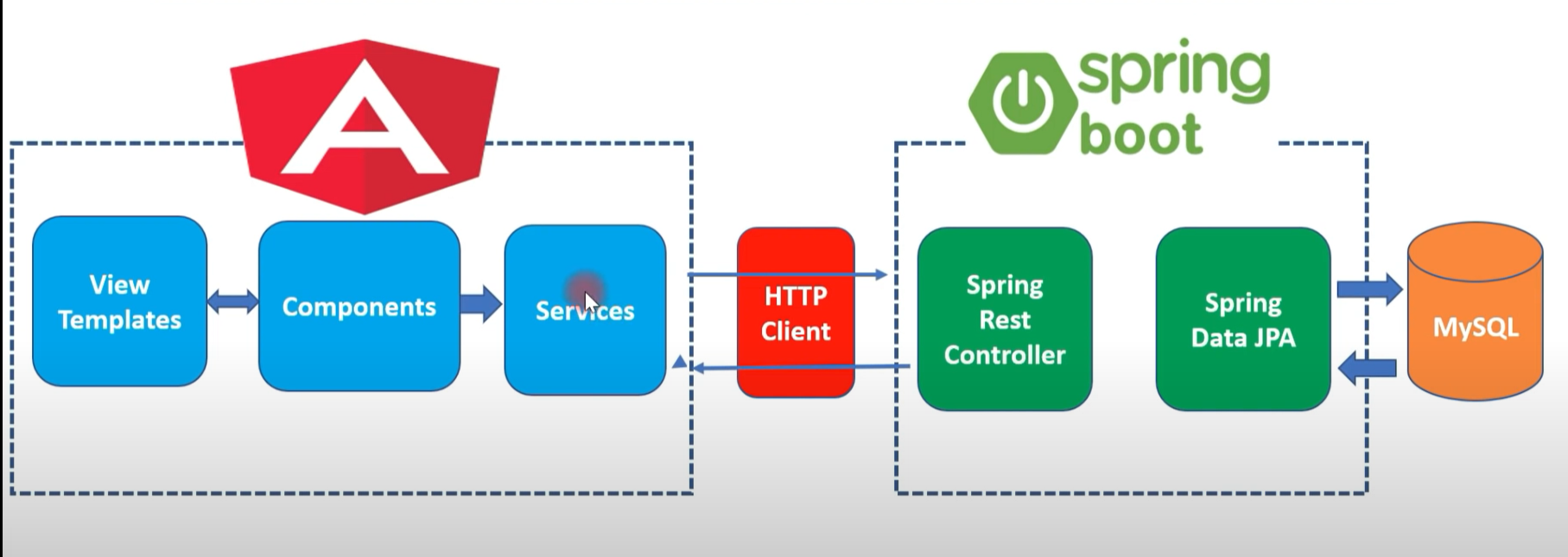
**1. Http Client là gì?**

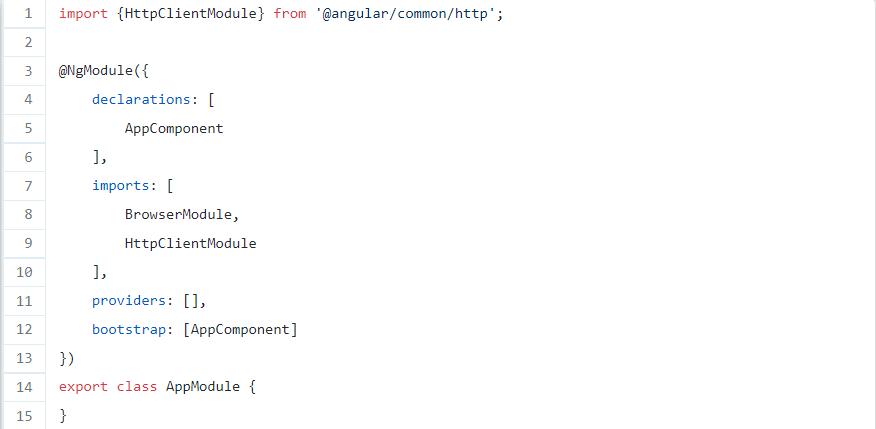
Hầu hết các ứng dụng front-end cần giao tiếp với máy chủ qua giao thức HTTP, để tải xuống hoặc tải lên dữ liệu và truy cập các dịch vụ back-end khác (CRUD). Angular cung cấp một API HTTP máy khách cho các ứng dụng Angular, lớp dịch vụ HttpClient trong @ angular / common / http.



