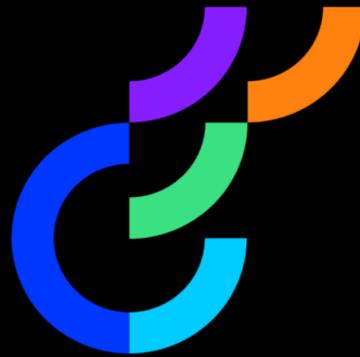


+

A simple white plus sign centered between the triangle and the logo.

# Implementing Optimizely flags through Vercel

Dom Sipowicz  
Solutions Architect  
Vercel



## Agenda

- Intro
- Demo 10 min
- Through the Vercel Toolbar
- Through the Vercel Platform
- Implementing feature flags as code
- Q&A

## DX

Feature flags enable developers to control the rollout of features with precision, allowing for safer deployments and testing in production environments.

## CSR and SSR

These flags can be implemented on either the client-side or server-side, each with its distinct implications.

# Optimize Feature Flags with Vercel Toolbar Use Cases

## Dev teams

Accelerate your team's workflow by using Vercel Toolbar to view, **override**, and **share your application's feature flags**.

Feature flags as code.

## QA / UAT

Test in any environment.

Deploy to Prod under feature flag.

Sign off UAT on QA/Preview/Prod.

## Debugging

Easily reproduce any flag configuration in any environment.

Observe how your flags are used through **Runtime Logs** and **Web Analytics**.



“Working with our feature flags in Vercel boosts our ability to quickly and safely iterate on experiments and features in development. The best part? It just works. It feels so magical during our development process.”

**Tanner Godarzi** Frontend Engineer at Notion

# Example

The screenshot shows a web browser window with the address bar displaying 'vercel.com'. The page content includes a navigation bar with 'Acme Store' on the left and 'GitHub' and 'Deploy' buttons on the right. The main content area features a product image of a black hoodie, a title 'Acme Hoodie', a descriptive paragraph, a price of '\$19.99', and selection options for 'Color' (Black) and 'Size' (XS, S, M, L, XL). At the bottom, there are buttons for 'Add to Cart', 'Buy Now', and 'View Cart'.

Acme Store

GitHub Deploy



## Acme Hoodie

Crafted for the modern trendsetter, this hoodie blends ultra-soft, breathable fabric with a sleek, minimalist design, perfect for both a relaxed day at home or a stylish outing. Featuring adjustable drawstrings and a spacious kangaroo pocket, it offers both comfort and convenience. The durable material ensures it can withstand the rigors of daily wear while maintaining its form and color after multiple washes.

**\$19.99**

Color

Black

Size

XS  S  M  L  XL

Add to Cart Buy Now

View Cart

## NEXT.js

- Client-side & Server-side Rendering
- Streaming
- Middleware
- Frontend Optimization

## ▲ Vercel

- Managed Infrastructure
- DX Platform
- Global Scalability & Security
- Built-in Observability

## Optimizely

- Experimentation
- Personalization
- Feature Flagging
- Measurement & Analysis

## Block Render



Harms performance and user experience leading to increased churn.

## Layout Shift



Harms SEO and user confidence. Risks incorrect exposure.

## Coupled Deploy & Release



Reduces iteration velocity and scalability.

## Client-side JavaScript & Snippets



Harms performance, increases page bloat, and requires CSP adjustment.

