# Blazor Testing from A to Z

**Egil Hansen**
@**egil**@mastodon.social
@**egilhansen**
github.com/**egil**

## Writing Tests for Blazor

# Writing **Valuable** Tests for Blazor

## Properties of valuable (Blazor) tests?

reliable specific independent

maintainable efficient predictable quick

feedback facilitate-refactoring document-functionality

enhance-quality scalable transparent robust repeatable

isolated thorough automatable fast adaptive

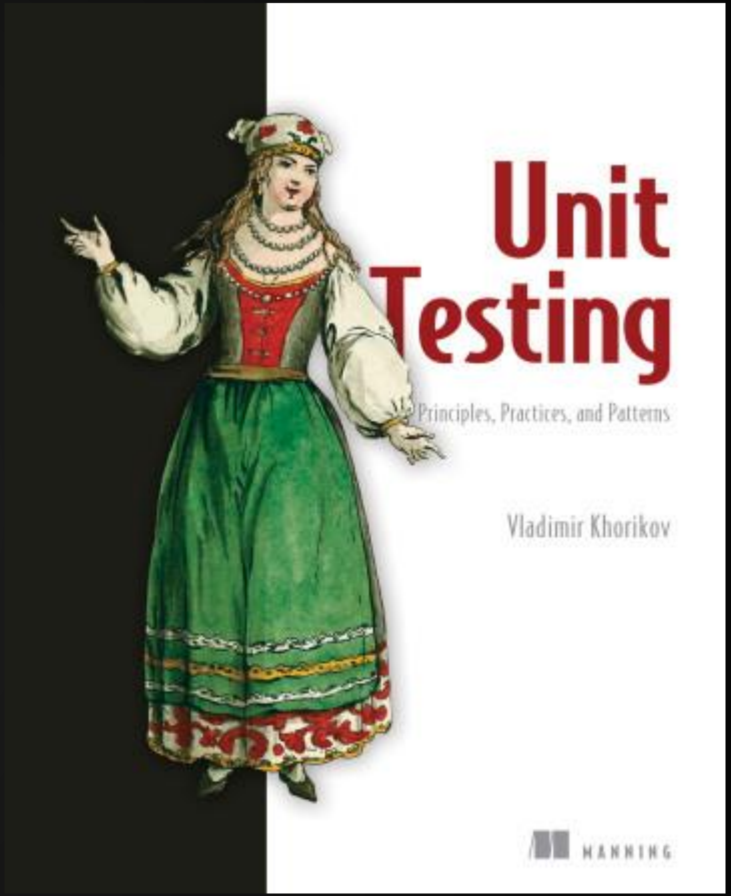
minimal focused structured

Protection against regressions

Resistance to refactoring

Fast feedback

Maintainable



"Unit Testing Principles Practices and Patterns“
**Vladimir Khorikov**
https://www.manning.com/books/unit-testing

