

MÔN: PHÁT TRIỂN ỨNG DỤNG PHÍA SERVER

Xây dựng ứng dụng ASP .NET Core WEB API có tên <Họ và tên><Mã số sinh viên> (vd: NguyenVanA1234)

1. Yêu cầu chung:

a. Tổ chức code

- Coding convention sử dụng PascalCase với tên các trường, class, interface. Sử dụng camelCase với các biến. Sử dụng _camelCase với các trường có access modifier là private.
- Cây thư mục code dạng như sau:
 - Properties: các config chạy ứng dụng
 - Constants: các hằng số
 - Controllers: các API Controller
 - DbContexts: các context entity framework
 - Dtos: chia các thư mục con theo nghiệp vụ
 - Entities: các class entity
 - Exceptions: các class xử lý ngoại lệ
 - Migrations: các migration tạo các cấu trúc bảng quan hệ
 - Services: xử lý các logic nghiệp vụ
 - Implements: triển khai các interface
 - Interfaces: khai báo các interface
 - Utils: các hàm hữu ích dùng chung
 - appsetting.json: các setting của ứng dụng dạng json
 - Program.cs

b. Yêu cầu từng thành phần

- Các class Dto phải xử lý **trim** đối với các trường kiểu dữ liệu **string**.
- Validate model bằng các built-in annotation attribute với các class Dto phục vụ cho việc create, update, delete (bao gồm bắt buộc nhập một số trường, số lượng ký tự với trường kiểu chuỗi, giá trị tối thiểu tối đa với trường kiểu số, ...).
- Kiểu dữ liệu trong các entity sử dụng phù hợp với bài toán.
- Các API Controller trả về kiểu dữ liệu phù hợp với interface IActionResult.
- Xử lý logic trả ra các ngoại lệ thông qua class UserFriendlyException:

```

public class UserFriendlyException : Exception
{
    14 references
    public UserFriendlyException(string message) : base(message)
    {
    }
}

```

- Tổ chức lưu trữ dữ liệu trong collection List nằm trong class ApplicationDbContext được add Singleton như các ví dụ demo.
- Các câu lệnh truy vấn cơ sở dữ liệu sử dụng các method Linq, keyword Linq.
- Xử lý logic nghiệp vụ theo design pattern Dependency injection (DI): dưới dạng các class service được inject vào controller khi sử dụng.
- Tổ chức các class theo hướng kế thừa (tùy chọn).
- Đặt tên tất cả các class theo dạng <Tên class><Mã số sinh viên><Đề thi số> (vd: StudentDto1234De1).

2. Bài toán cụ thể:

- Tạo một Database (bao gồm tạo migrations) để lưu trữ thông tin về sản phẩm gồm các trường sau: Id (kiểu số nguyên, tự tăng), Tên sản phẩm (kiểu chuỗi, bắt buộc nhập), Mô tả (kiểu chuỗi, không bắt buộc), Giá (kiểu số thực, bắt buộc nhập).
- Xây dựng các API để thực hiện các chức năng sau:
 - Lấy danh sách tất cả sản phẩm và sắp xếp danh sách sản phẩm theo giá từ cao đến thấp và theo Id giảm dần
 - Lấy thông tin của một sản phẩm dựa trên Id.
 - Thêm một sản phẩm mới.
 - Sửa thông tin của một sản phẩm dựa trên Id.
 - Xóa một sản phẩm dựa trên Id.
 - Tìm những sản phẩm có giá lớn nhất.
 - Sửa lại hàm thêm và cập nhật kiểm tra tên sản phẩm không được trùng với sản phẩm khác.
 - Tìm những sản phẩm có tên chứa chuỗi ký tự truyền vào.
 - Tìm những sản phẩm có giá nằm trong khoảng truyền vào.

Ghi chú: Sinh viên đọc kỹ đề, cán bộ coi thi không giải thích gì thêm.

----- Kết thúc -----