Outcome chaos strategy **CHEAT SHEET**

Basics

This reactive chaos **strategy injects outcome(s)** (result and/or Exception) to simulate unexpected response.

You can configure the behaviour of the strategy via the **ChaosOutcomeStrategyOptions<T>** object.

Specify single result - short form

```
new ResiliencePipelineBuilder<HttpResponseMessage>()
  .AddChaosOutcome(0.1,
    () => new HttpResponseMessage(
      HttpStatusCode.InternalServerError))
```

Specify single result – long form

```
new ResiliencePipelineBuilder<HttpResponseMessage>()
  .AddChaosOutcome(new
    ChaosOutcomeStrategyOptions<HttpResponseMessage>
    {
      InjectionRate = 0.1,
      OutcomeGenerator = static _ =>
        Outcome.FromResultAsValueTask(
          new HttpResponseMessage(
            HttpStatusCode.InternalServerError))
})
```

Specify multiple results with switch expression

```
new ResiliencePipelineBuilder<HttpResponseMessage>()
  .AddChaosOutcome(new ChaosOutcomeStrategyOptions<HttpResponseMessage>
  {
    OutcomeGenerator = static _ =>
    {
      var rnd = Random.Shared.NextDouble();
      HttpStatusCode statusCode = rnd switch
      {
        < 0.4 => HttpStatusCode.InternalServerError,
        >= 0.4 => HttpStatusCode.RequestTimeout,
        _ => HttpStatusCode.OK
      };
      return Outcome.FromResultAsValueTask(
         new HttpResponseMessage(statusCode));
    }
  })
```

Specify multiple results and an exception with OutcomeGenerator

```
new ResiliencePipelineBuilder<HttpResponseMessage>()
  .AddChaosOutcome(new ChaosOutcomeStrategyOptions<HttpResponseMessage>
  {
    OutcomeGenerator = new OutcomeGenerator<HttpResponseMessage>()
      .AddResult(() => new
HttpResponseMessage(HttpStatusCode.InternalServerError), weight: 40)
      .AddResult(() => new
HttpResponseMessage(HttpStatusCode.RequestTimeout), weight: 50)
      .AddException(() => new
HttpRequestException(HttpRequestError.ConnectionError), weight: 10),
  })
```

Specify asynchronous delegate for injection notification

```
new ResiliencePipelineBuilder<HttpResponseMessage>()
  .AddChaosOutcome(new ChaosOutcomeStrategyOptions<HttpResponseMessage>
  {
    OnOutcomeInjected = static async args => await NotifyAsync(args.Outcome)
  })
```