Software Resilience in .NET

- **Retry**
- **Timeout**
- **Fallback**
- **Hedging**
- **Circuit Breaker**
- **Rate Limiter**

Nabi Karampoor
@thisisnabi

SystemDesign

“

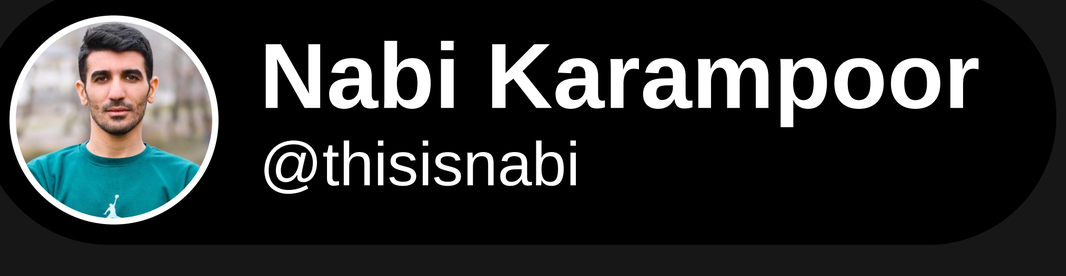
Basically, a system is resilient if it continues to carry out its mission in the face of adversity.

Nabi Karampoor
@thisisnabi

Polly is a powerful library for .NET that helps you handle transient faults and improve the resilience of your applications.

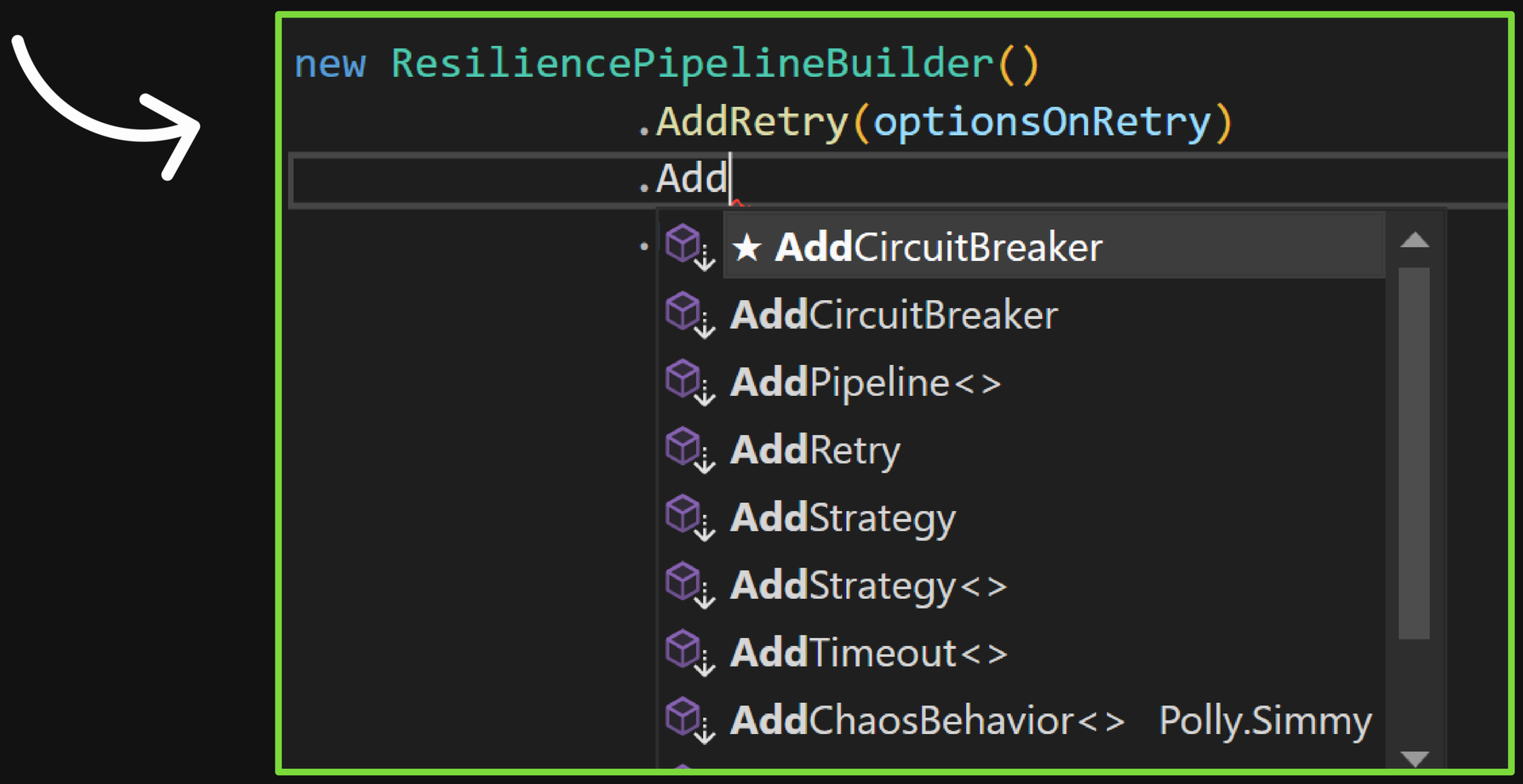
```
> dotnet add package Polly --version 8.3.0
```

