

LD API Homework

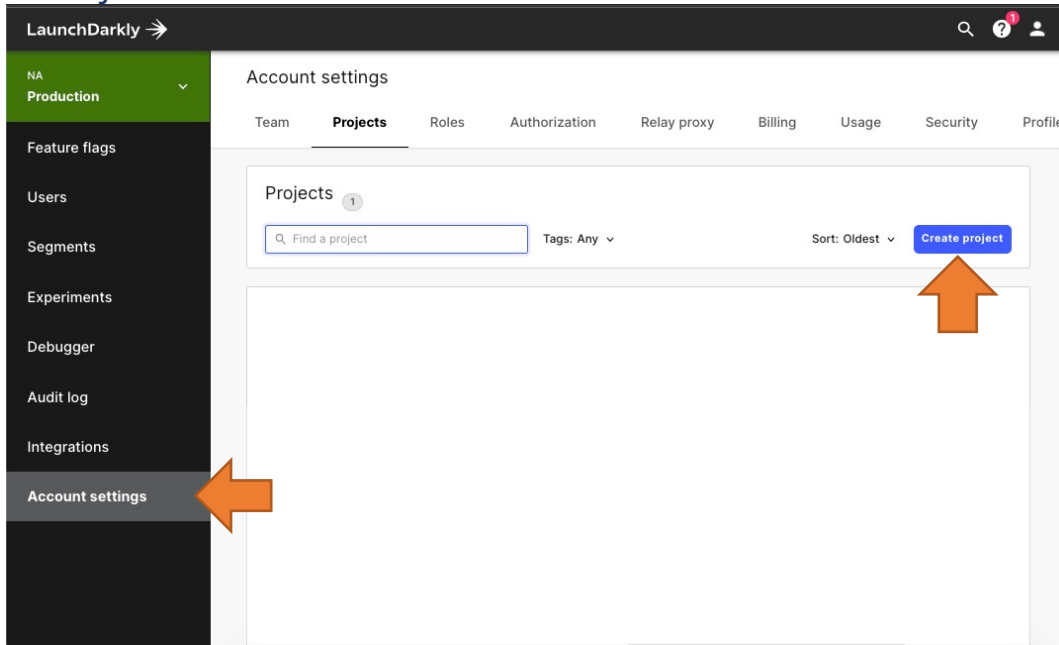
LaunchDarkly →

Amr Osman

Tuesday, December 14, 2021

Sample Execution Procedure

- 1- Download and copy [index_v2.html](#) from **GitHub** to your local drive
- 2- Create an account on Launch Darkly portal at [\[https://app.launchdarkly.com/signup\]](https://app.launchdarkly.com/signup)
- 3- Login to the Launch Darkly portal with your new credentials
- 4- Navigate to "Account Settings" → "Projects" and then press on the "Create Project" button.



- 5- Edit the project data: Name, Key, and select "SDK using client-side ID"

A screenshot of the 'Create a project' form in the LaunchDarkly portal. The form has a title 'Create a project' and a close button. It contains the following fields and sections:

- Name:** A text input field.
- Key:** A text input field. Below it, a note states: 'We use the key to give you friendly URLs. Keys should be at most 20 characters and must only contain letters, numbers, ., _ or -. You cannot use new as a key.'
- Tags:** A dropdown menu with the text 'Add tags'.
- Default client-side SDK availability:** A section with the text 'Control which client-side SDKs can use new flags by default. To learn more, [read the documentation](#).' It contains two checkboxes: 'SDKs using Mobile key' (unchecked) and 'SDKs using Client-side ID' (checked).
- SDKs:** A grid of buttons for various SDKs: Android, Flutter, Roku, Electron, JavaScript, React Native, React, Node.js (client), C/C++ (client), iOS, and Xamarin.
- Save project:** A blue button at the bottom.

