Feature Flags

Lab Manual

Conditions and Terms of Use

Microsoft Confidential

This training package is proprietary and confidential, and is intended only for uses described in the training materials. Content and software is provided to you under a Non-Disclosure Agreement and cannot be distributed. Copying or disclosing all or any portion of the content and/or software included in such packages is strictly prohibited.

The contents of this package are for informational and training purposes only and are provided "as is" without warranty of any kind, whether express or implied, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, and non-infringement.

Training package content, including URLs and other Internet Web site references, is subject to change without notice. Because Microsoft must respond to changing market conditions, the content should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication. Unless otherwise noted, the companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred.

© 2017 Microsoft Corporation

Copyright and Trademarks

© 2017 Microsoft Corporation. All rights reserved.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

For more information, see Use of Microsoft Copyrighted Content at

http://www.microsoft.com/en-us/legal/intellectualproperty/Permissions/default.aspx

DirectX, Hyper-V, Internet Explorer, Microsoft, Outlook, OneDrive, SQL Server, Windows, Microsoft Azure, Windows PowerShell, Windows Server, Windows Vista, and Zune are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Other Microsoft products mentioned herein may be either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. All other trademarks are property of their respective owners

.

Contents

LAB : FEATURE FLAGS 6

EXERCISE 1: CREATE A FEATURE FLAG 7
EXERCISE 2: USE A FEATURE FLAG IN YOUR APPLICATION 10

© 2017 Microsoft Corporation

Microsoft Confidential

Lab : Feature Flags

Introduction

In this lab, you will work with LaunchDarkly to implement feature flags, exploring their available functionality.

Objectives

After completing this lab, you will be able to:

- Use Feature Flags effectively

Prerequisites

None

Estimated Time to Complete This Lab

30 minutes

For More Information

Microsoft and LaunchDarkly: https://launchdarkly.com/microsoft/

LaunchDarkly: https://launchdarkly.com/

© 2017 Microsoft Corporation

Exercise 1: Create a feature flag

Introduction

Feature flags are an important DevOps technique that makes the process of continuous delivery a lot easier. They help with the following aspects of the release process:

- Nearly eliminate integration (merge) conflicts.
- Toggle features to hide, disable, or enable features at run-time.
- Revert a change deployed to production without rolling back your release.
- Present users with variants of a feature, to determine which one performs better.

Objectives

After completing this lab, you will be able to:

- Understand and control feature flags to turn features on/off

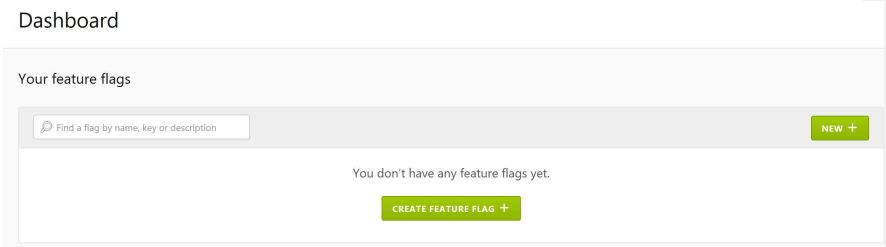
Prerequisites

None

Tasks

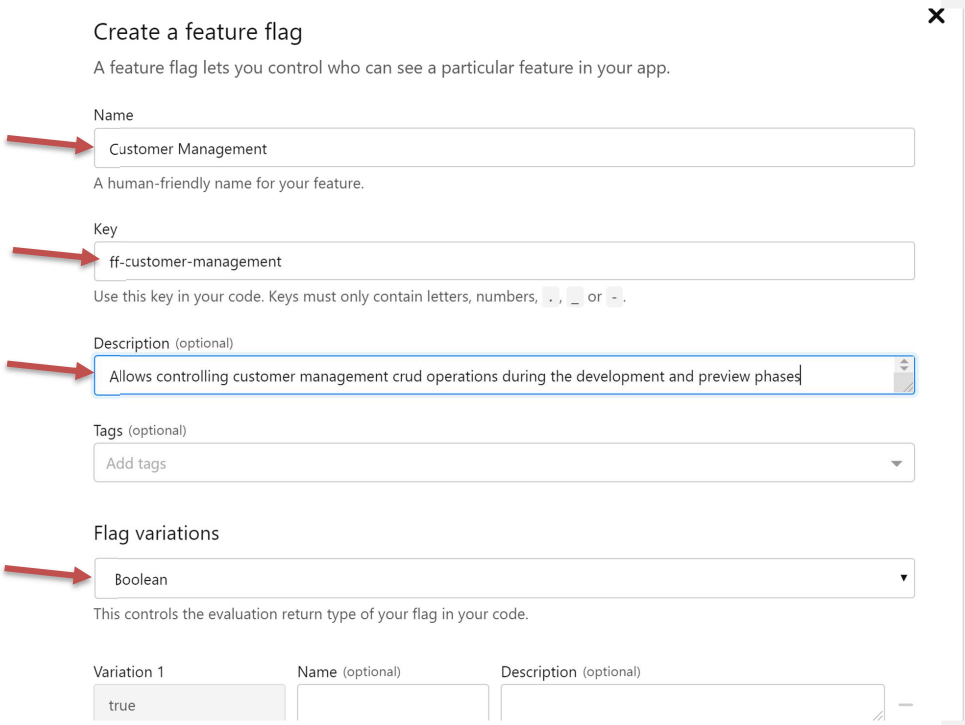
Implementing feature flags can either be achieved by building your own infrastructure or by leveraging one of the many products in the marketplace. In this hands-on lab, we have chosen the latter. We will use a product called LaunchDarkly. **Our choice is not a recommendation of the product**. You should evaluate the solution that best meet your needs.