# Flag on PDP



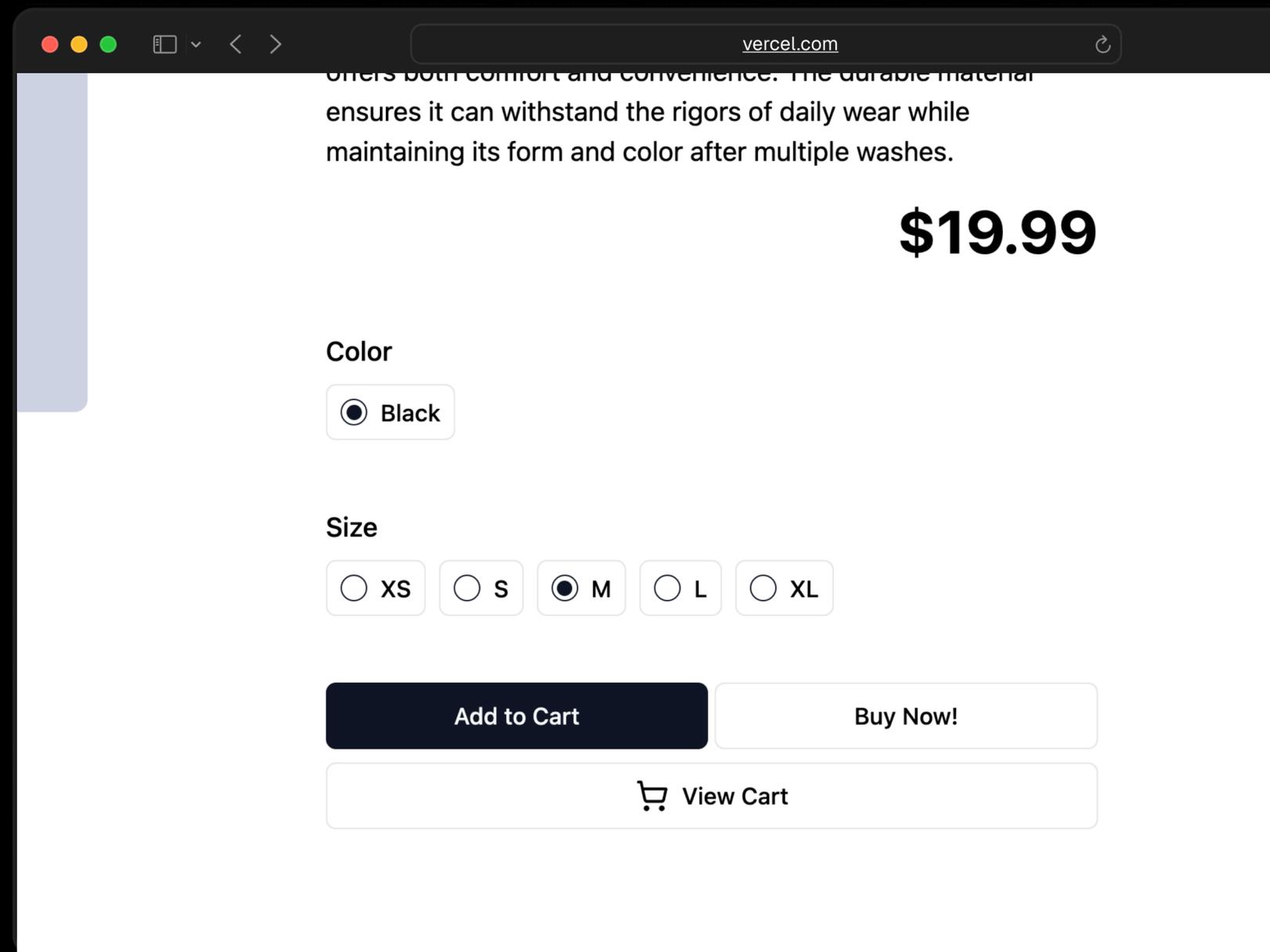
## Flag management

- “buynow” flag
- “buynow\_text” variable



## Implement Buy Now Button

- RSC with Suspense
- SSR
- Connect to Optimizely



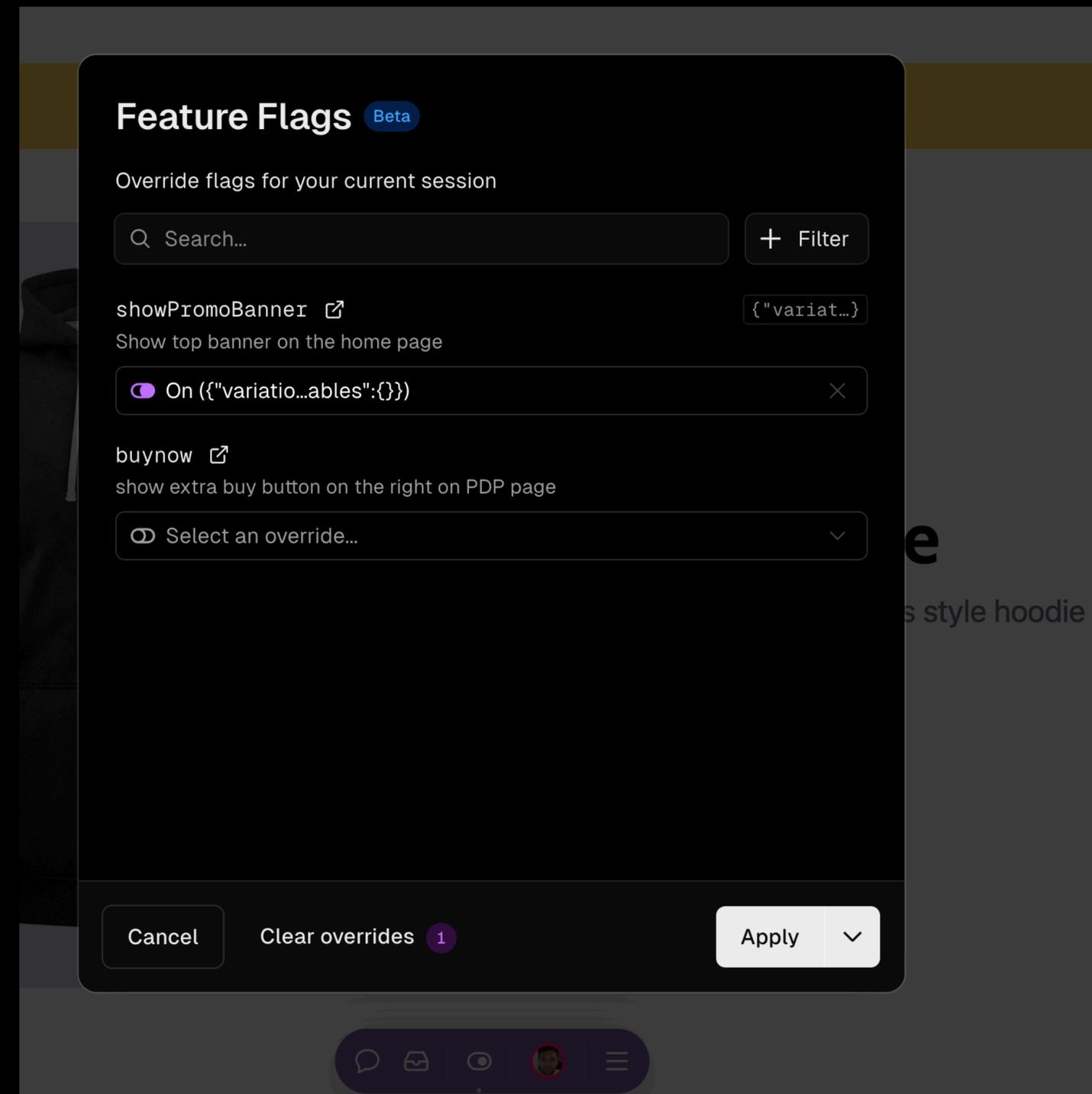
DEMO

# Through the Vercel Toolbar

You can recommend overrides to team members by either

Setting overrides as recommended for a given branch

Explicitly sharing a set of overrides through a URL with a team member





Apply and copy link

Copy the URL of the current page with the active overrides.



Apply and save as recommendation

Recommend the current overrides for branch main.

Clear overrides 1

Apply



Next.js (/app)

Next.js (/pages)

ts app/.well-known/vercel/flags/route.ts

TypeScript ▾



```
1 import { verifyAccess, type ApiData } from '@vercel/flags';
2 import { getOptimizelyData } from '@vercel/flags/providers/optimizely';
3 import { NextResponse, type NextRequest } from 'next/server';
4
5 export async function GET(request: NextRequest) {
6   const access = await verifyAccess(request.headers.get('Authorization'));
7   if (!access) return NextResponse.json(null, { status: 401 });
8