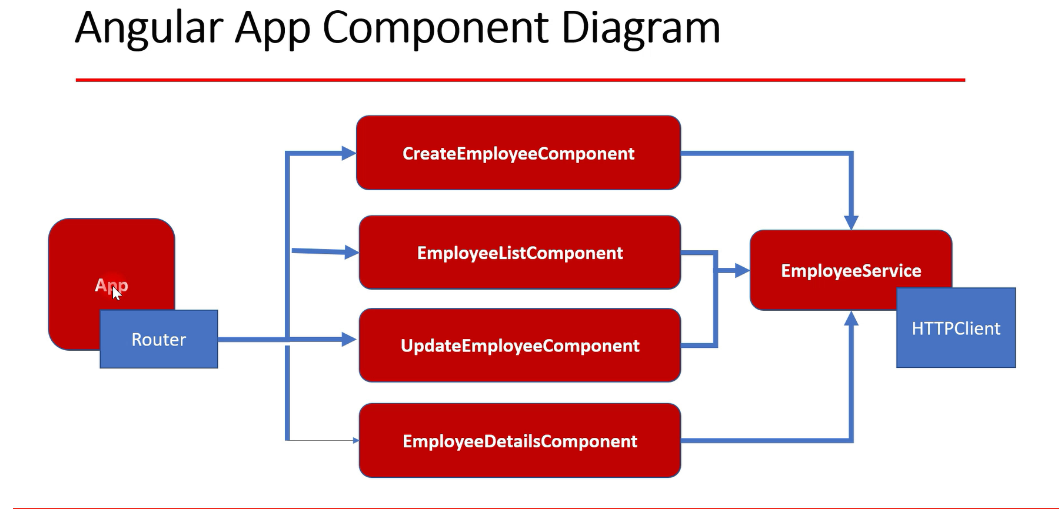
Mô hình kết nối angular app với spring boot thông qua httpClient

**2. Cài đặt liên kết với máy chủ**

* app modul: Trước khi có thể sử dụng HttpClient, bạn cần nhập Angular HttpClientModule. Hầu hết các ứng dụng đều làm như vậy trong AppModule gốc.



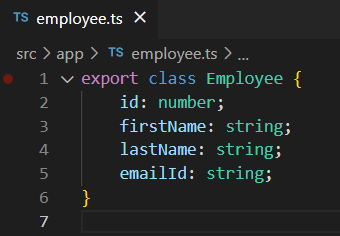
* service: Chia sẻ dữ liệu chung của các component
* Định nghĩa 1 class làm service và sử dụng @Injectable để đánh dấu class này có thể được tiêm vào các component





**4. Tạo model**

* Mục đích của tạo model là để làm việc với các object được gửi trong httpClient



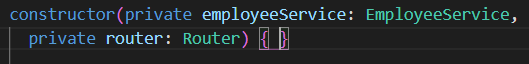
* Để thực hiện thêm mới 1 model, ta sử dụng câu lệnh ng new g class employee

**5.CRUD với httpClient**

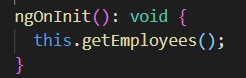
*3 bước:*

* Triển khai logic trong service
* Tạo template cho component
* Tạo component

*5.1 Lấy dữ liệu từ server sử dụng httpClient.get*

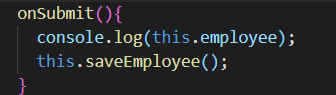


Inject vào component 1 đối tượng employeeService



Hàm ngOnInit() thực hiện khởi tạo 1 hàm mới mỗi khi reload trang

*5.2 Thêm với dữ liệu lên server sử dụng httpClient.post*



Gọi các hàm liên quan sử dụng onSubmit ( sẽ thực hiện khi người dùng submit form)

