# **VARIABLES TRÊN POSTMAN**

1. Đặt vấn đề
   1. Ví dụ thực tế

Giả sử bạn là HR quản lý chấm công cho 50 nhân viên IT của 1 công ty, trước đây có quy định thời gian làm của nhân viên IT là 9h -18h. Bạn sẽ dựa vào thời gian quy định để xét xem ai đi muộn sẽ bị phạt. Tuy nhiên hiện tại công ty thay đổi chính sách, yêu cầu nhân viên IT phải đi làm từ 8h – 17h. Vậy giờ làm sao để cập nhật mới thời gian chấm công quy định cho từng nhân viên ?

Nếu bạn vào từng profile của từng nhân viên để sửa thời gian chấm công quy định sẽ rất mất thời gian và có thể bị sót nhiều người.

Thay vào đó bạn có thể gom tất cả nhân viên đó vào 1 phòng ban là phòng IT, bạn sửa thời gian chấm công quy định của phòng IT đó từ 9h - 18h thành 8h - 17h. Vậy là giờ tất cả nhân viên thuộc phòng IT sẽ áp dụng thời gian chấm công mới của phòng IT mà bạn không cần vào profile từng người để sửa

* 1. Liên hệ với Postman

Trong postman có 1 collection gồm nhiều request, ví dụ có thể là

* <https://dev-umee.ssf.vn/cms/getAllUser>
* https://dev-umee.ssf.vn/cms/getAllAccount
* https://dev-umee.ssf.vn/profile/api/commune/v1/searchCommunesByName
* https://dev-umee.ssf.vn/nickname/api/shop/internal/register-nickname-shop
* https://dev-umee.ssf.vn/smartbank/api/savings/v1/onlineSavings/22420683
* <https://dev-umee.ssf.vn/smartbank/api/savings/v1/periods>
* <https://dev-umee.ssf.vn/nickname/api/edit-nickname>
* https://dev-umee.ssf.vn/cms/searchModule
* https://dev-umee.ssf.vn/cms/  
  smartbank/province?keyword=&page=0&size=10&deleted=&klbCode=&nameEn=

…

Bạn có thể thấy https/dev-umee.ssf.vn được lặp lại nhiều lần, vì tất cả các api trên đều nằm trên 1 server, nếu 1 ngày server được build lại và người quản trị đổi tên domain từ dev-umee.ssf.vn thành dev-flex-umee.ssf.vn, khi đó muốn chạy các api kia bạn phải vào từng api để đổi từng cái domain từ dev-umee.ssf.vn thành dev-flex-umee.ssf.vn, điều đó rất mất thời gian, cũng như dễ xảy ra sai sót và lỗi, vì nếu giả sử có 100 request bạn phải đổi 100 lần

Thay vào đó, bạn có thể gán <https://dev-umee.ssf.vn> với 1 biến (Variables) {{umeeDev}}, các api khác sẽ gọi biến {{umeeDev}} và map giá trị <https://dev-umee.ssf.vn> vào url, nếu có thay đổi bạn chỉ cần thay đổi giá trị của biến {{umeeDev}}, tất cả các api có biến {{umeeDev}} sẽ lấy giá trị bạn vừa thay đổi kia để map

Cách viết: <https://dev-umee.ssf.vn>/cms/getUser → {{umeeDev}}cms/getUser

1. Variables scope (biến)

Phạm vi truy cập:

A diagram of a variety of variables

Description automatically generated

Có 5 loại biến trong postman nhưng trong bài này chúng ta chỉ tìm hiểu 3 loại là global, collection và environment

* Global Variables:
* Collection Variables: biến có thể truy cập từ một request nằm trong collection tương ứng.
* Environment Variables: biến có thể truy cập với Environment tương ứng.(Environment trong Postman sẽ được đề cập tới ở phần dưới)
* Data Variables
* Local Variables

Dưới đây là sự giống và khác nhau giữa Collection Variables và Environment Variables

|  |  |  |
| --- | --- | --- |
|  | Collection Variables | Environment Variables |
| Giống nhau | * đều cho phép bạn lưu trữ các biến để sử dụng trong các api của bạn. * đều có thể được tham chiếu và sử dụng trong các yêu cầu bằng cách sử dụng cú pháp {{variable\_name}}. * đều cho phép bạn cập nhật và thay đổi giá trị của các biến. | |
| Khác nhau | * Được lưu trữ trong phạm vi của một collection. * Chỉ có thể sử dụng trong 1 collection | * Được lưu trữ trong phạm vi của environment. * Sử dụng ở nhiều collection khác nhau |

