**Authorization là gì?**

Ý nghĩa của **Authorization(ủy quyền)** có thể được xem là một câu hỏi, liệu chúng ta có đủ điều kiện để truy cập tài nguyên được bảo mật trên Server không? Nếu câu trả lời là có, thì về mặt kỹ thuật có thể nói rằng chúng ta được ủy quyền để truy cập vào tài nguyên. Nếu câu trả lời là Không thì tức là chúng ta không được ủy quyền để truy cập tài nguyên. Ví dụ: bạn đã thêm dấu vân tay của bạn và em gái của bạn vào điện thoại. Bạn và em gái của bạn có thể mở cùng một điện thoại di động, điều đó có nghĩa là chỉ bạn và em gái của bạn được phép mở điện thoại và xem dữ liệu. Tương tự như vậy trong khi có thể có nhiều **API** trong một công ty hoặc một dự án. Không nhất thiết là mọi người sẽ có quyền truy cập trên tất cả các API. Chỉ những người được ủy quyền mới có thể truy cập các API được bảo mật.

**Authorization và Authentication**

**Authorization(Ủy quyền)** và **Authentication (Xác thực)** là hai thuật ngữ liên quan chặt chẽ. Hai thuật ngữ này cũng có thể gây nhầm lẫn. Trong phần này, chúng tôi sẽ xóa bỏ nhầm lẫn về hai thuật ngữ này.

**Authentication**  là một quá trình trình bày thông tin đăng nhập của bạn cho hệ thống và hệ thống xác thực thông tin đăng nhập của bạn. Những thông tin này cho hệ thống biết bạn là ai. Cho phép hệ thống đảm bảo và xác nhận danh tính người dùng. Ở đây hệ thống có thể là bất cứ thứ gì, nó có thể là một máy tính, điện thoại, ngân hàng hoặc bất kỳ cơ sở văn phòng vật lý nào.

Trong khi **Authorization** là một quá trình cho phép hoặc từ chối ai đó truy cập vào một cái gì đó, một khi Authentication được thực hiện.

Vì vậy, trong thuật ngữ layman Authentication cho biết bạn là ai trong khi Authorization cho biết bạn có thể làm gì. Khi một người truy cập vào máy chủ bằng **key / password**, server sẽ kiểm tra xem người đó có sẵn trong thư mục hay không và có được liên kết với cùng key / password. Nếu có, bạn có thể đi tiếp (**Authentication**). Nếu bạn có quyền truy cập vào tài nguyên, thì bạn sẽ được cấp quyền truy cập vào tài nguyên (**Authorized**).



Chúng ta sẽ cùng xem ví dụ ngắn bên dưới để bạn hiểu được server từ chối 1 người không được ủy quyền như thế nào nhé.

**Authorization using Postman**

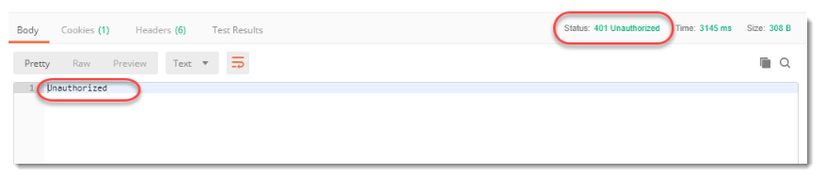
**Checking Authorization**

Ở phần này mình sẽ sử dụng endpoint là <https://postman-echo.com/basic-auth>

1. Tạo Get request và nhập endpoint <https://postman-echo.com/basic-auth>



1. Nhấn send và xem kết quả response



**Chú ý**: Mã trạng thái là 401 tương ứng với truy cập trái phép và thông báo trả về là Unauthorized. Mã trạng thái và trả về từ server cho biết chúng ta không được phép truy cập API mà mình đang cố gắng truy cập (Xem hướng dẫn Phản hồi để tìm hiểu thêm).

**Need for Authorization**

Trong phần cuối cùng, chúng ta đã thảo luận rằng chủ sở hữu tài nguyên không cho phép truy cập vào tài nguyên cho mọi người trong công ty. Điều này là do nó có thể dẫn đến vi phạm an ninh. Nếu tôi cho phép một thực tập viên truy cập vào cơ sở dữ liệu của mình thì anh ta có thể thay đổi dữ liệu và dữ liệu đó có thể bị mất mãi mãi, điều này có thể gây ra chi phí cho công ty. Có rất nhiều lý do có thể có cho cùng 1 vấn đề. Có thể một người thay đổi dữ liệu để lấy tiền hoặc một người có thể rò rỉ dữ liệu sang một công ty khác. Authorization đóng một vai trò rất quan trọng trong việc quyết định truy cập và thắt chặt an ninh. Chúng ta hãy cùng xem các loại Authentication khác nhau có sẵn nhé.

