# Http client & webservice

## Http client

Http Client là một Service Module được cung cấp bởi Angular giúp chúng ta thực hiện những yêu cầu Http, dễ dàng custom các request option và handle error một cách dễ dàng.

Bản chất nó được gọi là Service Module vì nó chỉ init các services (http client, http backend, etc) và không export bất cứ component hay directive nào.

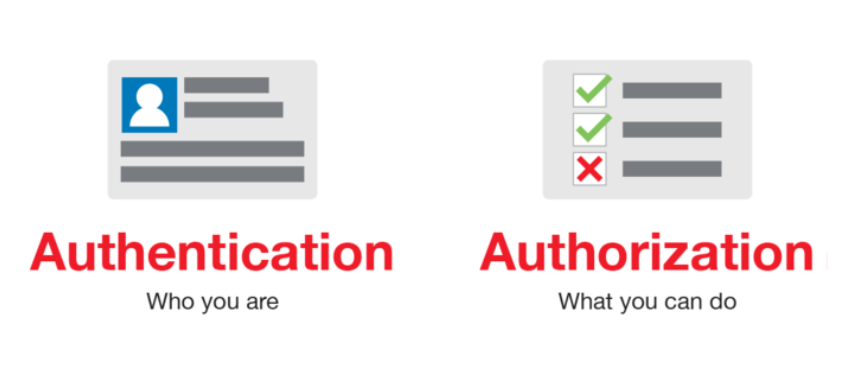
Nó cung cấp cho chúng ta đủ các methods để làm việc với API như:

* GET: được sử dụng để lấy thông tin từ server theo URL đã cung cấp.
* HEAD: giống với GET nhưng response trả về không có body, chỉ có header.
* POST: gửi thông tin tới server thông qua các biểu mẫu http (ví dụ như đăng ký..).
* PUT: ghi đè tất cả thông tin của đối tượng với những gì được gửi lên.
* PATCH: ghi đè các thông tin được thay đổi của đối tượng.
* DELETE: xóa tài nguyên trên server.
* CONNECT: thiết lập một kết nối tới server theo URI.
* OPTIONS: mô tả các tùy chọn giao tiếp cho resource.
* TRACE: thực hiện một bài test loop - back theo đường dẫn đến resource.

POST và GET là 2 phương thức phổ biến nhất. Thông thường mình có thể dùng POST cho cả việc tạo mới, update, delete đều được nhưng không nên khuyến khích dùng. Post chỉ dùng cho tạo, GET dùng cho xem thông tin, PUT dùng cho cập nhật và DELETE dùng cho việc xoá

# Authentication & Authorization

Cả 2 thuật ngữ này thường được sử dụng kết hợp với nhau để nói về bảo mật, đặc biệt là khi nói đến quyền truy cập vào hệ thống.



|  |  |
| --- | --- |
| Authentication (xác thực) | Authorization (ủy quyền) |
| Authentication xác nhận danh tính của bạn để cấp quyền truy cập vào hệ thống. | Authorization xác định xem bạn có được phép truy cập tài nguyên không. |
| Đây là quá trình xác nhận thông tin đăng nhập để có quyền truy cập của người dùng. | Đó là quá trình xác minh xem có cho phép truy cập hay không. |
| Nó quyết định liệu người dùng có phải là những gì anh ta tuyên bố hay không. | Nó xác định những gì người dùng có thể và không thể truy cập. |
| Authentication thường yêu cầu tên người dùng và mật khẩu. | Các yếu tố xác thực cần thiết để authorization có thể khác nhau, tùy thuộc vào mức độ bảo mật. |
| Authentication là bước đầu tiên của authorization vì vậy luôn luôn đến trước. | Authorization được thực hiện sau khi authentication thành công. |

Dựa trên cấp độ bảo mật, authentication factor có thể thay đổi theo một trong các cách sau:

* Single-Factor Authentication – Nó là phương thức xác thực đơn giản nhất thường dựa vào mật khẩu đơn giản để cấp cho người dùng quyền truy cập vào một hệ thống cụ thể là một website hoặc network. Ví dụ phổ biến nhất về xác thực một yếu tố sẽ là thông tin đăng nhập chỉ yêu cầu mật khẩu đối với tên người dùng hoặc địa chỉ email.
* Two-Factor Authentication – Nó có một quy trình xác minh gồm hai bước, không chỉ yêu cầu tên người dùng và mật khẩu, mà còn một thứ mà chỉ người dùng biết. Ví dụ như: sở thích, tên gọi lúc nhỏ, tên crush,…
* Multi-Factor Authentication – Đây là phương pháp xác thực tiên tiến nhất đòi hỏi hai hoặc nhiều mức bảo mật từ các loại xác thực độc lập để cấp quyền truy cập cho người dùng vào hệ thống. Hình thức xác thực này sử dụng các yếu tố độc lập với nhau để loại bỏ mọi phơi nhiễm dữ liệu. Thông thường các tổ chức tài chính, ngân hàng và các cơ quan thực thi pháp luật sử dụng xác thực nhiều yếu tố.

## ANGULAR AUTHENTICATION: SỬ DỤNG ROUTE GUARDS

Route Guards để giải quyết câu hỏi, liệu tôi có được phép redirect đến URL này hay không. Nó đưa ra quyết định này bằng cách tìm kiếm một giá trị trả về true hoặc false từ một lớp được implement từ guard interface.

