1)Install FluentValidation, FluentValidation.DependencyInjectionExtensions packages

builder.Services.AddValidatorsFromAssembly(typeof(Program).Assembly);

// Register all validators from the current assembly

2) Create a CreateOrderRequest, its dependencies CustomerInfo& CreateOrderItemRequest

public record CreateOrderRequest(

CustomerInfo Customer,

decimal TotalAmount,

DateTime OrderDate,

List<CreateOrderItemRequest> Items);

public record CustomerInfo(

string Name,

string Email);

public record CreateOrderItemRequest(

string ProductName,

int Quantity,

decimal price);

3)Create a folder called ‘Validator’ and create a class CustomerInfoValidator to validate CustomerInfo class and their properties, and also it implements AbstractValidator.

Inside constructor:

public CustomerInfoValidator()

{

**RuleFor**(x => x.Name)

.NotEmpty()

.WithMessage("Customer name is rquired");

**RuleFor** (x => x.Email)

.NotEmpty()

.WithMessage("Customer email is rquired")

.EmailAddress()

.WithMessage("Customer email is valid");

}

For CreateOrderItemRequest class

public CreateOrderItemRequestValidator()

{

**RuleFor** (x => x.ProductName)

.NotEmpty()

.WithMessage("Product name is required");

**RuleFor** (x => x.Quantity)

.GreaterThan(0)

.WithMessage("Quantity must be greater than zero");

**RuleFor** (x => x.price)

.GreaterThan(0)

.WithMessage("Price must be greater than zero");

}

For CreateOrderRequest

public class CreateOrderRequestValidator : AbstractValidator<CreateOrderRequest>

{

public CreateOrderRequestValidator()

{

**RuleFor**(x => x.TotalAmount)

.GreaterThan(0)

.WithMessage("Total amount must be greater than zero");

**RuleFor**(x => x.OrderDate)

.GreaterThan(DateTime.UtcNow.Date)

.WithMessage("Order date cannot be in the future");

**RuleFor**(x=>x.Customer)

.NotNull()

.WithMessage("Customer is required")

.SetValidator(new CustomerInfoValidator());

//SetValidator validate the object which pass here indirectly it validates Rulefor inside object

**RuleFor**(x => x.Items)

.NotEmpty()

.WithMessage("Order must contian at least one item");

**RuleForEach**(x=>x.Items).SetValidator(new CreateOrderItemRequestValidator());

//As we have List property in CreateOrderRequest so we are using RuleForEach to loop through all the items in the list to validate.

}

}

4. Finally include Validator in Controller endpoint

[HttpPost]

public IActionResult CreateOrder([FromBody] CreateOrderRequest request, IValidator<CreateOrderRequest> validator)

{

var validationResults = validator.Validate(request);

if (!validationResults.IsValid)

{

return BadRequest(validationResults.ToDictionary());

}

return Ok("Order created successfully");

}