- Go to this web site and sign up for a **free trial version**: https://launchdarkly.com/start/index.html#signup
- Open a shell window (Bash, Windows Command Line or PowerShell). You can use the integrated command line from within VS Code.
- Go to the solution root folder of your "ModernUISolution", and then to the "ModernUIApp" folder: "cd ModernAppUISolution"
  (Make sure this name matches the name you chose for the UI solution folder)
- From the "LaunchDarkly" portal add a new entry for the feature flag:



- Click on "Create feature flag", then enter the following information:

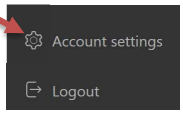
© 2017 Microsoft Corporation



Click "Save Flag".

Note: Make sure you take note of the feature flag key you entered in this screen, you will refer to it in the rest of this lab.

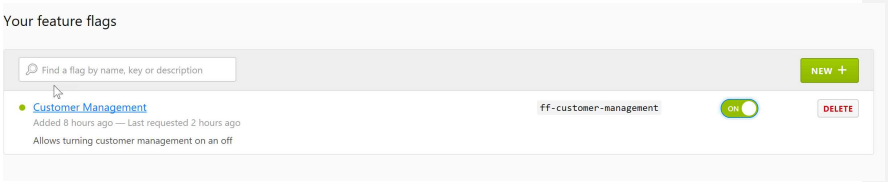
- From the "Launchdarkly" dashboard, go to the account settings.



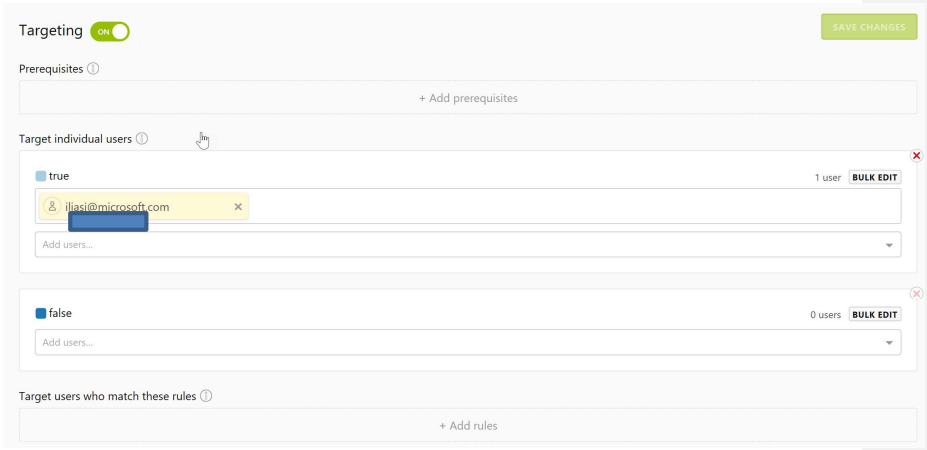
- Copy the production SDK key for the "Production" environment. You will need it to integrate the code with the feature flag:

- Go to the feature flags dashboard and click on the feature flag you created.

© 2017 Microsoft Corporation



- Click on the "Customer Management" feature flag:
- Enable targeting, and add a user in the true section (you can enter your own email address for instance):



- Make sure the default rule is false:

- Save the changes you've made.

© 2017 Microsoft Corporation

Exercise 2: Use a feature flag in your application

Objectives

After completing this lab, you will be able to:

- Integrate feature flags with your code.

Prerequisites

None

- Replace the content of your "ModernUIAppSolution" folder with the content of the "..\Exercice 2\ModernUIAppSolution" folder.
  We are now ready to test the full application interaction.
- Open a PowerShell window and go to the root folder containing both your "ModernApiAppSolution" and your "ModernUIAppSolution".
- Use the following command to open Visual Studio Code:
  Code .
- Open the "ui.Dockerfile" and replace the "LaunchDarklyKey" environment variable with the key you have copied in the previous step.