* 1. Global Varibles

Biến toàn cục: biến có thể truy cập ở bất kỳ đâu

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

B2: Kiểm tra biến đã tạo

A screenshot of a computer

Description automatically generated

* 1. Collection Variables

B1: Tạo biến

A screenshot of a computer

Description automatically generated

B2: Kiểm tra biến đã tạo

A screenshot of a computer

Description automatically generated

* 1. Environment Variables

B1: Tạo 1 Environment

A screenshot of a computer

Description automatically generated

B2: Tạo variable

* *Cách 1*: Chọn 1 api bất kỳ → ở góc bên trên bên phải chọn environment muốn tạo biến (ở đây là môi trường: umee)

A screenshot of a computer

Description automatically generated

Sau đó làm theo các bước dưới đây

A screenshot of a computer

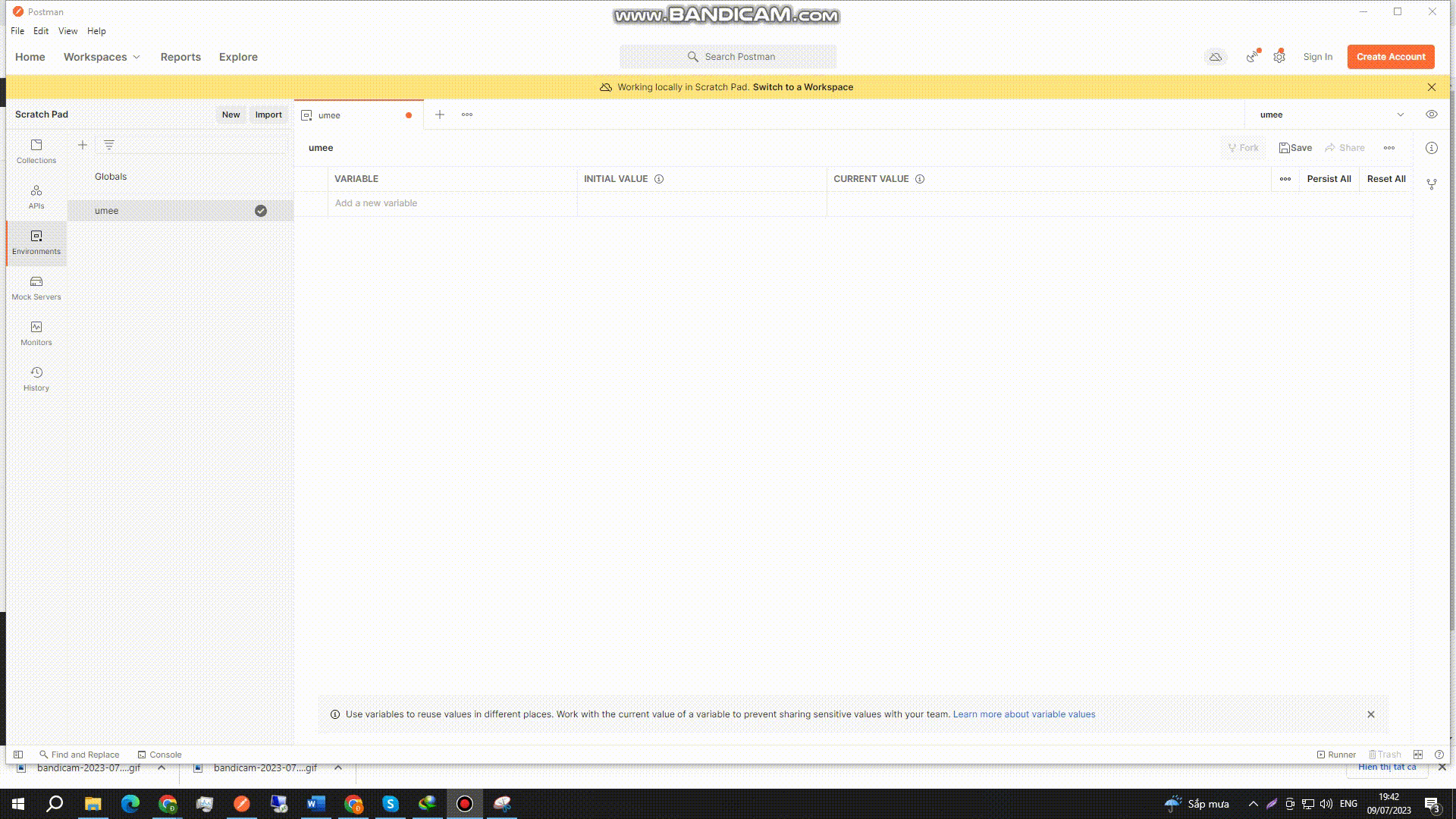
Description automatically generated

Kiểm tra biến

A screenshot of a computer

Description automatically generated

* *Cách 2*: Thêm trực tiếp vào environment



Initial Value: giá trị ban đầu – thứ khi bạn chia sẻ cho collection cho người khác, người ta sẽ thấy được giá trị đó