*5.3 Sửa dữ liệu sử dụng httpClient.put*



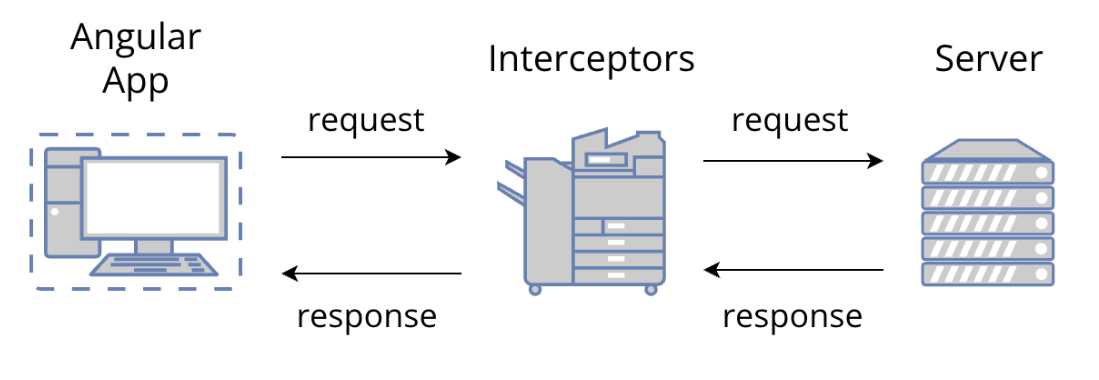
Lấy id của object tương ứng sử dụng snapshot.params

*5.4 Xóa dữ liệu sử dụng httpClient.delete*

**II. Angular interceptor**

**1. Angular interceptor là gì?**

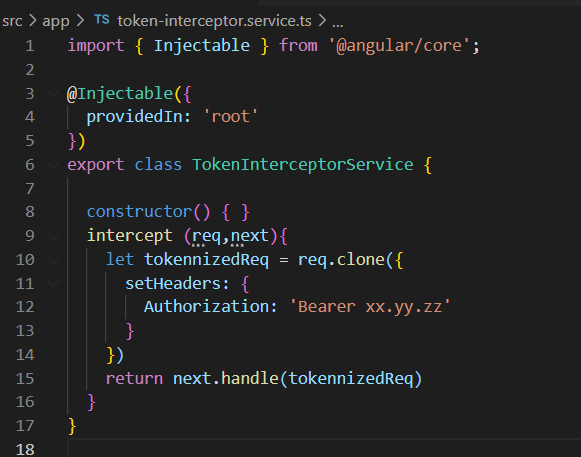
Angular interceptors là một loại dịch vụ HTTP client đặc biệt có mục đích duy nhất là chặn mọi yêu cầu HTTP được thực hiện. Điều này đúng cho cả yêu cầu HTTP đến và đi



Sơ đồ trên cho thấy rằng các bộ chặn HTTP sẽ luôn ở giữa bất kỳ yêu cầu HTTP đơn lẻ nào. Các dịch vụ này sẽ chặn tất cả các yêu cầu do ứng dụng thực hiện, cho phép chúng ta thực hiện nhiều thao tác trên chúng trước khi chúng được gửi đến máy chủ. Các chức năng bao gồm thêm tiêu đề HTTP tùy chỉnh vào yêu cầu gửi đi cuối cùng (ví dụ: thêm tiêu đề ủy quyền và chuyển mã thông báo ủy quyền trên tất cả các điểm cuối yêu cầu một bộ quyền, v.v.), lưu vào bộ nhớ đệm, ghi nhật ký để thu thập số liệu, xử lý lỗi,…