6- It's time to create the flags that will be used into your application. Go to "Feature Flag" → "Create Flag"

The screenshot shows the LaunchDarkly web interface. At the top, the 'LaunchDarkly' logo is on the left, and search, help, and user icons are on the right. The left sidebar contains a list of navigation items: 'NA Production' (highlighted in green), 'Feature flags' (highlighted in grey with an orange arrow pointing to it), 'Users', 'Segments', 'Experiments', 'Debugger', 'Audit log', 'Integrations', and 'Account settings'. The main content area is titled 'Feature flags / Maintained by me' and includes a 'Save' button. Below the title, there is a description: 'Use this page to see all feature flags in this project. Select a flag to manage the environment-specific targeting and rollout rules.' The main content area also displays '2 flags' and a search bar with the placeholder text 'Find a flag by name, key or description'. There are also filters for 'Filter (1)' and 'Sort: Newest', and a 'Create flag' button (highlighted with an orange arrow) next to a menu icon. A 'Maintainer: Me' dropdown is also visible.

7- Enter the flag data and press "Save" button

Create a feature flag

A feature flag lets you control who can see a particular feature in your app.

Name

A human-friendly name for your feature.

Key

Use this key in your code. Keys must only contain letters, numbers, `-`, `.` or `_`.
You cannot use `new` as a key.

Description (optional)

Describe what this feature flag controls

Tags (optional)

Add tags

Client-side SDK availability

Control which client-side SDKs can use this flag. To learn more, [read the documentation](#).

☐ SDKs using Mobile key ☒ SDKs using Client-side ID

AndroidFlutterRokuReact Native

C/C++ (client)IOSXamarin

ElectronJavaScriptReact

Node.js (client)

Flag variations

Boolean

This controls the evaluation return type of your flag in your code.

Variation 1

Name (optional)

Description (optional)

true

Variation 2

Name (optional)

Description (optional)

false

Default variations

ON

true

OFF

false

☐ This is a permanent flag

Save flag

- 8- Create 2 variations (postpone this step until you run your code for the first time). Click on any one of the new Flags that you just created, and set the targeting by clicking the "Targeting" TAB. Edit the form to have Targeting enabled. Note: at this point the users does not exist as they need to be created by your code. As soon as you run the code the users will show and you can target individual users as below. Save Changes.

The screenshot shows the 'Feature flags' configuration page for 'NA_Flag1'. The 'Targeting' tab is selected, and the 'Targeting' toggle is turned 'On'. The 'Prerequisites' section is empty. The 'Target individual users' section is active, showing two groups: 'true' and 'false'. The 'true' group has one user, 'Bob Seager', and the 'false' group has one user, 'Bob Loblaw'. The 'Target users who match these rules' section is empty. The 'Default rule' section shows 'SERVE' set to 'true'. The 'If targeting is off, serve' section shows 'false'.

Feature flags / NA_Flag1

NA_Flag1 Added 2 minutes ago

NA_Flag1

Targeting Insights Experimentation Variations History Code references Settings

Targeting

On Evaluated 3 minutes ago

Prerequisites

Add prerequisites

Target individual users

true

Bob Seager

Add users...

false

Bob Loblaw

Add users...

Target users who match these rules

Add rules

Default rule

SERVE true



If targeting is off, serve false

- 9- The users are created by the code and should look like this when you run the code for the first time

Users

Use this page to find users in this environment. Users appear here when they have been created by a feature flag.

Filter Attributes (2)


	Name	Key	Email
	Bob Seager	Bob@seager.com	-
	Bob Loblaw	bob@example.com	-

- 10-Now you need to go to the Code and run it. The File accepts URL variables to reflect the platform parameters you created above. Start as follows;
- Copy the code to your local drive anywhere.
 - Apply "Change #1". Replace the URL variable named "PK" with the production (Client-Side) project key you created from the platform.