9   const optimizelyData = await getOptimizelyData({
10     projectId: process.env.OPTIMIZELY_PROJECT_ID,
11     apiKey: process.env.OPTIMIZELY_API_KEY,
12   });
13
14   return NextResponse.json<ApiData>(optimizelyData);
15 }
```

# Optimize Feature Flags Through the Vercel Platform

## Runtime Logs

See your feature flag's values in Runtime Logs

## Web Analytics

Break down your pageviews and custom events by feature flags in Web Analytics

## Web Vitals (CWV)

Soon, you'll also be able to change a flag in production and instantly see how it affects your Web Vitals in real-time in your Speed Insights.

requestPath:/product/hoodie method:GET typ... × ⋮

Time	Status	Host	Request	Message
JUL 11 11:52:51.91	GET 200	opti-...	 /product/hoodie	
JUL 11 11:52:48.73	GET 200	opti-...	 /product/hoodie	
JUL 11 11:52:35.11	GET 200	opti-...	 /product/hoodie	

No more logs to show within selected timeline

GET /product/hoodie 200 ^ ∨ ×

Request User Agent Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_15\_7) AppleWebKit/537.36 (KHTML, ...

Level Info

Environment production

Branch main

Cache MISS

Type  Serverless Function

Function /product/[slug]

Location Washington, D.C., USA (iad1)

