

Bing

In an ASP.NET Core API project, you can set up dependency injection in the `Program.cs` file. Here's a step-by-step guide:

1. **Define your services and interfaces.** For example, if you have a service `MyService` that implements an interface `IMyService`, you would define them like this:

```
public interface IMyService
{
    void DoSomething();
}

public class MyService : IMyService
{
    public void DoSomething()
    {
        // Implementation here
    }
}
```

2. **Register your services in the `Program.cs` file.** Here's how you might register `MyService` as a singleton:

```
var builder = WebApplication.CreateBuilder(args);
builder.Services.AddSingleton<IMyService, MyService>();
var app = builder.Build();
```

3. **Inject the service into the constructor of the class where it's used.** Here's an example of how you might do this in an `IndexModel` class:

```
public class IndexModel : PageModel
{
    private readonly IMyService _service;

    public IndexModel(IMyService service)
    {
        _service = service;
    }

    public void OnGet()
    {
        _service.DoSomething();
    }
}
```

4. The framework takes on the responsibility of creating an instance of the dependency and disposing of it when it's no longer needed.

For more detailed information, you can refer to Microsoft's official documentation on [Dependency injection in ASP.NET Core](#) and [Dependency injection in .NET](#). They also provide sample code that you can view or download.