## Protection against regressions

## Resistance to refactoring

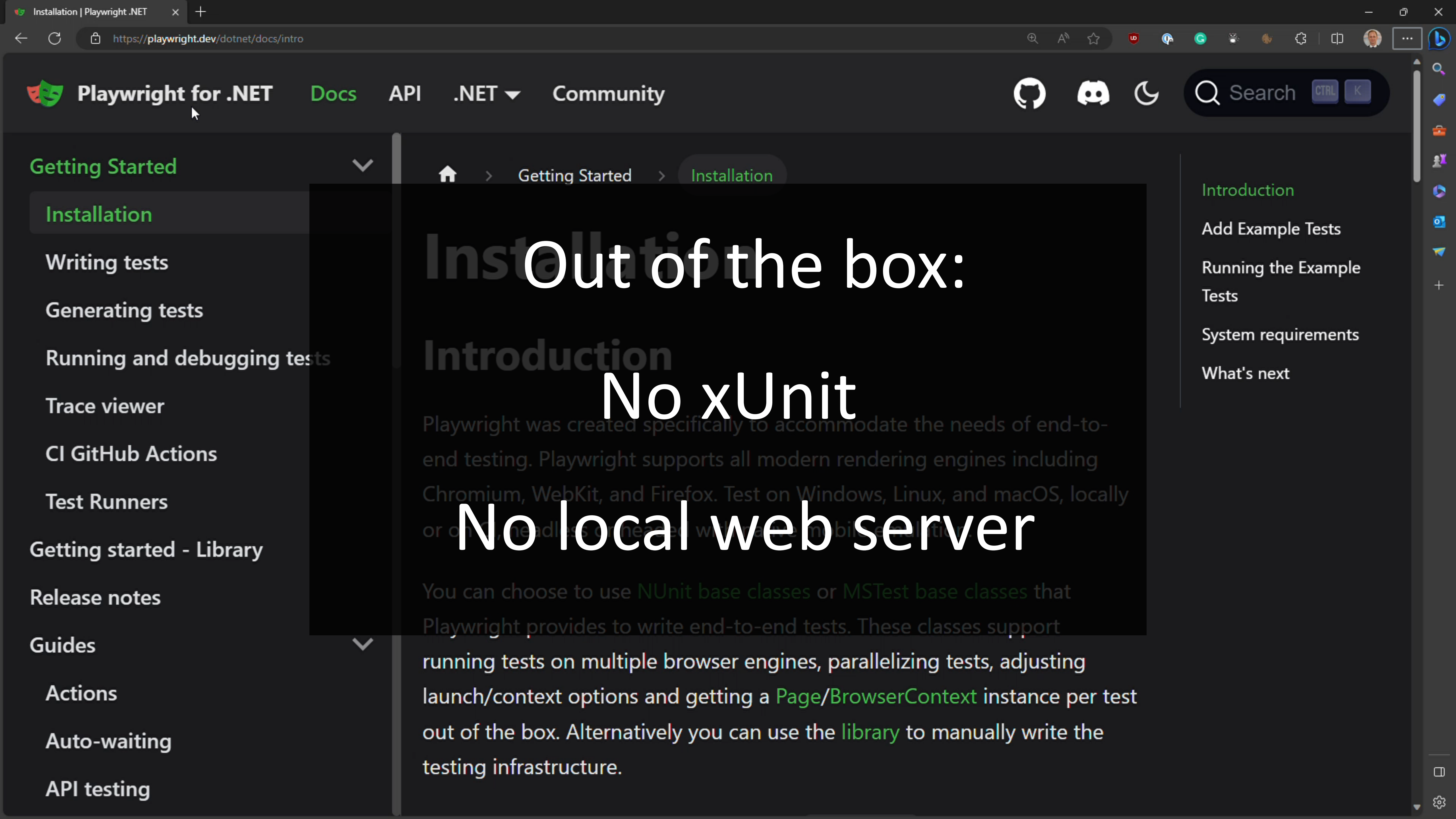
## Fast feedback

## Maintainable

## End-2-end testing

Playwright

playwright.dev/dotnet



Installation | Playwright .NET

https://playwright.dev/dotnet/docs/intro

Playwright for .NET Docs API .NET Community Search CTRL K

Getting Started
- Installation
- Writing tests
- Generating tests
- Running and debugging tests
- Trace viewer
- CI GitHub Actions
- Test Runners

