




Spring Data REST



Tạo các service có mệt không?


. Có!

- 
- `GET /students`: Lấy danh sách tất cả sinh viên.
 - `GET /students/{id}`: Lấy thông tin chi tiết của một sinh viên dựa trên ID.
 - `POST /students`: Thêm một sinh viên mới.
 - `PUT /students/{id}`: Cập nhật thông tin của một sinh viên dựa trên ID.
 - `DELETE /students/{id}`: Xóa một sinh viên dựa trên ID.



Spring Data REST

- Spring Data REST là một phần mở rộng của Spring Data JPA, cung cấp cách dễ dàng để tạo RESTful API từ các repository JPA mà không cần viết thêm mã điều khiển (controller) tùy chỉnh. Nó giúp tự động tạo các điểm cuối RESTful cho các thực thể của bạn và thực hiện các thao tác CRUD cơ bản.



```
<dependency>  
  <groupId>org.springframework.boot</groupId>  
  <artifactId>spring-boot-starter-data-rest</artifactId>  
</dependency>
```



Các bước để sử dụng Spring Data REST

- Đảm bảo StudentRepository của bạn extends từ JpaRepository như đã làm trước đó.
- Không cần phải tạo thêm một controller tùy chỉnh cho Student nữa. Spring Data REST sẽ tự động tạo ra các điểm cuối RESTful cho Student dựa trên repository.
- Chạy ứng dụng và truy cập các điểm cuối RESTful của Student

- `GET /students`: Lấy danh sách tất cả sinh viên.
- `GET /students/{id}`: Lấy thông tin chi tiết của một sinh viên dựa trên ID.
- `POST /students`: Thêm một sinh viên mới.
- `PUT /students/{id}`: Cập nhật thông tin của một sinh viên dựa trên ID.
- `DELETE /students/{id}`: Xóa một sinh viên dựa trên ID.



Lưu ý

- Ngoài ra, Spring Data REST cung cấp các tính năng bổ sung như **phân trang, sắp xếp, lọc** và tìm kiếm, tự động tạo HAL (Hypertext Application Language) links cho các tài nguyên liên quan, và nhiều tính năng khác.
- Tuy nhiên, nếu dự án của bạn có các yêu cầu tùy chỉnh phức tạp hơn, hoặc bạn muốn kiểm soát chặt chẽ hơn việc tạo API, bạn có thể tiếp tục sử dụng các lớp controller và service tùy chỉnh như đã triển khai trong các ví dụ trước.



Spring HATEOAS

2.1.1



OVERVIEW

LEARN

SUPPORT

SAMPLES

Spring HATEOAS provides some APIs to ease creating REST representations that follow the HATEOAS principle when working with Spring and especially Spring MVC. The core problem it tries to address is link creation and representation assembly.

Features

- Model classes for link, resource representation models
- Link builder API to create links pointing to Spring MVC controller methods
- Support for hypermedia formats like HAL



Có thể tùy chỉnh endpoint

- `Spring.data.rest.base-path=/xyz`



@RepositoryRestResource

- Tùy chỉnh Path

```
@RepositoryRestResource(path = "students")  
public interface StudentRepository extends JpaRepository<Student, Long> {  
    // Các phương thức tùy chỉnh khác có thể được khai báo ở đây  
}
```



Phân trang

- `spring.data.web.pageable.default-page-size=10`
- `spring.data.web.pageable.max-page-size=10`

```
GET /students?page=1
```



Sắp xếp

- Sắp xếp theo tên của thuộc tính

```
GET /students?sort=firstName,asc&sort=lastName,desc
```