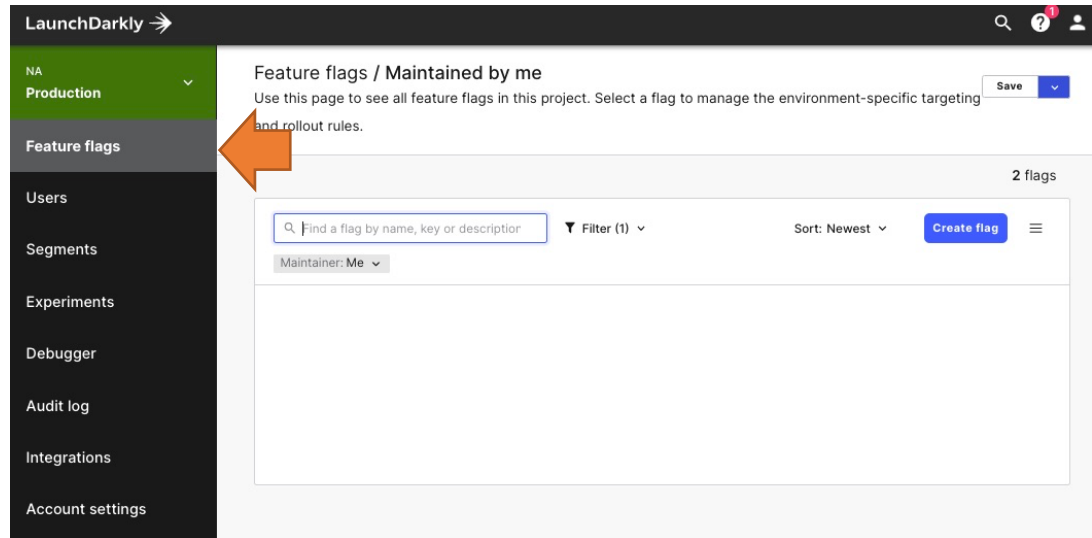
NA2 na-2 2

Edit Create environment

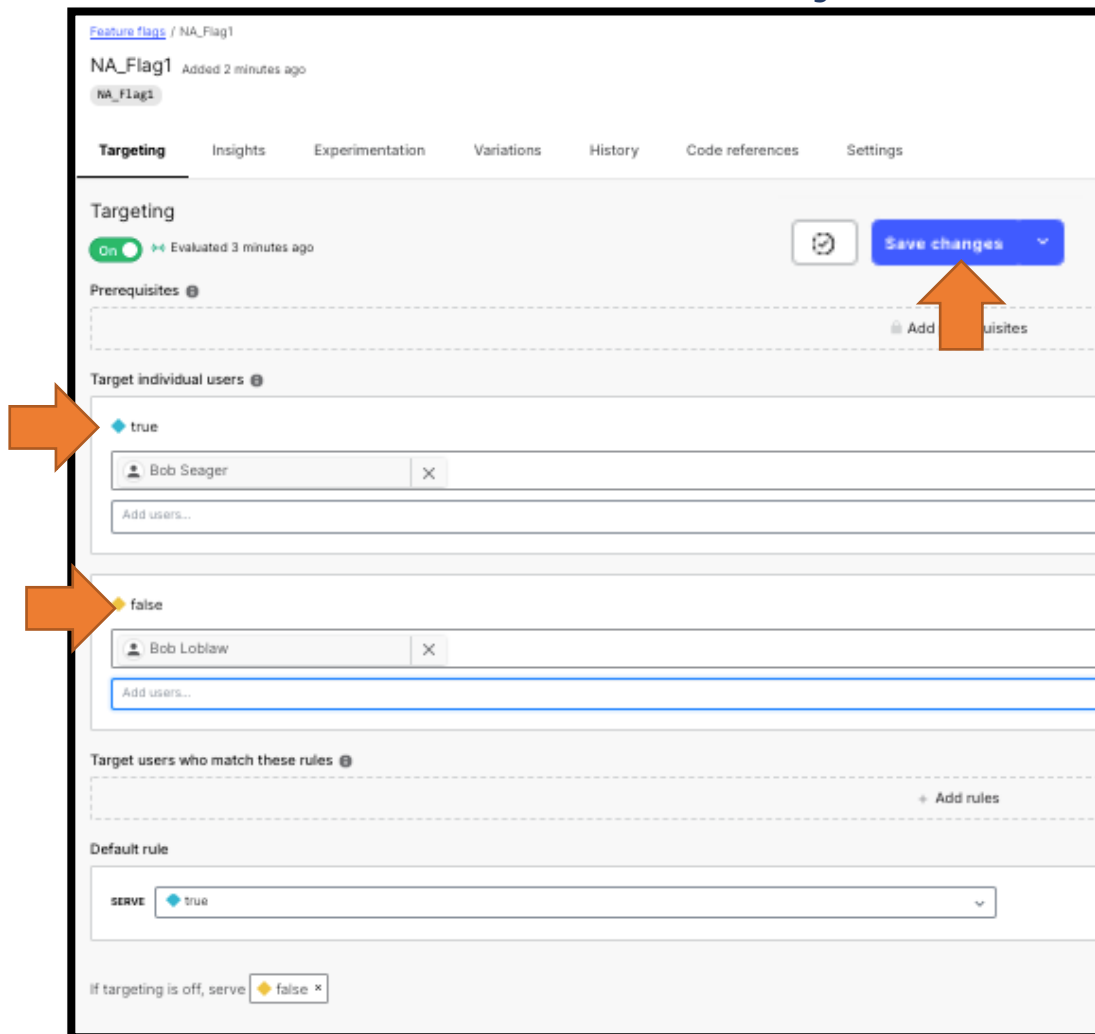
<div><div>Test</div><div>test</div></div>	0 minute TTL	SDK key	...
		Mobile key	
		Client-side ID	
<div><div>Production</div><div>production</div></div>	0 minute TTL	SDK key	...
		Mobile key	
		Client-side ID	