Getting started - Library

Release notes

Guides
- Actions
- Auto-waiting
- API testing

Getting Started › Installation

# Out of the box:

# No xUnit

# No local web server

Installation

Introduction

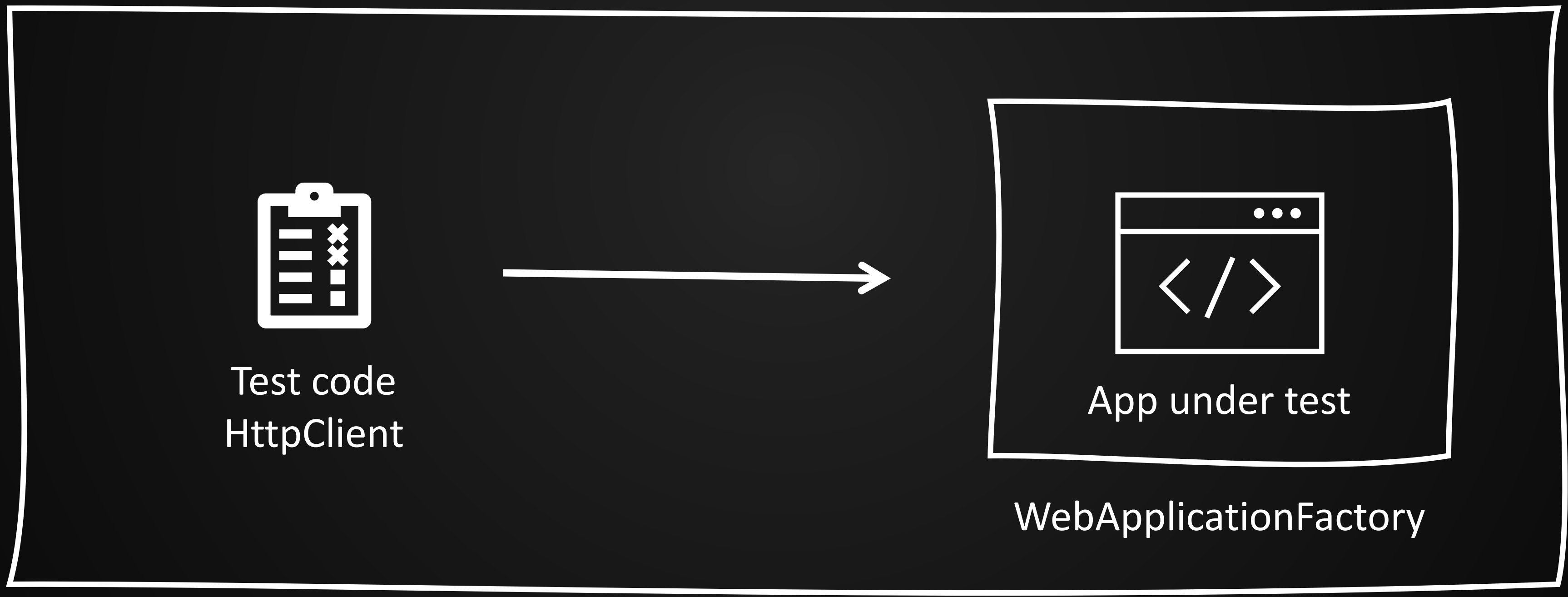
Playwright was created specifically to accommodate the needs of end-to-end testing. Playwright supports all modern rendering engines including Chromium, WebKit, and Firefox. Test on Windows, Linux, and macOS, locally or on CI, headless or headed with native mobile emulation.

You can choose to use NUnit base classes or MSTest base classes that Playwright provides to write end-to-end tests. These classes support running tests on multiple browser engines, parallelizing tests, adjusting launch/context options and getting a Page/BrowserContext instance per test out of the box. Alternatively you can use the library to manually write the testing infrastructure.

- Introduction
- Add Example Tests
- Running the Example Tests
- System requirements
- What's next

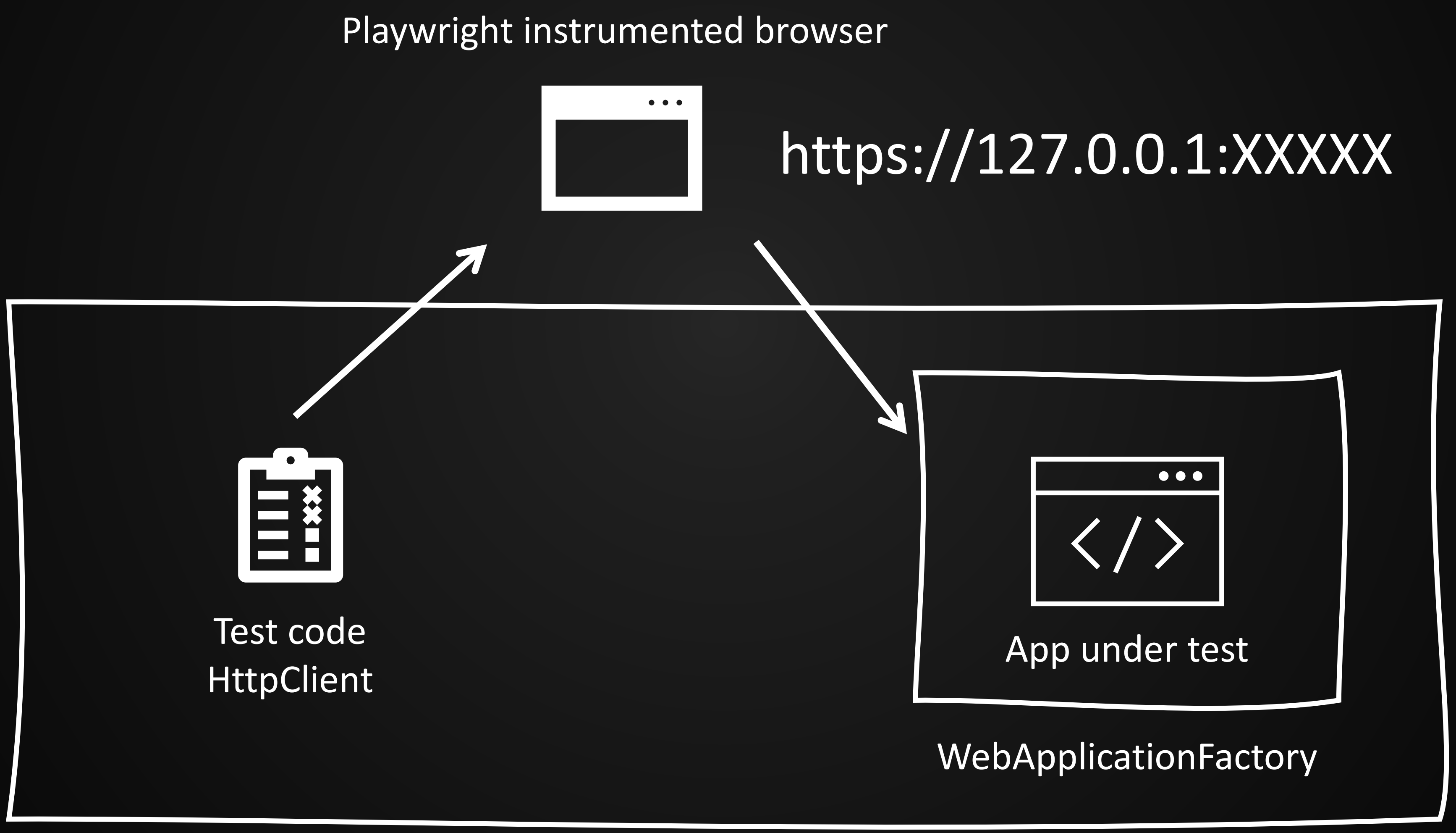
## WebApplicationFactory to the rescue

# WebApplicationFactory to the rescue



Test running everything in processes

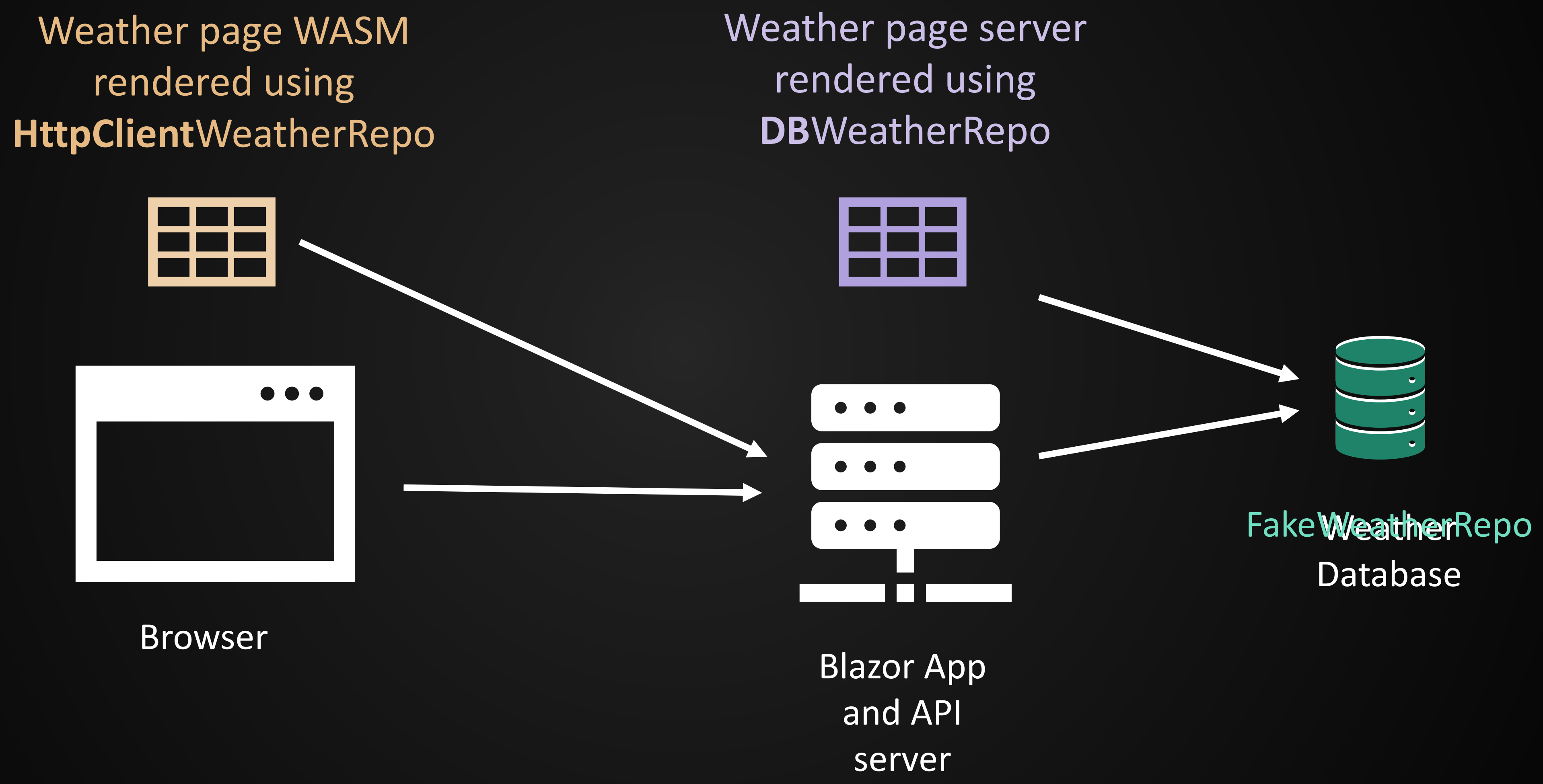
# Browser is out of process



## Pre-baked demo setup

```
dotnet new blazor -o NdcDemo
dotnet new nunit -o NdcDemo.Tests
dotnet new sln
dotnet sln add .\NdcDemo\
dotnet sln add .\NdcDemo.Tests\
cd .\NdcDemo.Tests\
dotnet add reference ..\NdcDemo\
dotnet add package Microsoft.Playwright
dotnet add package Microsoft.AspNetCore.Mvc.Testing
dotnet add package Verify.Playwright
dotnet add package Verify.NUnit
dotnet build
.\bin\Debug\net8.0\playwright.ps1 install
```

## End-2-end testing demo


Weather page WASM rendered using **HttpClient**WeatherRepo
Weather page server rendered using **DB**WeatherRepo
Browser
Blazor App and API server
FakeWeatherRepo
Weather Database

## End-2-end testing score card

Protection against regressions: **High**

Resistance to refactoring: **High**

Fast feedback: **Low**

Maintainable: **Medium**

## Component- or unit-testing

bUnit

bunit.dev

## Pre-baked demo setup

```
dotnet add package bunit
dotnet add package Verify.Bunit
```

## Component testing demo

## Component testing score card

Protection against regressions: **Medium**

Resistance to refactoring: **High**

Fast feedback: **High**

Maintainable: **High**

## What to pick?

| | End-2-end | Component |
|---|---|---|
| Regression | **High** | **Medium** |
| Refactoring | **High** | **High** |
| Feedback | **Low** | **High** |
| Maintainability | **Medium** | **High** |

## What to pick?

| | End-2-end | Component |
|---|---|---|
| Regression | **High** | **Medium** |
| Refactoring | **High** | **High** |
| Feedback | **Low** | **High** |
| Maintainability | **Medium** | **High** |

## Learn more:

Playwright: https://playwright.dev/dotnet

bUnit: https://bunit.dev

Verify: https://github.com/VerifyTests/Verify

NdcDemo: https://github.com/egil/NdcDemo

@**egil**@mastodon.social
**@egilhansen**
github.com/**egil**

Sponsor shout-out

Progress Telerik

Syncfusion

Amazon Web Services