

第14期-模型验证机制

2021年2月8日 12:01

基于 Attribute 特性的数据注解验证

System.ComponentModel.DataAnnotations

System.ComponentModel.Annotations

Error Message Resources

```
public class ProjectCreateOrUpdateRequestModel
{
    public int Id { get; set; }

    [StringLength(8, ErrorMessage = "The {0} must be at least {2} character long.",
        MinimumLength = 6)]
    [System.ComponentModel.DisplayName("ProjectName")]
    public string Name { get; set; } = null!;

    [Display(Name = nameof(CreationTime))]
    public DateTimeOffset CreationTime { get; set; }
}
```

基于 Fluent 的链式验证

Install-Package FluentValidation

Install-Package FluentValidation.DependencyInjectionExtensions

```
services.AddValidatorsFromAssembly(Assembly.GetExecutingAssembly());
```

```
public class ProjectCreateOrUpdateRequestModelValidator :
    AbstractValidator<ProjectCreateOrUpdateRequestModel>
{
    public ProjectCreateOrUpdateRequestModelValidator()
    {
        RuleFor(m => m.Name).Length(5, 20).WithName("ProjectName");
    }
}
```

Install-Package FluentValidation.AspNetCore

```
mvcBuilder.AddFluentValidation(fv =>
    fv.RegisterValidatorsFromAssemblies(AppDomain.CurrentDomain.GetAssemblies()));
```

零度基于.NET的FluentValidation验证教程

手动调用验证器

```
public class TestsController : ControllerBase
{
    private readonly IValidator<ProjectCreateOrUpdateRequestModel> _validator;

    public TestsController(IValidator<ProjectCreateOrUpdateRequestModel> validator)
    {
        _validator = validator;
    }

    [HttpPost]
    public async Task<ActionResult<ProjectGetResponseModel>> Post([FromBody]
ProjectCreateOrUpdateRequestModel value)
    {
        ValidationResult validationResult = _validator.Validate(value);

        if (validationResult.IsValid)
        {
            return await Task.FromResult(new ProjectGetResponseModel());
        }

        return BadRequest(validationResult);
    }
}
```