

# FluentEmail

## Sending Email with C#



@djokic-stefan

swipe >>>

# NuGet Package

```
dotnet add package FluentEmail.Core
```

## Dependency Injection



```
public void ConfigureServices(IServiceCollection services)
{
    services
        .AddFluentEmail("testfrom@gmail.com")
        .AddSmtpSender("smtp.gmail.com", 587);
}
```

### .AddFluentEmail()

FluentEmail setup with a default SendFrom address

### .AddSmtpSender()

Smtp Provider configuration

### .AddRazorRenderer()

(Optional) - Email templating support



```
var email = await _email
    .From("test.emaiik@gmail.com")
    .To("stefan@stefandjokic.tech", "Stefan")
    .Subject("Test for Stefan")
    .Body("Works!")
    .SendAsync();
```

**\_email = IFluentEmail from DI**

**Customized body mail?**



# Templates

**FluentEmail has core support for multiple template renderer providers.**



```
var template = "Hello @Model.FirstName @Model.LastName, ...";  
var model = new EmailModel  
{  
    FirstName = "Stefan",  
    LastName = "Djokic"  
};  
  
_email.DefaultRenderer = new RazorRenderer();  
  
var email = _email  
    .From("info@test.com")  
    .To("stefan@stefandjokic.tech")  
    .Subject("Template testing")  
    .UsingTemplate(template, model);
```

**RazorRenderer - the most popular option for templating. Works like Razor in .NET.**

# Email Senders

**FluentEmail allows to implement of custom email providers by implementing the **ISender** interface.**

**Email providers already supported:**

- **SmtpClient**
- **SendGrid**
- **MimeKit**
- ...

**SmtpClient example:**



```
services
    .AddFluentEmail("testfrom@gmail.com")
    .AddSmtpSender("smtp.gmail.com", 587)
    .AddSmtpSender(new SmtpClient() {});
```

# Why FluentEmail?

- **Simple to use**
- **Simple to implement**
- **Good Dependency Injection Support**

---

Stefan Djokic



**WANT MORE POSTS LIKE THIS?**

**CLICK ON THE NOTIFICATION  
BELL ON MY PROFILE** 