There are different resilience strategies to cope with various scenarios, let's to see...



Nabi Karampoor
@thisisnabi

Polly lets you use and combine different resilience strategies to cope with various scenarios



Resilience Pipeline

allows executing arbitrary user-provided callbacks. It is a combination of one or more resilience strategies.

Nabi Karampoor
@thisisnabi

Retry

Try again if something fails

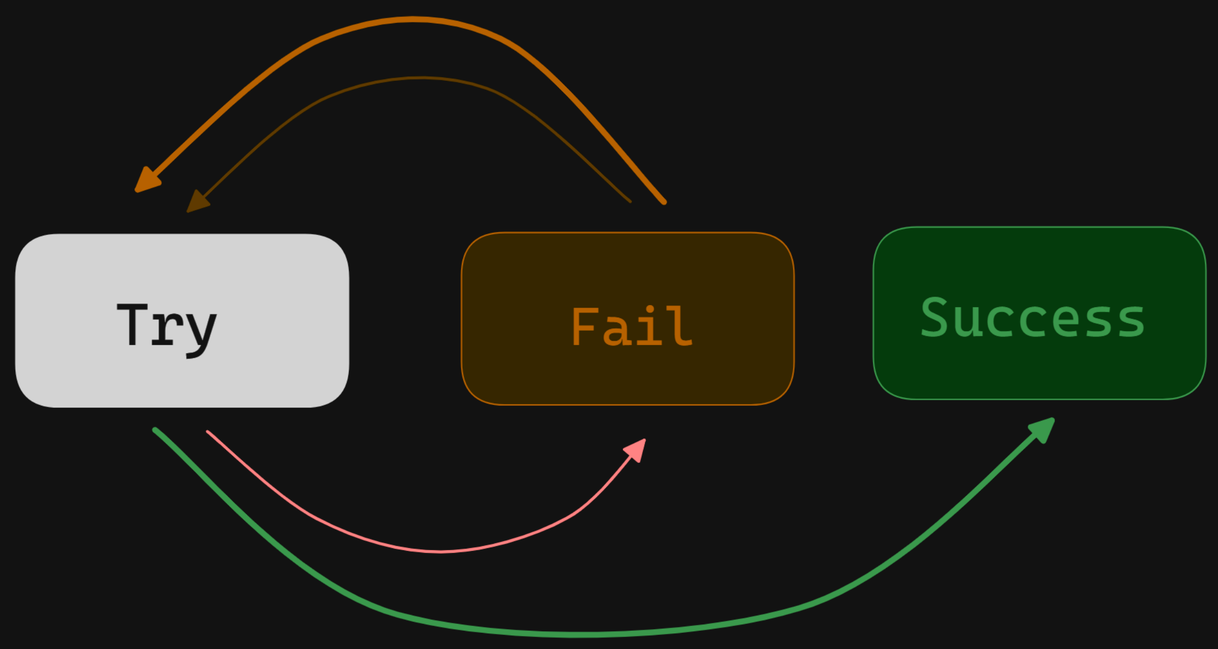
```
var optionsOnRetry = new RetryStrategyOptions
{
    MaxRetryAttempts = 2,
    // You can do the configurations here
};

var pipelineTry = new ResiliencePipelineBuilder()
                                .AddRetry(optionsOnRetry)
                                .Build();

pipelineTry.Execute(YourClassOrAnything .DoOnTry);

// pipelineTry.ExecuteAsync is available
```