### 2. Custom Header Interceptor

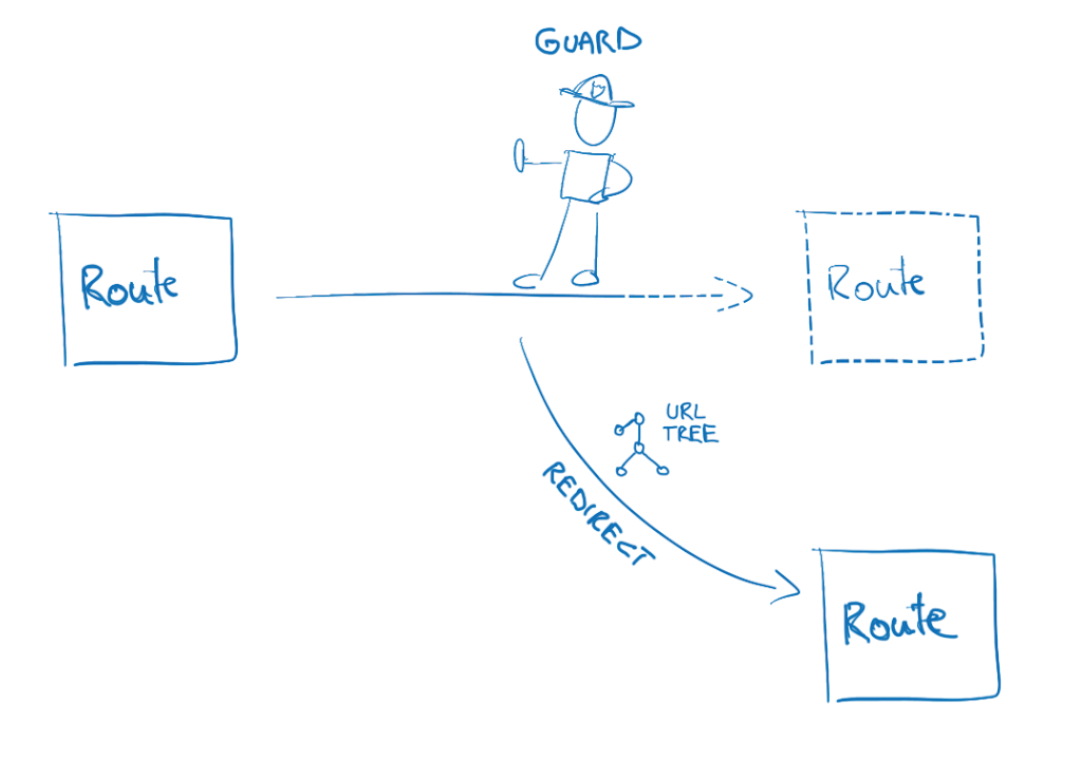
Một trong những trường hợp sử dụng phổ biến nhất cho máy đánh chặn là xử lý các yêu cầu của Auth. Điều này có thể đạt được dễ dàng bằng cách thêm các tiêu đề cần thiết vào yêu cầu đi bên trong phương thức chặn.