Runtime  Node.js 20.x

Execution Duration / Limit 301ms / 15s

Memory Used / Limit 99 MB / 1769 MB

Request Metrics Early Access

GET cdn.optimize.ly.com/datafiles/DWkPr6Tg7GcZP2tQtzqpM.json 84ms ▬

Flags Beta

buynow {"ENABLED":TRUE,"VARIATIONKEY":"ON","VARIABLES":{"BUYNOW\_TEXT":"15% OFF"}}

[Open Deployment](#) ↗

[Open in Monitoring](#) ↗

# Web Analytics

opti-vercel-toolbar-flags.vercel.app 0 online

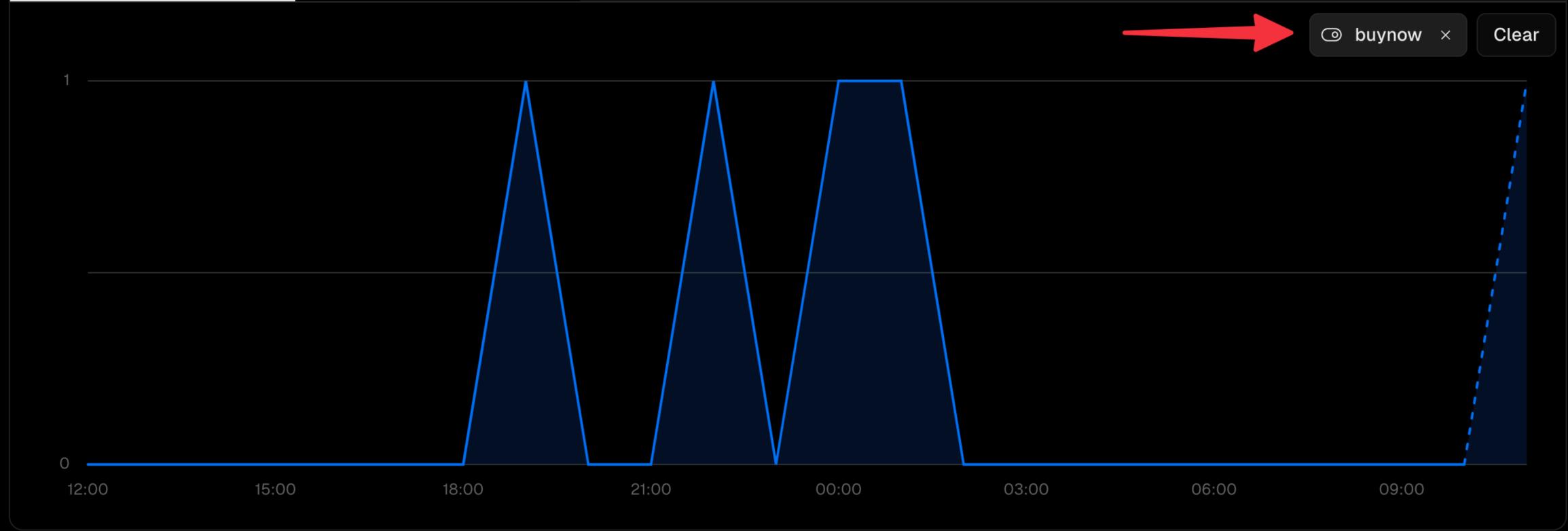
Production Last 24 Hours

Visitors

2

Page Views

54



Pages	VISITORS
/	2
/product/hoodie	2
/product/hat	1

Referrers	VISITORS
vercel.com	1



Project

Deployments

Code Checks

Analytics

Speed Insights

Logs

Sonar

Firewall

Storage

Settings



United Kingdom

2

View All ↗



Mac

2

View All ↗



Chrome

2

View All ↗



Events *i*

VISITORS

TOTAL



No data for active filter

Flags *i*

Beta

buynow ×

VISITORS

TOTAL

off

Click to filter

2

4

on

2

18

View All ↗



Next.js (/app) Next.js (/pages)

TS app/api/test/route.ts

TypeScript ▾ 

```
1 import { reportValue } from '@vercel/flags';
2
3 // force dynamic mode so the flag actually gets reported,
4 // otherwise the route would be static
5 export const dynamic = 'force-dynamic';
6
7 export async function GET() {
8   reportValue('summer-sale', false);
9   return Response.json({ ok: true });
10 }
```

 If you are using an implementation of the [Feature Flags pattern](#) you don't need to call `reportValue`. The respective implementation will automatically call `reportValue` for you.

# Flags Architectural Patterns

## Challenges with client-side loading

When feature flags are loaded on the client-side, they can introduce visual disruptions such as Layout Shift (**CLS**)

## Advantages of server-side loading

Server-side loading of feature flags resolves these issues by ensuring that the page is fully prepared with the correct configurations before reaching the user.

## Feature flags pattern

**Flags Pattern**  
– SSR, dynamic

**Precomputed Flags Pattern**  
– middleware + static/ISR

# Vercel Edge Config or Data Cache

TS flags.ts



```
1 import { unstable_flag as flag } from '@vercel/flags/next';
2 import { get } from '@vercel/edge-config';
3
4 export const showSummerSale = flag({
5   key: 'summer-sale',
6   async decide() {
7     const value = await get(this.key); // this.key refers to "summer-sale"
8     return value ?? false;
9   },
10 });
```

# Q&A

Dom Sipowicz  
Solutions Architect  
Vercel

@dom\_sipowicz