**Basic Access Authentication / HTTP Basic Authentication**

**Basic Access Authentication(Xác thực truy cập cơ bản)** là loại đơn giản và cơ bản nhất hiện có. Nó chỉ yêu cầu tên người dùng và mật khẩu để kiểm tra ủy quyền của 1 người nào đó (Đó là lý do tại sao chúng ta nói xác thực truy cập cơ bản).

Tên người dùng và mật khẩu được gửi dưới dạng giá trị tiêu đề trong **Authorization header**. Trong khi sử dụng xác thực cơ bản, chúng ta thêm từ Basic trước khi nhập tên người dùng và mật khẩu.

Các giá trị tên người dùng và mật khẩu này phải được mã hóa bằng **Base64** nếu không Server sẽ không thể nhận ra nó. Chúng ta sẽ làm theo các bước sau để kiểm tra xem mình có thể truy cập vào cùng API đã sử dụng ở trên hay không nhé.

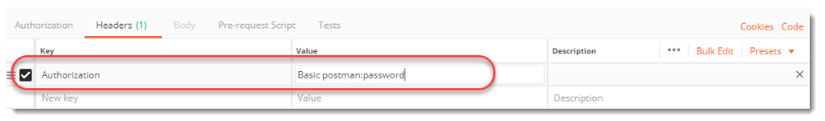
**Kiểm tra xác thực sử dụng thông tin đăng nhập**

1.Nhập endpoint <https://postman-echo.com/basic-auth> với **GET** request

1. Đi đến tab **Headers**



3.Nhập cặp giá trị key như bên dưới trong **Header** **Authorization   :    Basic postman:password**



Chú ý: Chúng ta đang sử dụng **username** là **postman** và **password** là **password**

4.Nhấn **Send** và xem kết quả trả về và status code.



Kết quả trả về vẫn là **400, Bad Request**. Bạn có thể phán đoán là tại sao không? Nếu bạn nhớ lại điều mình đã học ở phần cuối cùng thì basic access authentication yêu cầu username và password cần được mã hóa bằng Base64 nhưng ở đây mình lại truyền username và password dạng plain text. Do vậy server đã trả về 400, Bad Request. Mình cần hiểu Base64 coding là gì nhé.

**Base64 encoding là gì**

Mã hóa được sử dụng trong xác thực vì không muốn dữ liệu của chúng ta được truyền trực tiếp qua mạng. Có rất nhiều lý do cho điều đó. Máy quét mạng có thể đọc Yêu cầu của bạn và truy xuất Tên người dùng và Mật khẩu được gửi mà không cần mã hóa. Ngoài ra, các bit và byte được truyền trực tiếp có thể được coi là các bit lệnh sẵn có của modem hoặc các thiết bị khác trong chuỗi mạng. Ví dụ: nếu có một lệnh sẵn có là **0101101010** có nghĩa là đặt lại cho modem thì trong khi truyền, chúng ta có thể muốn có một chuỗi dữ liệu là **001101010010110101011020**. Ở đây, modem có thể hiểu nó là lệnh đặt lại và sẽ tự đặt lại. Để tránh những vấn đề như vậy có lợi cho việc mã hóa dữ liệu.Chúng ta đặc biệt sử dụng base64 vì nó truyền dữ liệu thành dạng văn bản và gửi nó ở dạng dễ hơn như dữ liệu dạng HTML. Hơn nữa có thể tin tưởng vào cùng 64 ký tự trong bất kỳ ngôn ngữ mã hóa nào mà mình sử dụng. Mặc dù chúng ta cũng có thể sử dụng các phương thức mã hóa cơ sở cao hơn nhưng chúng rất khó chuyển đổi và truyền tải gây lãng phí thời gian không cần thiết.Quay trở lại vấn đề ban đầu về việc gửi một chuỗi được mã hóa Base64 trong Authorization header. Trước mắt ta có 2 cách để tạo chuỗi mã hóa Base64:

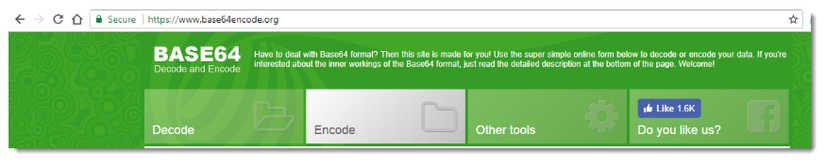
***- Thông qua trang web của bên thứ ba***

***- Thông qua Postman***

Chúng ta sẽ xem từng cách một. Hiện tại, hãy làm theo các bước để truy cập api bằng cách giải mã từ trang web của bên thứ ba.

