc cập nhật tài nguyên tại URI được chỉ định |
| delete | DELETE | Thao tác này sẽ xóa tài nguyên được chỉ định bởi URI |
| optionsForAllow | OPTIONS | Phương thức nào trả về giá trị của các allowed header cho URL được chỉ định |
| execute and exchange | Any | Thực hiện phương thức HTTP và trả về phản hồi ResponseEntity |

RestContollers chỉ xử lý yêu cầu nếu nó được yêu cầu bởi URL đúng. Spring MVC controllers cũng xử lý request là tham số truy vấn(request-parameter) và yêu cầu truy vấn (request-query) trong khi các URL được các RESTful web services xử lý là định hướng tài nguyên. Việc xác định tài nguyên để ánh xạ được thực hiện bởi toàn bộ URL mà không có bất kỳ tham số truy vấn nào.

Các URL được viết dựa trên danh từ số nhiều và cố gắng tránh sử dụng các động từ hoặc tham số truy vấn. Hình dưới đây là URL RESTful cho tài nguyên là sự kết hợp của ngữ cảnh Servlet, danh từ tài nguyên cần tìm nạp và biến đường dẫn:

**Ví dụ về RESTful URLs:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Supported HTTPmethods**  **Resource to fetch** | **GET method** | **POST method** | **PUT method** | **DELETE method** |
| /books | Trả về danh sách sách | Thêm một cuốn sách mới | Cập nhật sách hoặc sách | Thao tác này sẽ xóa nhiều cuốn sách |
| /books/100 | Trả về cuốn sách | Trả về lỗi HTTP 405, là mã lỗi cho phương thức không được phép | Cập nhật sách | Thao tác này sẽ xóa sách |

**Ví dụ sau đây sử dụng các phương thức và URL HTTP khác nhau bằng các Spring RESTful web service như sau:**

public class SpringRestTestClient {  
  
 public static final String *REST\_SERVICE\_URI* = "<a class="vglnk" href="http://localhost:8080/Spring4MVCCRUDRestService" rel="nofollow"><span>http</span><span>://</span><span>localhost</span><span>:</span><span>8080</span><span>/</span><span>Spring4MVCCRUDRestService</span></a>";  
  
 /\* GET \*/  
 @SuppressWarnings("unchecked")  
 private static void listAllUsers(){  
 System.*out*.println("Testing listAllUsers API-----------");  
  
 RestTemplate restTemplate = new RestTemplate();  
 List<LinkedHashMap<String, Object>> usersMap = restTemplate.getForObject(*REST\_SERVICE\_URI*+"/user/", List.class);  
  
 if(usersMap!=null){  
 for(LinkedHashMap<String, Object> map : usersMap){  
 System.*out*.println("User : id="+map.get("id")+", Name="+map.get("name")+", Age="+map.get("age")+", Salary="+map.get("salary"));;  
 }  
 }else{  
 System.*out*.println("No user exist----------");  
 }  
 }  
  
 /\* GET \*/  
 private static void getUser(){  
 System.*out*.println("Testing getUser API----------");  
 RestTemplate restTemplate = new RestTemplate();  
 User user = restTemplate.getForObject(*REST\_SERVICE\_URI*+"/user/1", User.class);  
 System.*out*.println(user);  
 }  
  
 /\* POST \*/  
 private static void createUser() {  
 System.*out*.println("Testing create User API----------");  
 RestTemplate restTemplate = new RestTemplate();  
 User user = new User(0,"Sarah",51,134);  
 URI uri = restTemplate.postForLocation(*REST\_SERVICE\_URI*+"/user/", user, User.class);  
 System.*out*.println("Location : "+uri.toASCIIString());  
 }  
  
 /\* PUT \*/  
 private static void updateUser() {  
 System.*out*.println("Testing update User API----------");  
 RestTemplate restTemplate = new RestTemplate();  
 User user = new User(1,"Tomy",33, 70000);  
 restTemplate.put(*REST\_SERVICE\_URI*+"/user/1", user);  
 System.*out*.println(user);  
 }  
  
 /\* DELETE \*/  
 private static void deleteUser() {  
 System.*out*.println("Testing delete User API----------");  
 RestTemplate restTemplate = new RestTemplate();  
 restTemplate.delete(*REST\_SERVICE\_URI*+"/user/3");  
 }  
  
  
 /\* DELETE \*/  
 private static void deleteAllUsers() {  
 System.*out*.println("Testing all delete Users API----------");  
 RestTemplate restTemplate = new RestTemplate();  
 restTemplate.delete(*REST\_SERVICE\_URI*+"/user/");  
 }  
  
 public static void main(String args[]){  
 *listAllUsers*();  
 *getUser*();  
 *createUser*();  
 *listAllUsers*();  
 *updateUser*();  
 *listAllUsers*();  
 *deleteUser*();  
 *listAllUsers*();  
 *deleteAllUsers*();  
 *listAllUsers*();  
 }  
}

Last modified: Monday, 25 March 2019, 11:23 AM