Current Value: giá trị gần nhất – giá trị đang được sử dụng trong postman của bạn và sẽ ko hiển thị khi bạn chia sẻ collection cho người khác

Environment trên Postman có tác dụng lưu trữ và quản lý các biến môi trường và giá trị tương ứng của chúng. Một môi trường trong Postman thường đại diện cho một bộ các giá trị biến mà bạn có thể sử dụng trong các request HTTP.

Trong Postman có hỗ trợ để thiết lập các biến (collection variables) và sử dụng các biến đó trong các request, khi giá trị được cập nhật thì chúng ta chỉ cần sửa giá trị của biến là được mà không cần vào từng request để điều chỉnh. Để tạo các biến các bạn chọn vào biểu tượng 3 chấm kế bên tên Collection và chọn Edit.

Đây là chức năng mình thấy khá là hữu ích . Thông thường thì chức năng chính của Environments có thể hiểu nôm na là một nơi lưu “biến” giống như “biến” trong code để mình có thể tái sử dụng ở nhiều nơi khác nhau .

Việc sử dụng environment trên Postman có các tác dụng sau:

1. Quản lý giá trị biến: Bằng cách lưu trữ các giá trị biến trong environment, bạn có thể dễ dàng quản lý và thay đổi các giá trị này mà không cần chỉnh sửa trực tiếp trong các request. Điều này giúp tiết kiệm thời gian và giảm thiểu lỗi khi cần thay đổi giá trị biến.
2. Tích hợp với nhiều môi trường: Postman cho phép bạn tạo nhiều môi trường khác nhau, chẳng hạn như Development, Production, Staging, để phù hợp với các môi trường khác nhau trong quá trình phát triển và triển khai ứng dụng. Bằng cách chuyển đổi giữa các môi trường, bạn có thể tự động thay đổi các giá trị biến phù hợp với từng môi trường mà không cần thay đổi code.
3. Chia sẻ và sử dụng lại: Bạn có thể chia sẻ environment với đồng nghiệp hoặc thành viên khác trong nhóm để đảm bảo mọi người sử dụng cùng một bộ giá trị biến. Điều này giúp đảm bảo sự nhất quán và tiết kiệm công sức khi phải cấu hình từ đầu. Ngoài ra, bạn cũng có thể sử dụng lại environment trong các bộ kiểm thử hoặc kịch bản tự động hóa.
4. Động và linh hoạt: Environment cho phép bạn tạo các biến động, chẳng hạn như sử dụng giá trị trả về từ một request trước đó và sử dụng nó làm giá trị cho request tiếp theo. Điều này giúp tạo ra các kịch bản phức tạp và linh hoạt hơn trong việc xử lý và truyền dữ liệu giữa các request.

Tóm lại, environment trên Postman giúp quản lý và cung cấp các giá trị biến môi trường cho các request HTTP, giúp bạn tiết kiệm thời gian, linh hoạt và nhất quán trong quá trình kiểm thử và phát triển ứng dụng.

Các thành phần cơ bản của Postman

1. Collection

Hiểu nôm na, nó chính là Folder, giúp đóng gói những request vào chung 1 chỗ. Ờ thế không dùng có được không? Câu trả lời là ĐƯỢC, tuy nhiên sẽ gặp phải 1 số vấn đề sau đây.

* Sẽ phải dùng History để tìm lại những request đã dùng, tương tự như bạn suốt ngày phải lục lọi phần History của Chrome, trong khi chỉ cần 1 động tác bookmark lại là xong.
* Không dùng được chức năng tạo API documents tự động mà Postman cung cấp
* Không thể dùng được chức năng Runner, giúp chạy liên tục các Request.

Các bước tạo 1 collection

B1: Chọn menu Collection trên menu → Chọn “+”

A screenshot of a computer

Description automatically generated

B2: Khi tạo mới sẽ có tên New collection, bạn đổi tên theo ý của bạn, ở đây tôi đẩy thành Umee nickname

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated