

SETTING UP FEATURE FLAGS WITH .NET

Talia Nassi
@talial_nassi

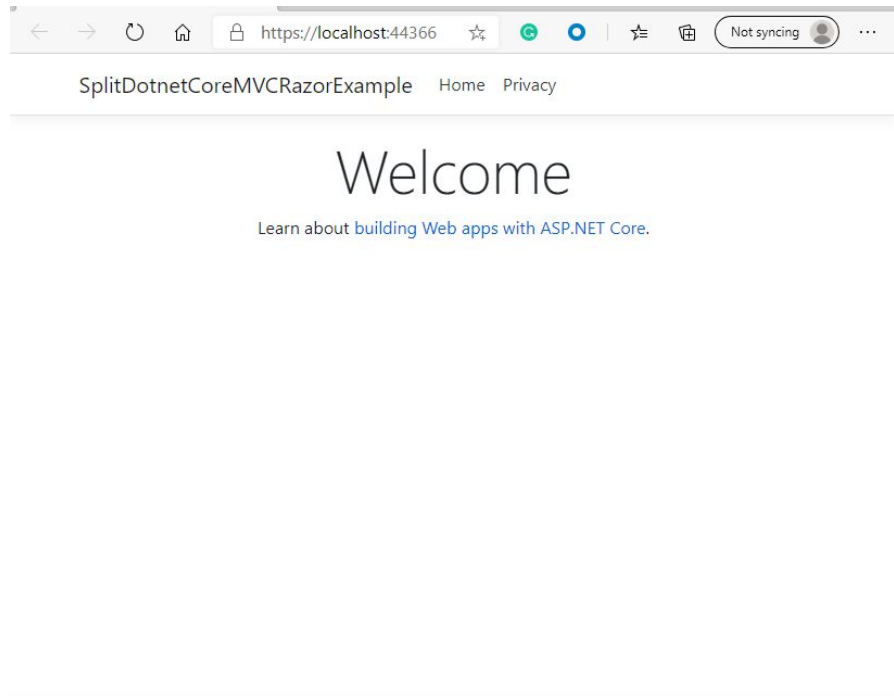


FEATURE FLAG USE CASES

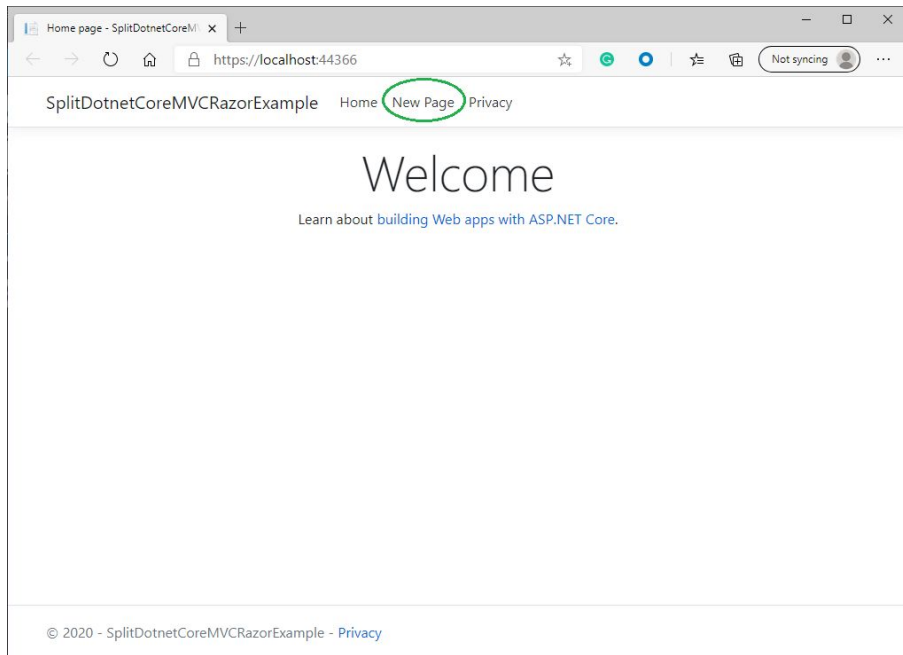
- TESTING IN PROD
- KILL SWITCH
- MIGRATION TO MICROSERVICES
- A/B TESTING
- SUBSCRIPTION MANAGEMENT
- CANARY RELEASES
- EXPERIMENTATION

Why do you care?

- Minimizes risk
- Maximizes impact



TREATMENT = OFF



TREATMENT = ON

Process



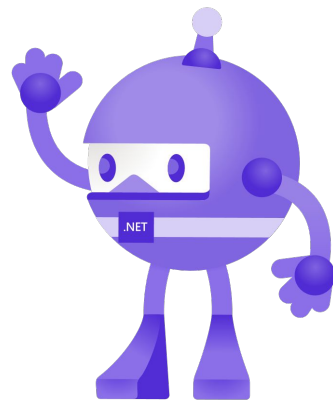
CREATE A
FEATURE
FLAG

INSTALL
DEPENDENCIES
AND
CONFIGURE
YOUR .NET APP

INSTANTIATE
AND USE THE
.NET CORE
SPLIT SDK

01

CREATE THE FEATURE FLAG



TARGET



Splits



Segments

MEASURE



Metrics

REVIEW



Data hub

No splits created

Splits allow you to feature flag and experiment on your product.



Separate code deploy from feature release & understand the impact of your features on the metrics you care about!

Create split

Create a split

Once saved, the split name and traffic type cannot be changed.

Name Use this name in your code.

v2_navigation_view

Traffic Type

user

Owners

Administrators

quorralyne x

Start typing a user or group. Use owners to organize your splits by team.

Tags (optional)

landing page x

Start typing a tag. Use tags to organize your splits by team, feature, or however you'd like.

Description (optional)

Add a description. As best practice, it's good to always add a description to explain the functionality controlled by this split.



Cancel

Create

Splits

Create a feature flag or experiment in your application.

Create split

OWNED BY ME

STARRED BY ME

SEARCH

Sort: Name A → Z

Filter

Search split name

v2_navigation_view

☆ | Traffic Type: User | Tags: landing ...


v2_navigation_view

Enter description

☆ | Traffic Type: User | Tags: landing page

Environment: Prod-Default

Targeting rulesLive tailAudit logsAlerts



No targeting rules have been defined

Add targeting rules to the **Prod-Default** environment to set your treatments and define your targeting criteria for this Split.

Define new targeting rules

Add Rules

or

Copy targeting rules from

Select environment

DE

Search

My Work

Environments

TARGET

Splits

Segments

MEASURE

Metrics

REVIEW

Data hub

Segments

Define a set of your customers to be reused across your splits.

Create segment

OWNED BY MESTARRED BY MESearch

Sort: Name A→ZFilter

Search segment name

QA

☆ | Traffic Type: User | Tags: qa

QA

Enter description

☆ | Traffic Type: User | Tags: qa

EnvironmentProd-Default

DefinitionAudit logs

More people in Split? Limit who can do what

Your plan does not include permissions. [Contact us](#) to upgrade your plan to limit who can edit what across your organization.

▼ Current Splits (0)

This segment is not being used in any Splits. [Learn more](#)

▼ User (1)

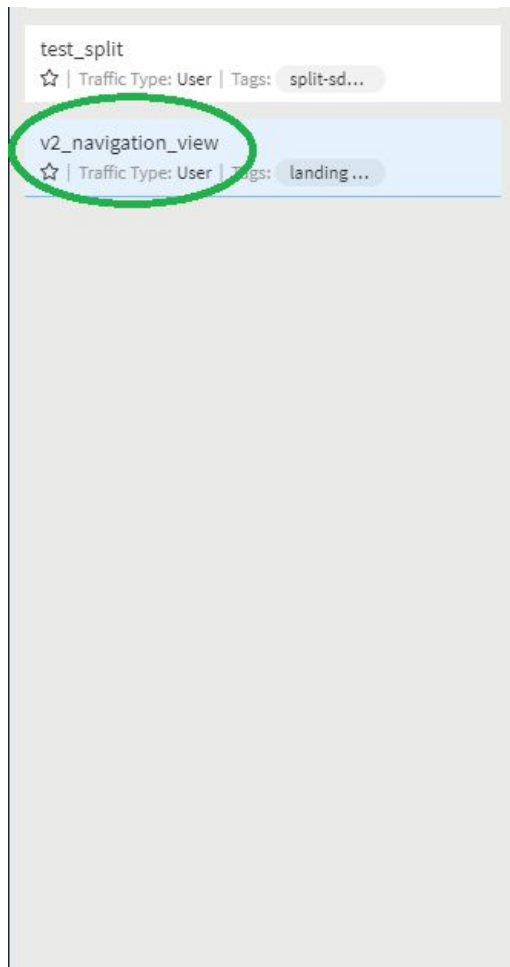
Add userAddImport

Added

User

11/11/2020 11:06...quorralyne

1-1 of 1



▼ Create whitelists

Create a whitelist or select a segment to place specific customers (e.g. your QA team, internal team, or a limited beta) into a

Add whitelist

› Allocate traffic

Advanced: If you're running an experiment and want to limit the number of customers that are exposed to it, you can select

▼ Set targeting rules

Target specific subsets of your customers based off of any attribute you want. (e.g. location or last-login date) These subsets between your treatments based off of percentages you decide.

A screenshot of a targeting rule configuration form. The form has two rows. The first row is for the 'IF' condition, with 'user' selected, 'Add attribute (optional)' as a placeholder, 'is in segment' as the operator, and 'QA' as the value. The second row is for the 'serve' action, with a blue toggle set to 'on' and a dropdown arrow. A blue 'Add rule' button is at the bottom. Green circles highlight the 'QA' value and the 'Add rule' button. A green oval highlights the 'serve' row.

▼ Set the default rule

For any of your customers that weren't assigned a treatment in the sections above, use this section to place them into a treat

serve

off

▼ Set the default rule

For any of your customers that weren't assigned a treatment in the sections above, use this section to place them into a treatment or randomly distribute these customers between your treatments/variations based off of percentages you decide.

serve ☐ off ▼

▼ Set the default treatment

Select a fallback treatment that will be assigned if the split is killed and for customers excluded from the experiment in the "Allocate Traffic" section.

If Split is **killed** or user is not included in **traffic allocation**, serve default treatment

☐ o.. ▼

Change

- > Treatments
- > Status
- > Seed
- > Default treatment
- > Whitelist user in on
- > Whitelist segment in on
- > Whitelist user in off
- > Whitelist segment in off
- > Traffic allocation

✓ Targeting Rules

- + if user is in segment QA
- + then split
- + 100% on
- + 0% off

> Default Rule

Title (optional)

Add a title

Comments (optional)

Describe and explain the change you're making

Add approvers (Your plan doesn't include approvals. [Contact us](#) to upgrade your plan.)

Start typing users or groups

Pro tip: Two sets of eyes are better than one. If you don't have approvers, your changes will

Cancel

Confirm

v2_navigation_view

Enter description



Traffic Type: User

Tags:

landing page



Owners:

Administrators

quorralyne



Environment

Prod-Default



Targeting rules

Live tail

Audit logs

Alerts

Metrics impact

Save changes

Kill



Syntax

Reallocate

Delete targeting rules



More people in Split? Limit who can do what

Your plan does not include permissions. [Contact us](#) to upgrade your plan to limit who can edit what across your organization.

Syntax

Language **.NET**  [Learn more about the .NET SDK](#)

```
using S
Classes;

/*
 * The
 * when
 */
var con
.NET Options();

var fac
Nodejs ("7i4qu1bpl8cpq42tdrh5ci7lfei22atd3quq", config);
var spl
PHP nt();
try
{
    spl
    Python y(10000);
}
catch (
{
    // log & handle
}
```

```
using Splitio.Services.Client.Classes;

/*
 * The SplitClient should be instantiated as a singleton and injected
 * wherever a new feature is being rolled-out.
 */
var config = new ConfigurationOptions();

var factory = new SplitFactory("7i4qu1bpl8cpq42tdrh5ci7lfei22atd3quq", config);
var splitClient = factory.Client();
try
{
    splitClient.BlockUntilReady(10000);
}
catch (Exception ex)
{
    // log & handle
}

var treatment = splitClient.GetTreatment(user_id, // unique identifier for your user
                                         "v2_navigation_view");

if (treatment == "on")
{
    // insert on code here
}
else if (treatment == "off")
{
    // insert off code here
}
else
{
    // insert control code here
}
```


02

INSTALL DEPENDENCIES
AND CONFIGURE YOUR
.NET CORE

WEB APP WITH RAZOR
PAGES



Create a new project

Search for templates (Alt+S)



Clear all

Recent project templates



Blazor App

C#



ASP.NET Core Web Application

C#



ASP.NET Web Application (.NET Framework)

C#



Android App (Xamarin)

C#



Azure Functions

C#



AWS Lambda Project with Tests (.NET Core - C#)

C#



Console App (.NET Core)

A project for creating a command-line application that can run on .NET Core on Windows, Linux and MacOS.

C#

Linux

macOS

Windows

Console



ASP.NET Core Web Application

Project templates for creating ASP.NET Core web apps and web APIs for Windows, Linux and macOS using .NET Core or .NET Framework. Create web apps with Razor Pages, MVC, or Single Page Apps (SPA) using Angular, React, or React + Redux.

C#

Linux

macOS

Windows

Cloud

Service

Web



Blazor App

Project templates for creating Blazor apps that run on the server in an ASP.NET Core app or in the browser on WebAssembly (wasm). These templates can be used to build web apps with rich dynamic user interfaces (UIs).

C#

Linux

macOS

Windows

Cloud

Web



Class Library (.NET Standard)

A project for creating a class library that targets .NET Standard.

C#

Android

iOS

Linux

macOS

Windows

Library

Back

Next

Configure your new project

ASP.NET Core Web Application

C#

Linux

macOS

Windows

Cloud

Service

Web


Project name

SplitDotnetCoreMVC RazorExample

Location

C:\Users\quorr\source\repos

...

Solution name 

SplitDotnetCoreMVC RazorExample

☒ Place solution and project in the same directory

Back

Create

Create a new ASP.NET Core web application

.NET Core

ASP.NET Core 3.1



Empty

An empty project template for creating an ASP.NET Core application. This template does not have any content in it.



API

A project template for creating an ASP.NET Core application with an example Controller for a RESTful HTTP service. This template can also be used for ASP.NET Core MVC Views and Controllers.



Web Application

A project template for creating an ASP.NET Core application with example ASP.NET Razor Pages content.



Web Application (Model-View-Controller)

A project template for creating an ASP.NET Core application with example ASP.NET Core MVC Views and Controllers. This template can also be used for RESTful HTTP services.



Angular

A project template for creating an ASP.NET Core application with Angular



React.js

[Get additional project templates](#)

Authentication

No Authentication

[Change](#)

Advanced

☒ Configure for HTTPS

☐ Enable Docker Support

(Requires [Docker Desktop](#))

Linux

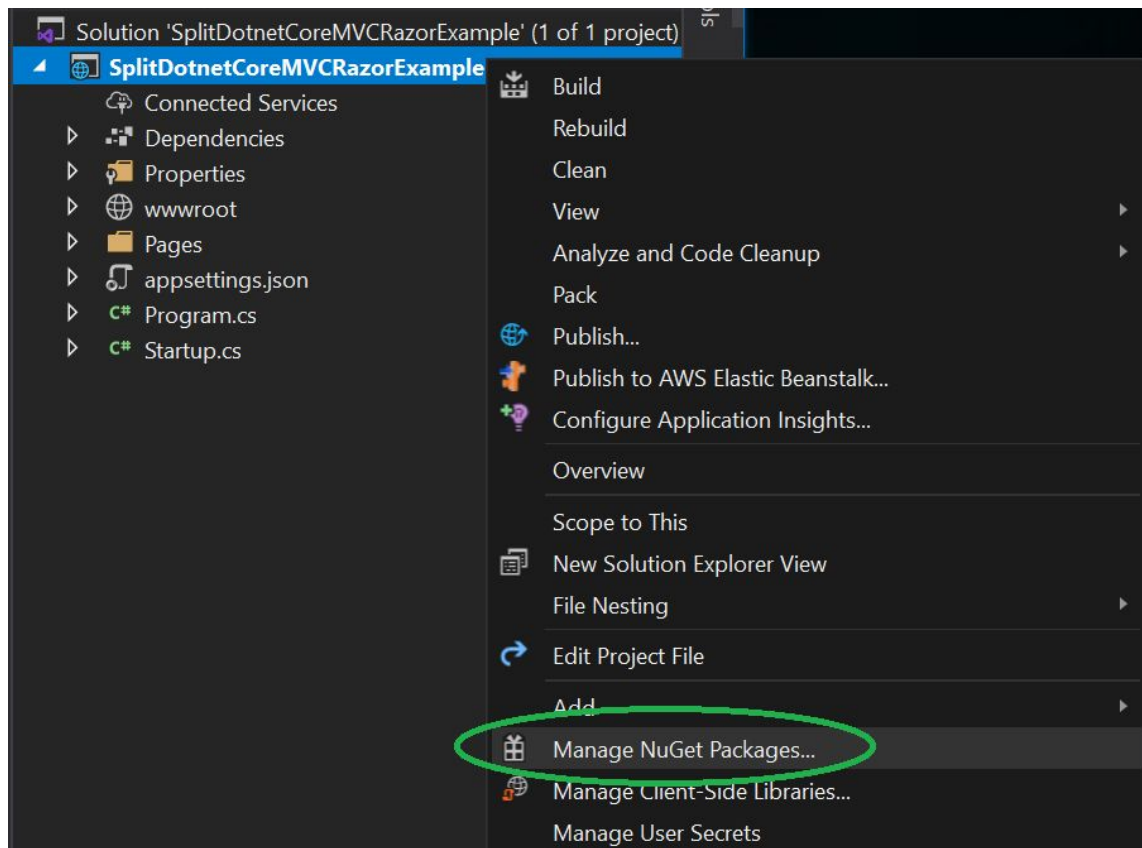
☐ Enable Razor runtime compilation

Author: Microsoft

Source: Templates 3.1.10

Back

Create



Browse

Installed

Updates

NuGet Package Manager: SplitDotnetCoreMVC RazorExample

Splitio-net-core



Include prerelease

Package source: nuget.org



Splitio-net-core by Splitio-net-core, 241K downloads

v6.2.0

Package Description



Splitio-net-core.Redis by Splitio-net-core.Redis, 20.9K downloads

v6.2.0

Package Description



Microsoft.AspNetCore.Http by Microsoft, 287M downloads

v2.2.2

ASP.NET Core default HTTP feature implementations.



Microsoft.AspNetCore.Http.Features by Microsoft, 345M downloads

v5.0.0

ASP.NET Core HTTP feature interface definitions.



Microsoft.AspNetCore.Http.Abstractions by Microsoft, 332M downloads

v2.2.0



Splitio-net-core

nuget.org

Version: Latest stable 6.2.0

Install

Options

Description

Package Description

Version: 6.2.0

Author(s): Splitio-net-core

Date published: Tuesday, October 13, 2020 (10/13/2020)

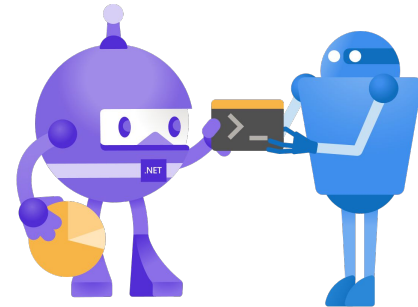
Report Abuse: <https://www.nuget.org/packages/Splitio-net-core/6.2.0/ReportAbuse>

Welcome

Learn about [building Web apps with ASP.NET Core](#).

03

INSTANTIATE AND USE THE SDK

















Installed

Visual C#

ASP.NET Core
Xamarin.Forms

Online

Sort by: Default

	Class	Visual C#
	Interface	Visual C#
	Razor Component	Visual C#
	Controller Class - Empty	Visual C#
	Controller Class with read/write actions	Visual C#
	API Controller Class - Empty	Visual C#
	API Controller Class with read/write actions	Visual C#
	Razor Page	Visual C#
	Razor View - Empty	Visual C#
	Content Page	Visual C#
	Content Page (C#)	Visual C#
	Content View	Visual C#
	Content View (C#)	Visual C#
	Partial View	Visual C#

Search (Ctrl+E)

Type: Visual C#

A Razor page with a page model

Name:

NewPage.cshhtml

Add

Cancel

Properties

wwwroot

Pages

Shared

_Layout

_Validation

_ViewImports

_ViewStart

Error.cshtml

Index.cshtml

Privacy.cshtml

appsettings.js

Program.cs

Startup.cs

Solution Explorer Team

Properties

Folder Properties

Icon

isc

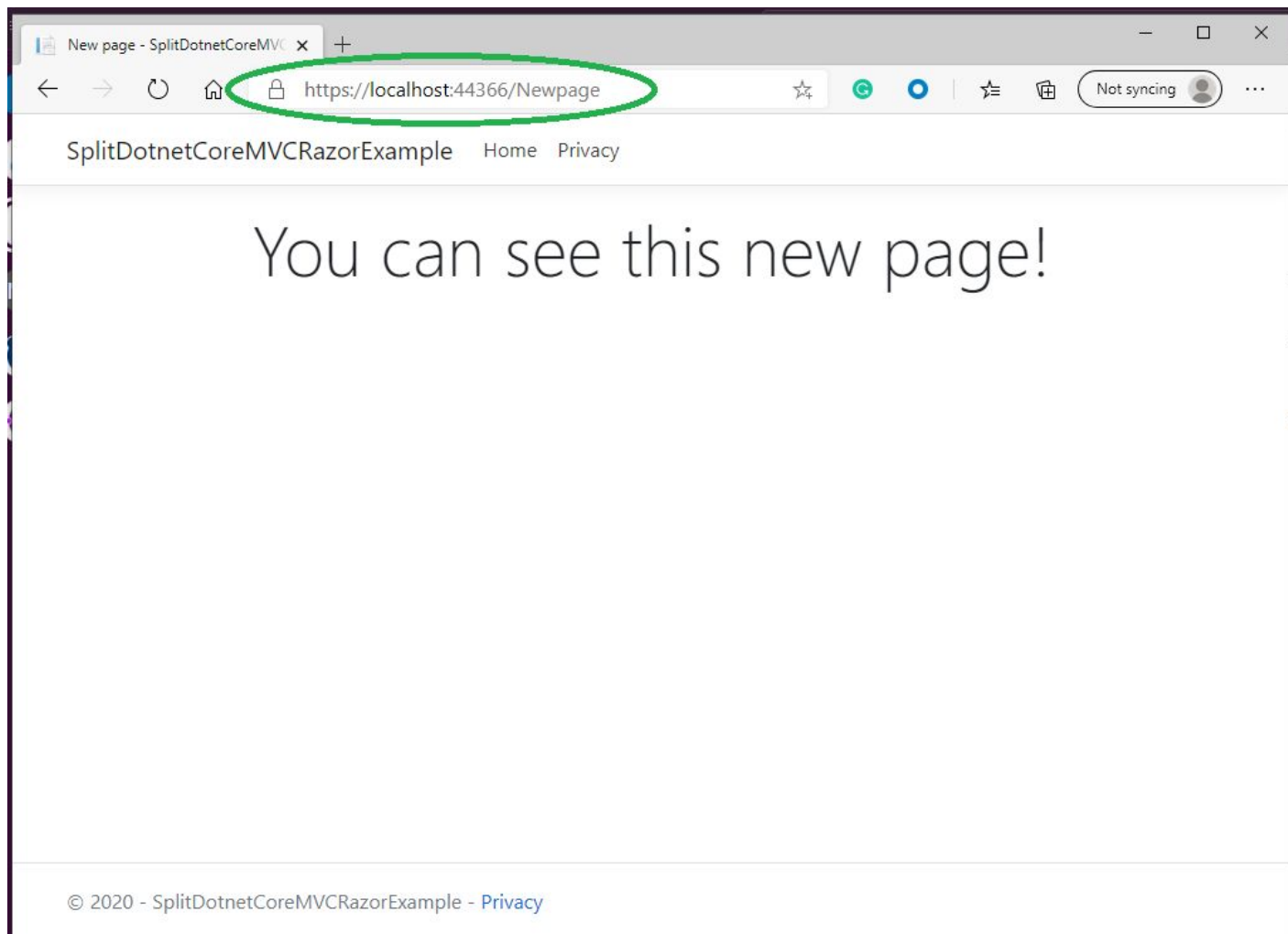
Folder Name

Full Path

Index.cshtml

NewPage.cshtml

```
1 @page
2 @model SplitDotnetCoreMVC RazorExample.Pages.NewPageModel
3 @{
4     ViewData["Title"] = "New page";
5 }
6
7 <div class="text-center">
8     <h1 class="display-4">You can see this new page!</h1>
9 </div>
10
11
```



Add New Item - SplitDotnetCoreMVC RazorExample

Installed















Sort by: Default

Search (Ctrl+E)

Visual C#

ASP.NET Core
Xamarin.Forms

Online

	Content Page	Visual C#
	Content Page (C#)	Visual C#
	Content View	Visual C#
	Content View (C#)	Visual C#
	List View Page	Visual C#
	Master Detail Page	Visual C#
	Razor Layout	Visual C#
	Tabbed Page	Visual C#
	View Cell	Visual C#
	Code File	Visual C#
	Razor View Start	Visual C#
	Razor View Imports	Visual C#
	Tag Helper Class	Visual C#
	Middleware Class	Visual C#

Type: Visual C#

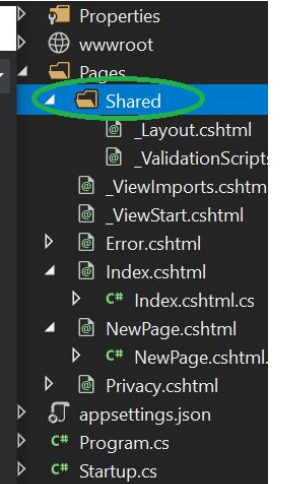
Razor View Layout Page

Name:

_LayoutV2.cshtml

Add

Cancel



Solution Explorer Team Explorer

Properties

Folder Properties

isc

Folder Name

Full Path

C:\U

```
<body>
  <header>
    <nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light bg-white border-bottom box-shadow mb-3">
      <div class="container">
        <a class="navbar-brand" asp-area="" asp-page="/Index">SplitDotnetCoreMVCRazorExample</a>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-target=".navbar-collapse"
          aria-expanded="false" aria-label="Toggle navigation">
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="navbar-collapse collapse d-sm-inline-flex flex-sm-row-reverse">
          <ul class="navbar-nav flex-grow-1">
            <li class="nav-item">
              <a class="nav-link text-dark" asp-area="" asp-page="/Index">Home</a>
            </li>
            <li class="nav-item">
              <a class="nav-link text-dark" asp-area="" asp-page="/NewPage">New Page</a>
            </li>
            <li class="nav-item">
              <a class="nav-link text-dark" asp-area="" asp-page="/Privacy">Privacy</a>
            </li>
          </ul>
        </div>
      </div>
    </nav>
  </header>
  <div class="container">
```

```
using Splitio.Services.Client.Classes;

namespace SplitDotnetCoreMVCRazorExample
{
    3 references
    public class Startup
    {
        public static bool V2NavigationIsActive = false;
        0 references
        public Startup(IConfiguration configuration)
```

```
public void ConfigureServices(IServiceCollection services)
{
    services.AddRazorPages();

    var config = new ConfigurationOptions();
    var factory = new SplitFactory("7i4qu1bpl8cpq42tdrh5ci7lfei22atd3quq", config);
    var splitClient = factory.Client();
    try
    {
        splitClient.BlockUntilReady(10000);
    }
    catch (Exception ex)
    {
        // log & handle
    }

    var treatment = splitClient.GetTreatment(Configuration.GetValue<string>("SplitUser"), "v2_navigation_view");
    if (treatment == "on")
    {
        V2NavigationIsActive = true;
    }
    else if (treatment == "off")
    {
        V2NavigationIsActive = false;
    }
}
```

```
_ViewStart.cshtml  Startup.cs  _LayoutV2.cshtml  _Layout.cshtml  NewPage.cshtml
1  @if (Startup.V2NavigationIsActive)
2  {
3      Layout = "_LayoutV2";
4
5  }
6  else
7  {
8      Layout = "_Layout";
9  }
```


appsettings.json _ViewStart.cshtml Startup.cs _LayoutV2.cshtml _Layout.cshtml

Schema: <https://json.schemastore.org/appsettings>

```
1  {
2  }
3  "Logging": {
4    "LogLevel": {
5      "Default": "Information",
6      "Microsoft": "Warning",
7      "Microsoft.Hosting.Lifetime": "Information"
8    }
9  },
10 "AllowedHosts": "*",
11 "SplitUser": "talía"
12 }
```

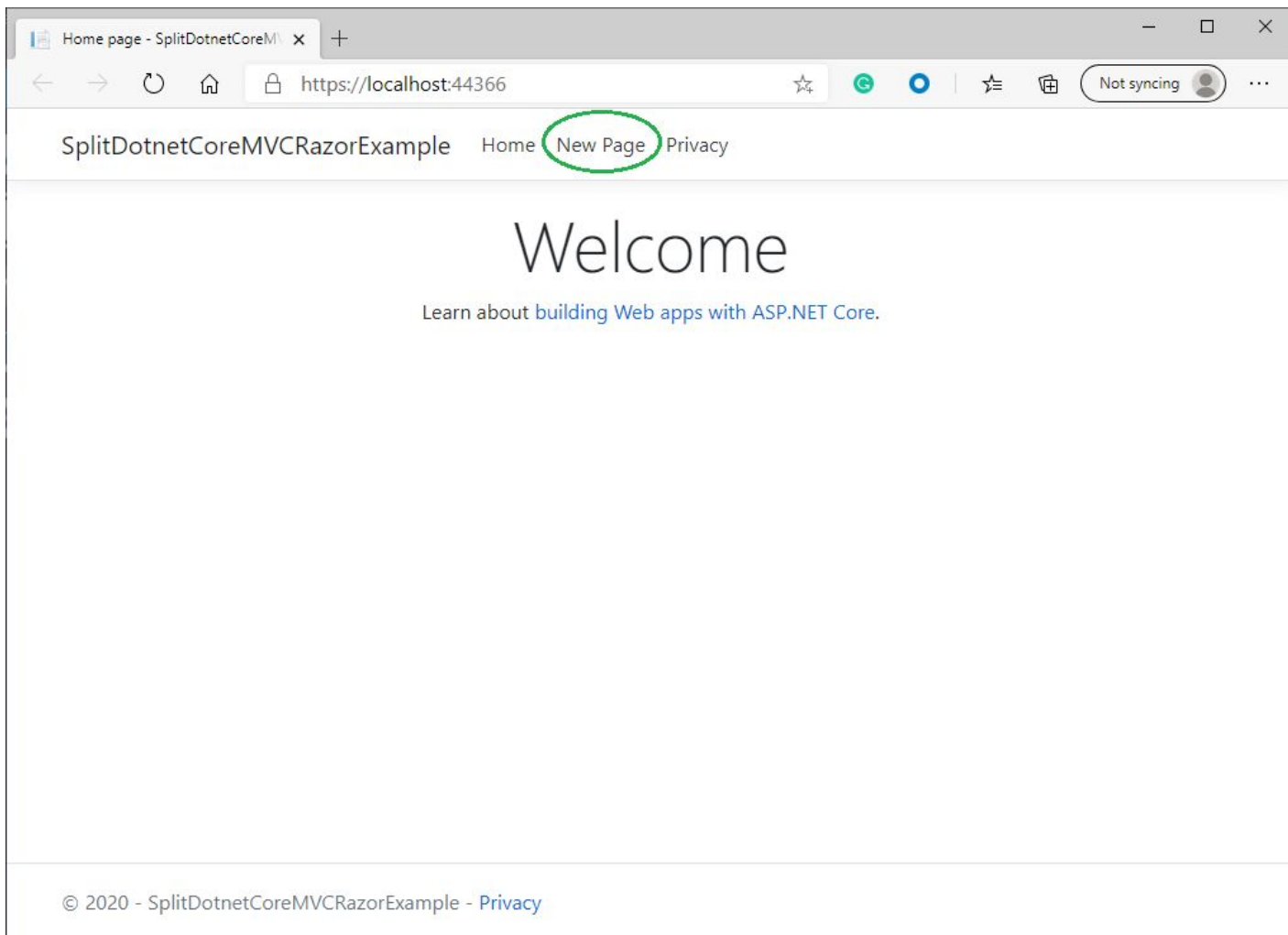
SplitDotnetCoreMVC RazorExample Home Privacy

Welcome

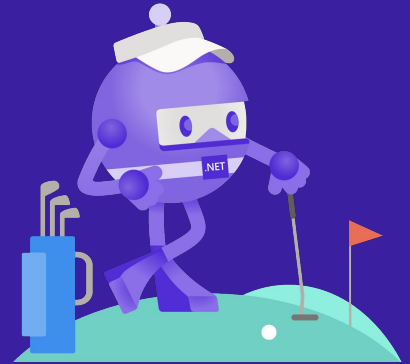
Learn about [building Web apps with ASP.NET Core](#).

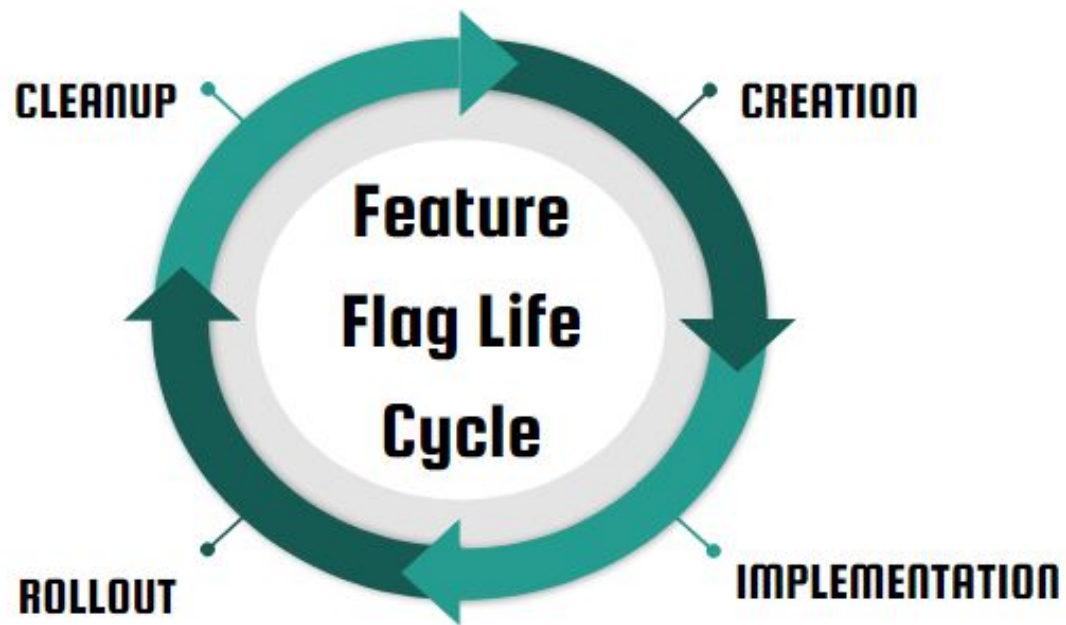
```
appsettings.json  X  _ViewStart.cshtml  Startup.cs  _LayoutV2.cshtml  _Layout.cshtml
Schema: https://json.schemastore.org/appsettings

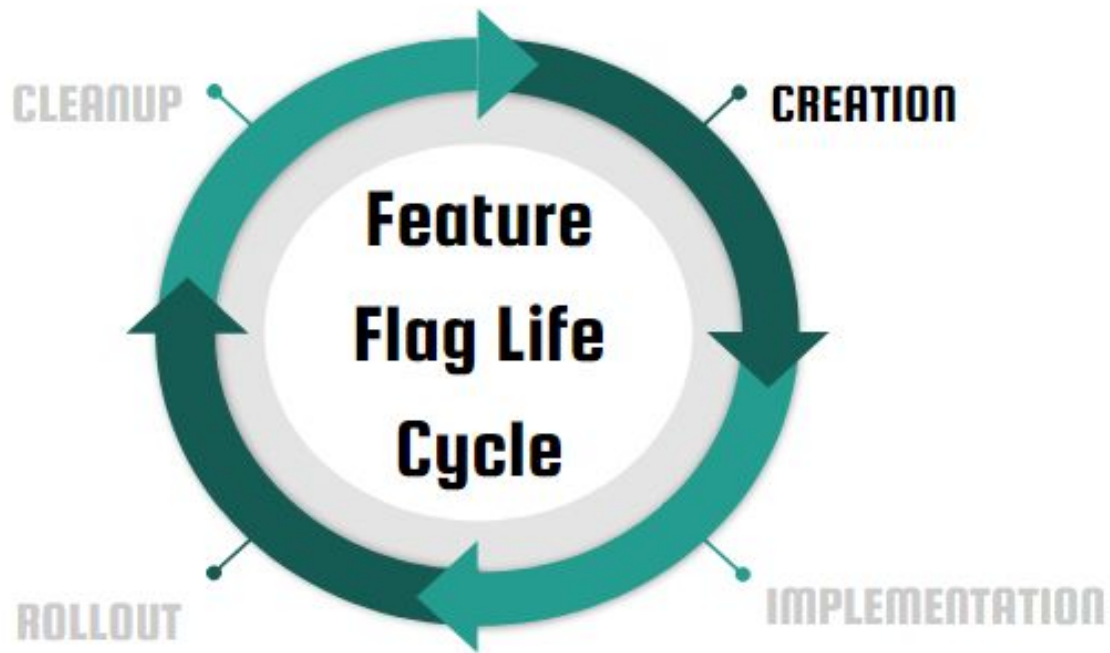
1  {
2  "Logging": {
3    "LogLevel": {
4      "Default": "Information",
5      "Microsoft": "Warning",
6      "Microsoft.Hosting.Lifetime": "Information"
7    }
8  },
9  "AllowedHosts": "*",
10 "SplitUser": "quorralyne"
11 }
12
```



Best Practices



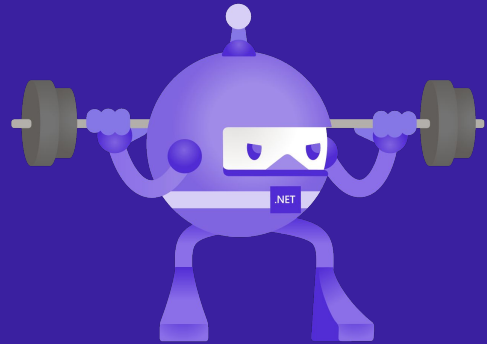


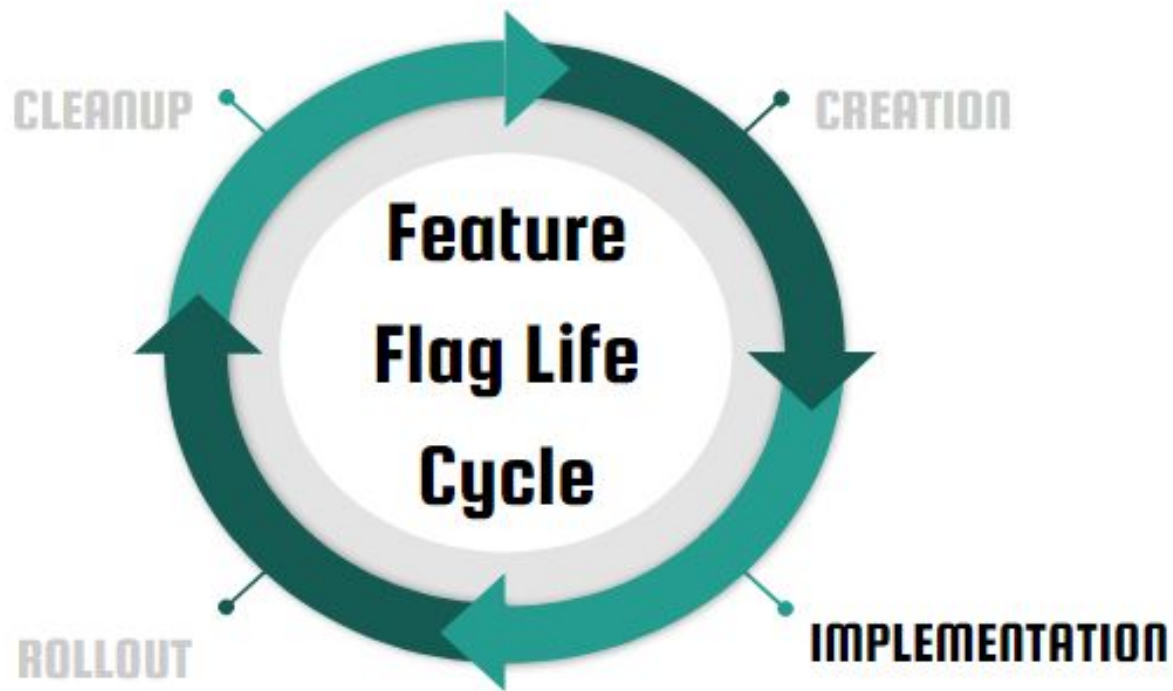


Creation

- INCLUDE TICKET NUMBER
- INCLUDE THE NAME OF THE SERVICE
- INCLUDE THE NAME OF THE TEAM THAT THE FLAG BELONGS TO
- INCLUDE THE NAME OF THE FUNCTIONALITY
- ENGAGE WITH THE DIFFERENT STAKEHOLDERS
- BE SURE THERE IS CLEAR OWNERSHIP

Consistency = Key





Implementation

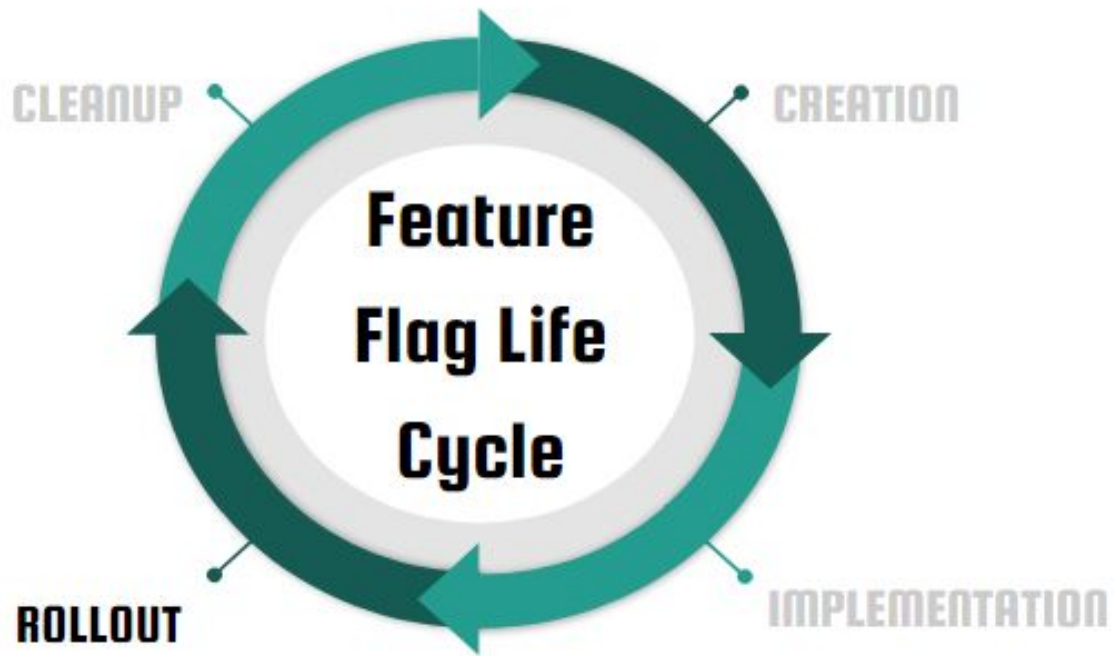
1. CHANGES TO FF SHOULD BE TREATED LIKE CODE DEPLOYS
2. MAKE FLAGGING DECISIONS ON THE SERVER
3. DON'T CHANGE THE SCOPE OF YOUR COMPONENTS JUST TO SUPPORT FEATURE FLAGGING DECISIONS

Implementation - Testing

Choose the most important flows to automate and test

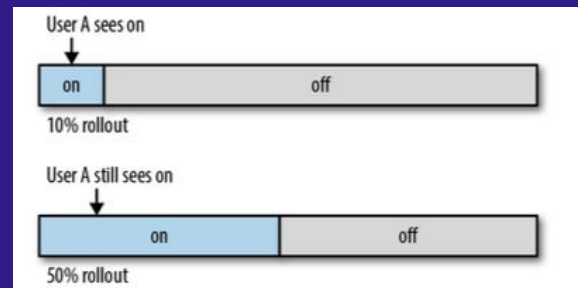
Work with your product owner to prioritize what these flows are

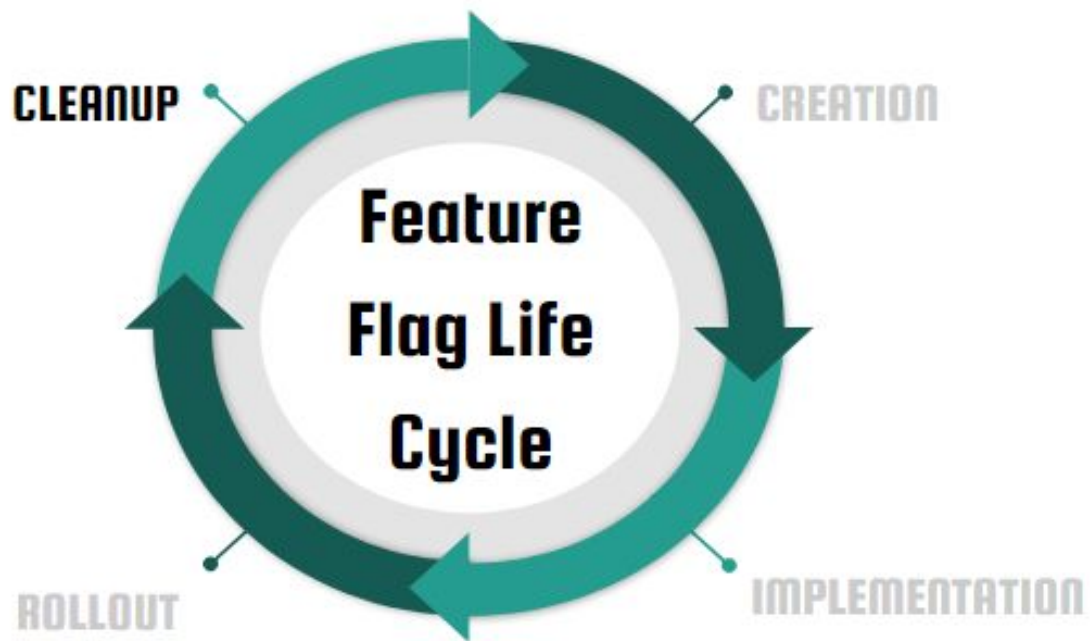
Target your automation bots inside the feature flags



Rollout

1. USERS SHOULD CONSISTENTLY HAVE THE SAME EXPERIENCE
2. BUILD A FEEDBACK LOOP
3. SET UP ALERTS TO PAGE YOU WHEN YOU GET ANY STATISTICALLY SIGNIFICANT CHANGES





Cleanup

1. ASSIGN AN EXPIRATION DATE FOR EACH FLAG
2. SET UP ALERTS THAT WILL POKE YOU WHEN A FLAG IS STALE
3. ADD A SUBTASK TO YOUR BACKLOG TO RETIRE THE FLAG AT THE SAME TIME THE FLAG IS CREATED
4. BE SURE TO REMOVE THE OLD CODE WHEN YOU DELETE THE FEATURE FLAG

Useful Links

Feature Flag Maintenance - <https://www.youtube.com/watch?v=qb-VNbMSzy0>

Free Split Account - www.split.io (try feature flagging for free!)

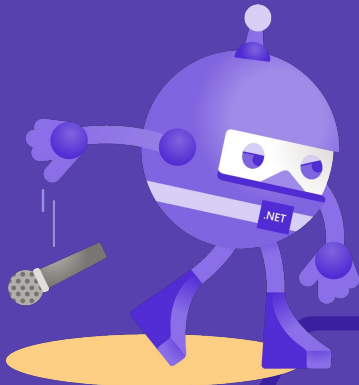
Link to .NET Example Repo - <https://bit.ly/3phMqsF>

TALIA NASSI

@talial_nassi

.NET Questions?
HEATHER DOWNING

@quorralyne



Thanks for joining!