

RESTFul API

1. Khái niệm RESTFul

RESTful API là một tiêu chuẩn dùng trong việc thiết kế API cho các ứng dụng web (thiết kế Web services) để tiện cho việc quản lý các resource. Nó chú trọng vào tài nguyên hệ thống (tệp văn bản, ảnh, âm thanh, video, hoặc dữ liệu động...), bao gồm các trạng thái tài nguyên được định dạng và được truyền tải qua HTTP

2. Cách RESTFul hoạt động

REST hoạt động chủ yếu dựa vào giao thức HTTP. Các hoạt động cơ bản nêu trên sẽ sử dụng những phương thức HTTP riêng.

- GET (SELECT): Trả về một Resource hoặc một danh sách Resource.
- POST (CREATE): Tạo mới một Resource.
- PUT (UPDATE): Cập nhật thông tin cho Resource.
- DELETE (DELETE): Xóa một Resource.

Những phương thức hay hoạt động này thường được gọi là **CRUD** tương ứng với Create, Read, Update, Delete – Tạo, Đọc, Sửa, Xóa.

```
@GetMapping("/employees")
List<Employee> all() {
    return repository.findAll();
}
```

Get all employees

```
@PostMapping("/employees")
Employee newEmployee(@RequestBody Employee newEmployee) {
    return repository.save(newEmployee);
}
```

Add employee

```
@GetMapping("/employees/{id}")
Employee one(@PathVariable Long id) {

    return repository.findById(id)
        .orElseThrow(() -> new EmployeeNotFoundException(id));
}
```

Get employee by id

```

@PutMapping("/employees/{id}")
Employee replaceEmployee(@RequestBody Employee newEmployee, @PathVariable Long id) {

    return repository.findById(id)
        .map(employee -> {
            employee.setName(newEmployee.getName());
            employee.setRole(newEmployee.getRole());
            return repository.save(employee);
        })
        .orElseGet(() -> {
            newEmployee.setId(id);
            return repository.save(newEmployee);
        });
}

```

Update employee by id

```

@DeleteMapping("/employees/{id}")
void deleteEmployee(@PathVariable Long id) {
    repository.deleteById(id);
}

```

Delete employee by id

3. Thêm link vào hoạt động liên quan.

Một thành phần quan trọng đối với bất kỳ dịch vụ RESTful nào là thêm liên kết vào các hoạt động liên quan. Để làm cho bộ điều khiển của bạn trở nên RESTful hơn.

```

@GetMapping("/employees/{id}")
EntityModel<Employee> one(@PathVariable Long id) {

    Employee employee = repository.findById(id) //
        .orElseThrow(() -> new EmployeeNotFoundException(id));

    return EntityModel.of(employee, //
        linkTo(methodOn(EmployeeController.class).one(id)).withSelfRel(),
        linkTo(methodOn(EmployeeController.class).all()).withRel("employees"));
}

```

Thêm link

```
{
  "id": 1,
  "name": "Bilbo Baggins",
  "role": "burglar",
  "_links": {
    "self": {
      "href": "http://localhost:8080/employees/1"
    },
    "employees": {
      "href": "http://localhost:8080/employees"
    }
  }
}
```

Kết quả trả về client