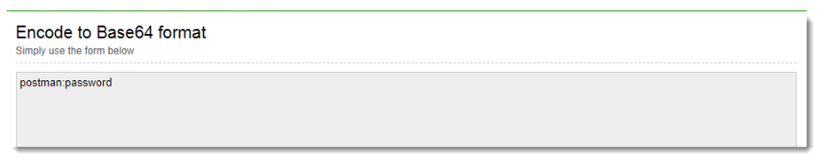
**Xác thực bằng mã hóa qua wesite thứ 3**

1.Đi đến <https://www.base64encode.org/>



**Lưu ý:** Có hàng ngàn trang web có sẵn cho cùng một mục đích. Bạn có thể sử dụng bất cứ trang nào chỉ cần đảm bảo rằng bạn mã hóa cùng giá trị với chúng tôi. Ngoài ra, chúng tôi đang sử dụng Microsoft Edge làm trình duyệt, mặc dù nó không tạo ra sự khác biệt nào.

2.Dán vào box giá trị như sau **postman:password**



3.Nhấn **Encode**.



4.Copy text đã được mã hóa

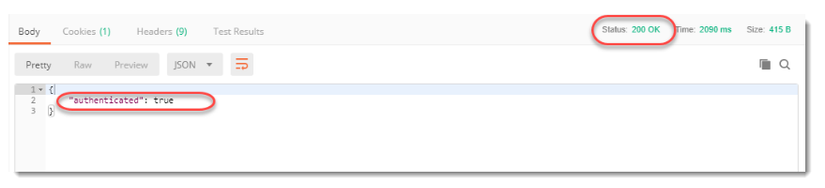


**Chú ý**: Không sử dụng khoảng trắng giữa bất ký text hay kí tự nào. Postman:password sẽ mã hóa thành giá tị khác trong khi postman: password sẽ mã hóa thành giá trị khác. Chỉ sử dụng postman:password thôi nhé.

5.Đi đến postman app và thay thế p**ostman:password** thành giá trị đã được mã hóa



6.Nhấn **send** và xem kết quả trả về và status code.

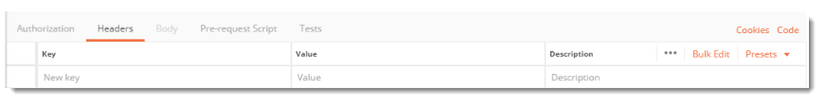


**200 OK, authenticated** có nghĩa là bạn đã cung cấp thông tin đăng nhập đúng và giờ bạn được ủy quyền để truy cập vào dữ liệu.

**Xác thực bằng mã hóa qua Postman**

Thay vì đi đến trang thứ 3, chúng ta thử mã hóa sử dụng Postman nhé.

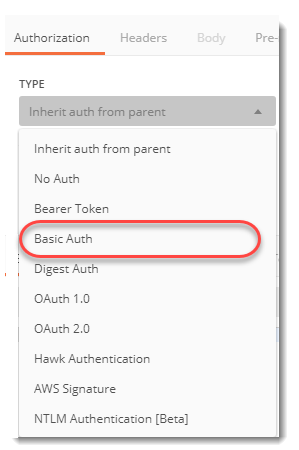
1.Bỏ cặp giá trị mà mình đã nhập trước đó đi



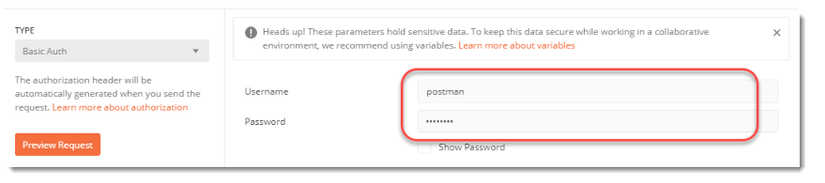
2.Đi đến tab **Authorization**



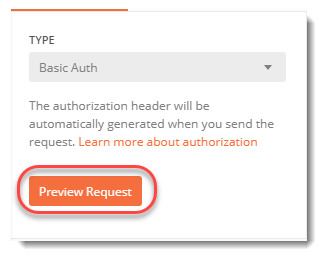
3.Chọn **BasicAuth** ở dropdown **Type**



4.Nhập **username** là **postman**, **password** là **password**



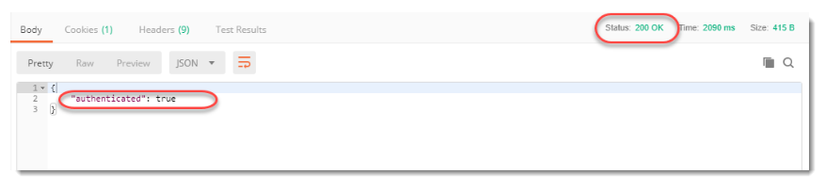
5.Nhấn **Preview Request**



6.Di chuyển sang tab **Header** và xem **Postman** đã chuyển đổi **username** và **password** cho bạn.



7.Nhấn **send** và kết quả là chúng ta đã được chứng thực.



Phần hướng dẫn của chúng tôi đã xong. Hy vọng bạn đã hiểu cơ bản về Authorization trong Postman. Authorization và Authentication sẽ là một trong những chủ đề quan trọng nhất mà bạn sẽ tìm hiểu, vì vậy hãy ghi nhớ tất cả các khái niệm có ở đây nhé.