Startup.cs | ui.Dockerfile | CustomersController.cs

```
1  FROM microsoft/dotnet:2.1-sdk AS build
2  WORKDIR /app
3
4  # Copy csproj and restore as distinct layers
5  COPY . .
6  RUN dotnet restore /app/ModernUIApp/ModernUIApp.csproj
7  RUN dotnet publish /app/ModernUIApp/ModernUIApp.csproj --output /out/ --configuration Release
8
9
10
11
12 FROM microsoft/dotnet:2.1-aspnetcore-runtime AS runtime
13 WORKDIR /app
14 COPY --from=build /out .
15
16 ENV ApiURL http://modernapiapp/api/customers
17 ENV ASPNETCORE_ENVIRONMENT Development
18 ENV LauchDarklyKey sdk-c9[redacted]abc9a1a91eb7
19 ENV APPINSIGHTS_INSTRUMENTATIONKEY 553ee494-f1ee-444e-99af-b623e761c82e
20 EXPOSE 80
21
22 ENTRYPOINT ["dotnet", "ModernUIApp.dll"]
```

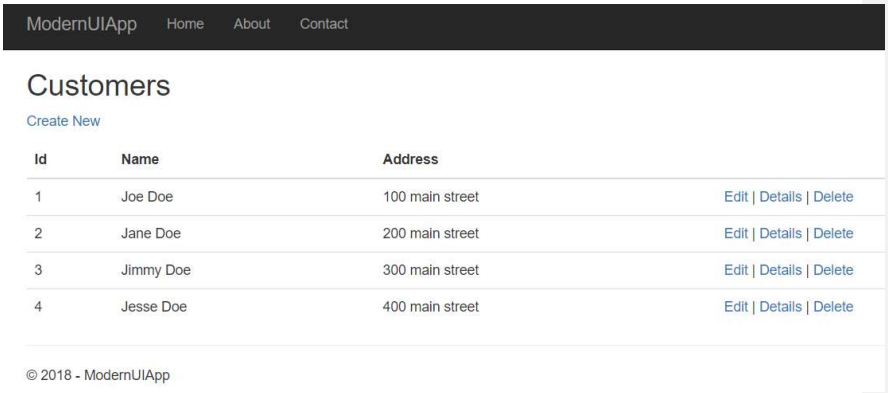
Commented [JB1]: I had issues with this and had to make certain the ASPNETCORE_ENVIRONMENT was the same as the environment defined in the Account Settings in Launch Darkly.

Commented [IJ2R1]: Indeed.
Need two keys for Dev and Prod.

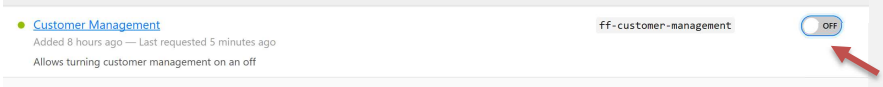
© 2017 Microsoft Corporation

- Change the username in the "CustomersController.cs"

- Run the following commands:
  ".\stopApiAndUi.ps1" (Please note that if there are not running containers, you will get an error message informing you that there aren't any running containers)
  ".\startApiAndUi.ps1"
- Once the application runs go to the page: http://localhost:9090/Customers, you should be able to see the list of customers:

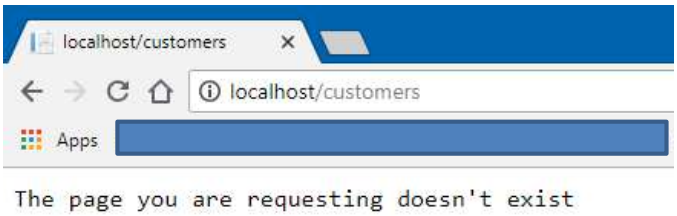


- Go to the Feature flag dashboard, and turn it off:



- Refresh the customers page in your application. You should now get the following response:

© 2017 Microsoft Corporation



You are now able to control the visibility of this feature directly from the launch darkly site by targeting specific users. Please note that in a real application, you will most likely get the user ids from the application itself instead of having them hardcoded, and you would use the JavaScript SDK as well so that the UI doesn't even offer a link to the page we requested if the feature is off.

Examine the code responsible of enabling the feature flag on and off. It is in the "CustomersController.cs" class in the "ModernUIApp" application:

```
public CustomersController(ICustomerDataService customerDataService,
IFeatureFlagService featureFlagService)
        {
            _customerDataService = customerDataService;
            _featureFlagService = featureFlagService;
        }
        public IActionResult Index()
        {
            if (_featureFlagService.ViewFeature(_userName))
            {
                var customers = _customerDataService.GetAll();
                return View(customers);
            }
            else
            {
                return Content("The page you are requesting doesn't exist");
            }
        }
```

© 2017 Microsoft Corporation