Có 5 loại guards khác nhau và mỗi loại được gọi theo một trình tự cụ thể. Hành vi của bộ định tuyến được điều chỉnh khác nhau tùy thuộc vào guard nào được sử dụng. Các loại đó là:

* CanActivate
* CanActivateChild
* CanDeactivate
* CanLoad
* Resolve

Sử dụng JSON Web Token (JWT) là một cách để đưa ra quyết định về việc có cho truy cập vào một route hay không bằng cách kiểm tra thời gian hết hạn của token. Khi token hết hạn, đây là dấu hiệu cho thấy người dùng đang không được xác thực và không được phép truy cập.

# Các mã trạng thái (Status Codes)

HTTP định nghĩa nhiều mã trạng thái, mỗi cái có một ý nghĩa cụ thể. Một vài loại phổ biến là:

* **2xx:** Các mã trạng thái rơi vào series 2xx có nghĩa là request thành công và không có lỗi. Mã 200 là ví dụ thường thấy.
  + **200: OK - Truy vấn đến máy chủ thành công. Thông tin trả lời của máy chủ sẽ phụ thuộc vào phương thức yêu cầu (GET – HEAD – POST – TRACE)**
  + **201: Created** – Đã khởi tạo. Yêu cầu đã được chấp nhận và kết quả sẽ dẫn tới tài nguyên mới được tạo ra.
* **3xx:** Một mã trong series 3xx có nghĩa là chuyển hướng. Điều này có nghĩa là máy chủ chuyển hướng đến một nơi khác để nhận request.
* **4xx:** Một lỗi 4xx ví dụ 400, 403, 404, ... được sử dụng khi có một lỗi trong request. Điều này có thể do nhiều nguyên nhân, chẳng hạn truy cập trái phép tới một tài nguyên, hoặc tài nguyên không tồn tại, các tham số không hợp lệ, ...
* **400: Bad request** – yêu cầu này bị lỗi: (Do cổng timeout)
* Do cổng thời gian để phản hồi cho website bị nghẽn hoặc bị lỗi. Lỗi này tương tự với lỗi 504: Gateway Timeout. Bạn có thể kéo xuống dưới tìm lỗi 504 tham khảo nhé.
* Sau khi thực hiện hết tất cả các cách trên mà website bạn truy cập vào vẫn báo lỗi 400: Bad File Request thì bạn nên liên hệ với đơn vị cho thuê máy chủ lưu trữ website của bạn để nhờ họ hỗ trợ.
* **403: Forbidden** – Bị cấm truy nhập
* Máy chủ hiểu yêu cầu nhưng không muốn cho máy con nhận thông tin.
* Thỉnh thoảng bạn có thể thấy thông điệp HTTP Error 403 – Forbidden, điều này có nghĩa là thông tin hoặc luồng dữ liệu được gửi bởi trình duyệt của máy trạm nhưng máy chủ web từ chối cho truy cập vì một vài lý do.
* Bạn cần kiểm tra lại đường dẫn của website xem đã chính xác chưa, đôi khi thông tin đó được bảo mật bởi người sở hữu website. Bạn có thể gặp thông báo như 401- Not Authorised cũng như thông báo HTTP Error 403 – Forbidden. Cũng có thể lỗi do traqng web bị đóng cửa nên khi bạn truy cập sẽ bị trả về lỗi này.
* **404: Not found**– không tìm thấy.
* Lỗi này có nghĩa trình duyệt có thể kết nối đến website, nhưng không tìm thấy tài liệu bạn cần. Hoặc trang web tạm thời không sẵn sàng hay cũng có thể website đã bị thay đổi.
* Việc kiểm tra và sửa trang web này cần có sự can thiệp của người quản trị. Ngoài ra, có thể trang web đã được chuyển tới địa chỉ khác nên khi bạn truy cập sẽ gặp lỗi.
* **5xx:**Cuối cùng một response 5xx được sử dụng khi có lỗi ở phí server. Điều này có nghĩa là server biết lỗi và không thể xử lý request. Thông thường, response kèm theo một mô tả ngắn gọn về nguyên nhân có thể gây ra lỗi.
* **500: Internal Server Error**
* Server bất ngờ bị lỗi HTTP nên nó không thể hoàn thành yêu cầu.
* Lỗi này thường do máy chủ nhận thấy có lỗi xảy ra nhưng không thể xác định chính xác lỗi gì và bạn không thể truy cập trang web bị lỗi hoặc tải các tệp tin mà bạn muốn.
* Để giải quyết vấn đề này, đầu tiên bạn cần refresh (tạm dịch: “làm tươi”) trang web bằng phím F5 hoặc biểu tượng refresh của các trình duyệt. Ở một vài trường hợp thì lỗi này chỉ là lỗi tạm thời và việc refresh trang web có thể giải quyết được.
* Nếu không được, bạn có thể ghé thăm trang web này sau hoặc báo lại với người quản trị web nếu thấy cần thiết.
* **503: Service Unavailable**
* Server không thể trả lời vì quá bận hoặc đang được bảo trì.
* Có nhiều loại thông báo của lỗi HTTP này như “503 Service Temporarily Unavailable” hoặc “503 Service Unavailable”… lỗi này do máy chủ web tạm thời không hoạt động hoặc máy chủ đang bận hoặc đang được bảo trì.
* Vấn đề liên quan tới máy chủ nên giải pháp sẽ không có tác dụng từ phía máy trạm, thử refresh lại trang web hoặc báo lại với người quản trị web. Nếu gặp lỗi “Service Unavailable – DNS Failure” có thể modem/Router của bạn có vấn đề, hãy thử khởi động lại chúng.