**III. Route Guards**

**1. Route Guards là gì?**

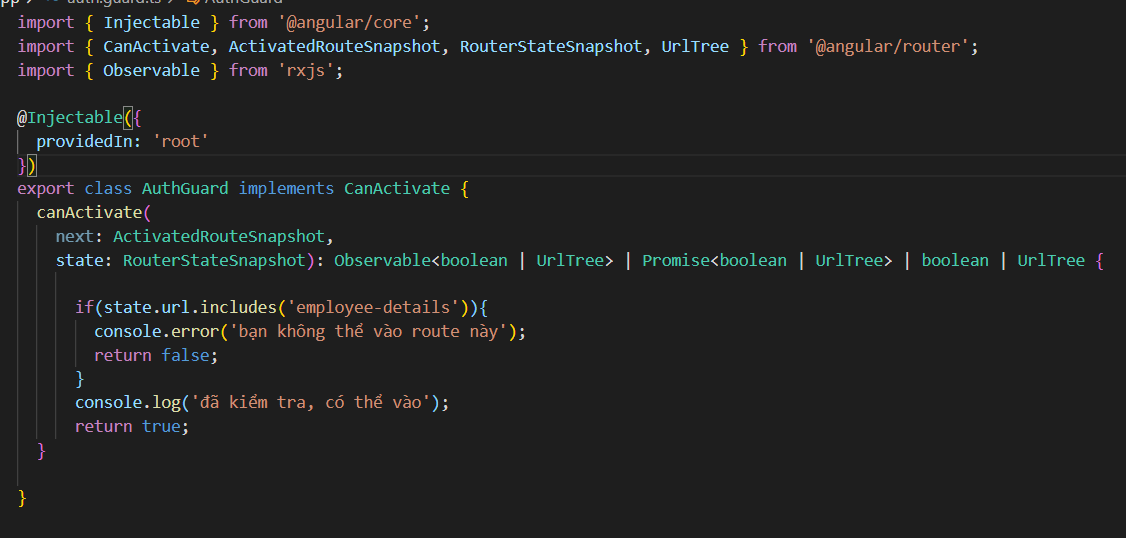
Trong một ứng dụng web, chúng ta thường đối mặt với kịch bản một số page cho phép tất cả mọi người truy cập, ngược lại một số khác chỉ dành cho các user đã login vào hệ thống như admin system chẳng hạn. Để giải quyết vấn đề này chúng ta hoàn toàn có thể thực hiện dễ dàng trên server side. Tuy nhiên, nếu chúng ta xây dựng Single page application(SPA) như Angular thì phải làm thế nào. Với SPA thì client side thường độc lập hoàn toàn với server, do vậy chúng ta cũng cần cơ chế để handle việc bảo vệ routes.

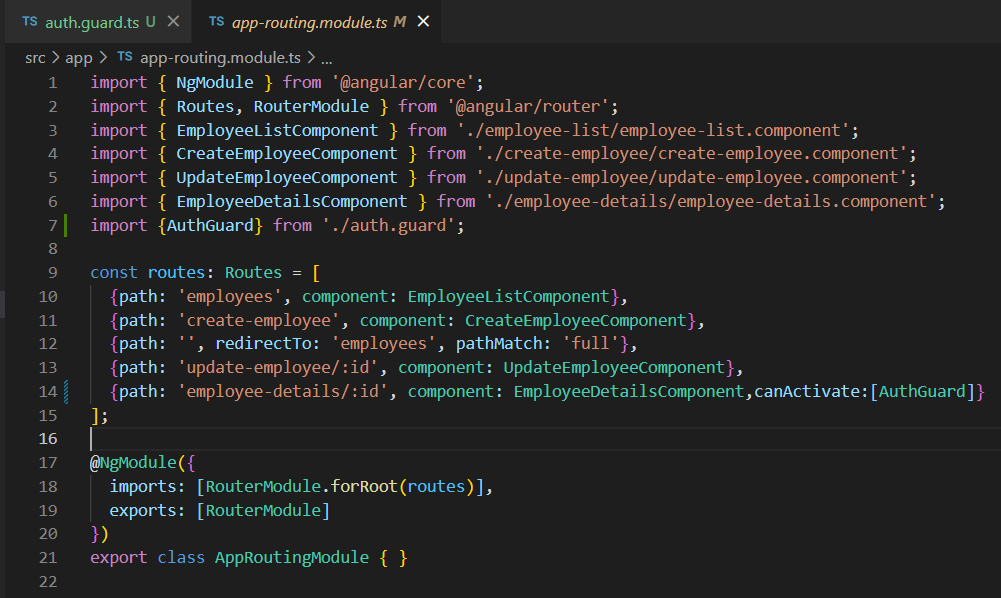


**2. Guard Types**

* **CanActivate:** Quyết định việc một route được kich hoạt.
* **CanActivateChild**: Quyết định việc children routes được kich hoạt
* **CanDeactivate**: Quyết định việc một route hủy kích hoạt
* **CanLoad**: Quyết định một module được lazy loading

**3. Ví dụ về Canactivate Guard**





Trên thực tế Guard thường được sử dụng để phân quyền người dùng truy cập vào trang, mỗi người dùng với 1 vai trò riêng sẽ được ủy quyền truy cập vào các trang phù hợp.

Ví dụ trên chỉ làm nổi bật được cách đánh chặn truy cập vào 1 Route của trang, chưa kết hợp được JWT để phân quyền.