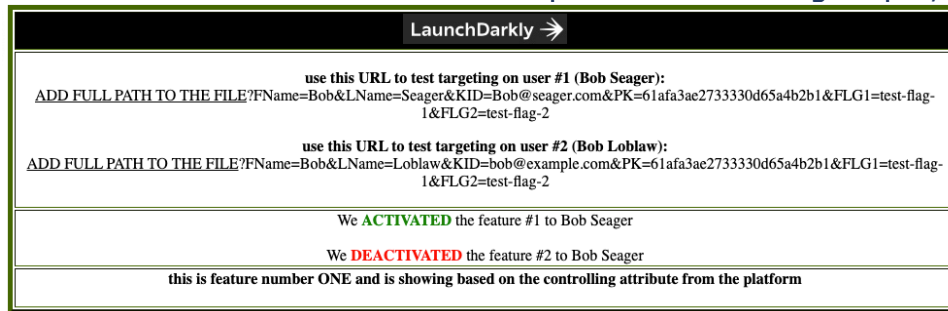
- c. Apply "Change #2". Return to the platform "Feature Flag" screen, and copy the IDs (in Gray) of the 2 flags you created earlier and replace the URL Variables named "FLG1" and "FLG2" with them.



- d. After you run successfully you should see the 2 new users created on the Launch Darkly "USERS" screen.
- e. Now start experimenting with enabling and disabling flags to see the changes
- f. Also start experimenting with the user targeting to see changes differ from one user to the other on the same flag



g. The Browser should show a similar output as the following output;



Example 1: <**FILE FULL**

PATH>?FName=Bob&LName=Seager&KID=Bob@seager.com&PK=61afa3ae2733330d65a4b2b1&FLG1=test-flag-1&FLG2=test-flag-2

Example 2: <**FILE FULL**

PATH>?FName=Bob&LName=Loblaw&KID=bob@example.com&PK=61afa3ae2733330d65a4b2b1&FLG1=test-flag-1&FLG2=test-flag-2