Enum in TS

Tập hợp nhóm các kiểu dữ liệu cùng một nhóm

AGENDA

1. Number enum
2. String enum
3. When to use enum
4. Bonus: how enum compiled to js

**What**: Tập hợp các giá trị cùng nhóm

**Why**: Dễ dàng truy xuất và quản lý

**When**: Sử dụng cho data một chiều, định nghĩa enum => rồi sử dụng

1. **Number enum**

enum *Status* {

    PENDING,    //0

    IN\_PROCESS, //1

    DONE,       //3

    CANCELLED   //4

}

enum *Status* {

    PENDING = 3, //3

    IN\_PROCESS,  //4

    DONE,        //5

    CANCELLED    //6

}

enum *Status* {

    PENDING = 3,

    IN\_PROCESS = 5,

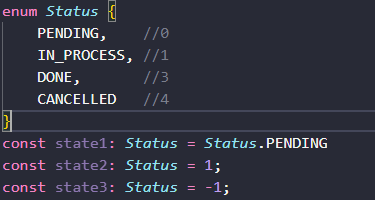
    CANCELLED = 10,

    DONE = 8,

}

Có thể gán bất kì giá trị number nào cho biến enum

VD:



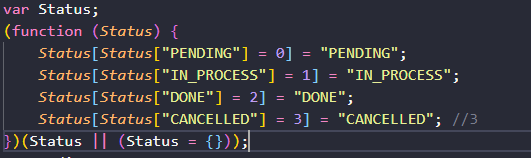
Đối với number enum có support reverse mapping



Trên console DevTool



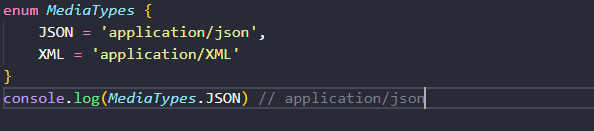
Numer enum in JS



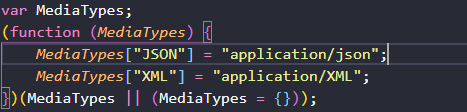
2 chiều vừa có giá trị number vừa có thể gán chuỗi string

1. **String enum**

Tập hợp các biến cùng một nhóm, thay vì giá trị của number enum là một con số thì string enum là một chuỗi tự định nghĩa



String enum in JS



Chỉ có giá trị chuỗi khi truy xuất (MediaTypes.JSON), (MediaTypes.XML)

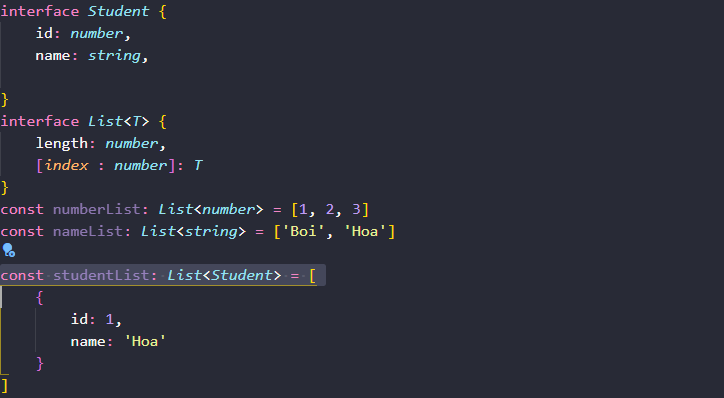
1. **When to use enum**

* Dữ liệu tĩnh – định nghĩa 1 chiều (định nghĩa ra và sử dụng) từ phía FE, không phải dữ liệu trả về từ API – Không parse dữ liệu từ server trả về (API) thành enum => chuyển về dạng union type

GENERICS

Generic: là kiểu dữ liệu có nhận về tham số

**Syntax:**



Key of operator: lấy ra tất cả các key của một kiểu dữ liệu nào đó