

# Module 06:

## "Validation"



# Agenda

- ▶ Introduction
- ▶ **Validation using FluentValidation**
- ▶ MediatR Pipelines and Behaviors
- ▶ Summary



# Types of Validation

- ▶ Model Validation
  - In Presentation (or in entry to Application)
- ▶ Domain Validation
  - Business rules in Domain



# FluentValidation

- ▶ Freely available nuget package
  - <https://docs.fluentvalidation.net/en/latest/>
- ▶ **FluentValidation** is a .NET library for building strongly-typed validation rules using a Fluent API syntax
- ▶ Additionally;
  - `FluentValidation.AspNetCore`



# FluentValidation Basics

- ▶ Define type-safe validator for specified object

```
public class CustomerDtoValidator : AbstractValidator<CustomerDto>
{
    public CustomerDtoValidator()
    {
        RuleFor(customer => customer.Age).LessThan(100);
        RuleFor(customer => customer.FullName).NotNull();
    }
}
```

```
record class CustomerDto
{
    int Age { ... }
    string FullName { ... }
}
```

```
CustomerDto customer = new CustomerDto() { ... };
CustomerDtoValidator validator = new CustomerDtoValidator();

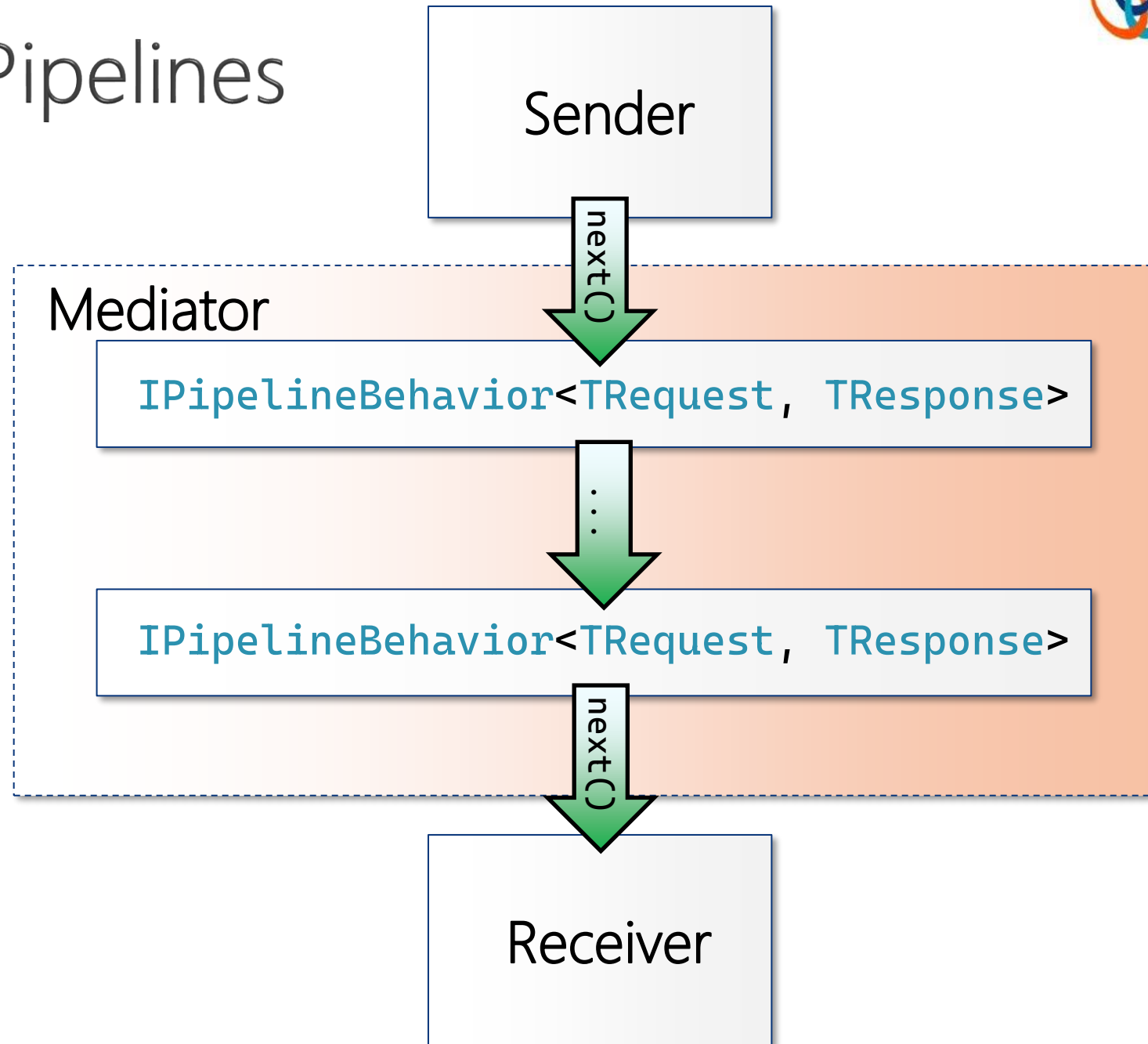
ValidationResult result = validator.Validate(customer);
```

# Agenda

- ▶ Introduction
- ▶ Validation using FluentValidation
- ▶ **MediatR Pipelines and Behaviors**
- ▶ Summary



# MediatR Pipelines



# Registering MediatR Pipeline Behaviors

```
services.AddMediatR(cfg =>
{
    cfg.AddBehavior<IPipelineBehavior<TRequest, TResponse>, ValidatorBehavior>();
});
```

```
services.AddMediatR(cfg =>
{
    cfg.AddOpenBehavior(typeof(GenericValidatorBehavior<, >));
});
```





# Registering All Validators

- ▶ Use add-on nuget package **FluentValidation.AspNetCore**

```
services  
    .AddValidatorsFromAssemblyContaining(typeof(DependencyInjection));
```

- ▶ Note: This registers all public validators in the specific assembly



# Summary

- ▶ Introduction
- ▶ Validation using FluentValidation
- ▶ MediatR Pipelines and Behaviors