The Method that you want to try to execute it.



Nabi Karampoor
@thisisnabi

Timeout

Give up if something takes too long

```
var optionsOnTimeout = new TimeoutStrategyOptions
{
    // other configuration
    OnTimeout = static args =>
    {
        // tell to me 😅
        return default;
    }
};

var pipelineTimeout = new ResiliencePipelineBuilder()
                                .AddTimeout(optionsOnTimeout)
                                .Build();
try
{
    await pipelineTimeout .ExecuteAsync(YourDelegate, CancellationToken.None);
}
catch (TimeoutRejectedException)
{
    Console.WriteLine("Timeout limit has been exceeded");
}
```

If the task is not done in this time, we will have an exception

Nabi Karampoor
@thisisnabi

Fallback

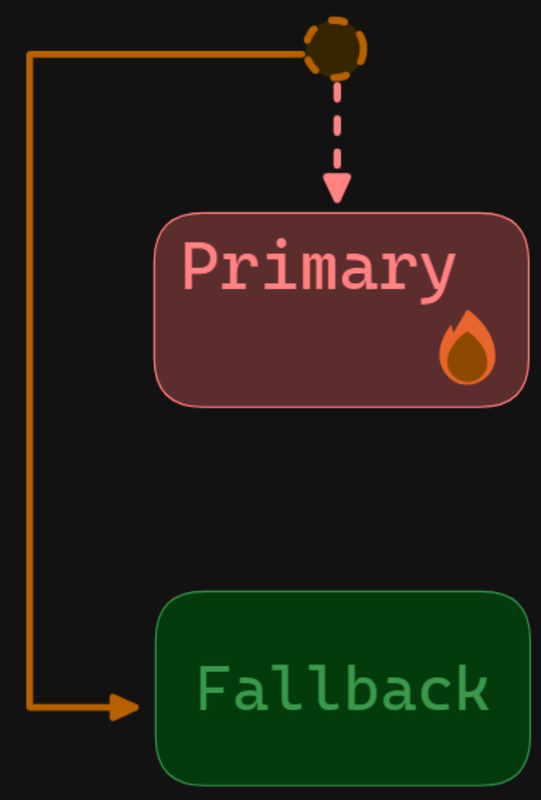
Do something else if something fails

```
var optionsOnFallback = new FallbackStrategyOptions<UserAvatar>
{
    // other configurations
    FallbackAction = static args =>
    {
        var avatar = UserAvatar.GetRandomAvatar();
        return Outcome.FromResultAsValueTask(UserAvatar.Blank);
    }
};

var pipelineFalback = new ResiliencePipelineBuilder()
                                    .AddFallback(optionsOnFallback)
                                    .Build();

pipelineFalback.Execute(YourPrimaryDelegate);
```

Always have an alternate plan



Nabi Karampoor
@thisisnabi

Other items in brief

Circuit Breaker

Stop trying if something is broken or busy

Rate Limiter

Limit how many requests you make or accept.

Hedging

Do more than one thing at the same time and take the fastest one.

Nabi Karampoor
@thisisnabi

Repost, so your fiends can learn too.

:) let’s follow