Feature Flags

Drop Feature Flags

> Sources: `src/drop-app/src/lib/feature-flags.ts`, `src/drop-app/src/lib/features.ts`

Feature Flag System

Source: `feature-flags.ts`

Architecture

Feature flags are controlled via **environment variables** with the pattern:

```
NEXT_PUBLIC_FF_<SCREAMING_SNAKE_CASE>=true|false
```

The `NEXT_PUBLIC_` prefix ensures flags are available on both server and client (inlined at build time by Next.js).

Conversion example: `physicalCards` → `NEXT_PUBLIC_FF_PHYSICAL_CARDS`

Source: `feature-flags.ts:42-45`

Available Flags

Flag Name	Env Var	Default	Description
virtualCards	NEXT_PUBLIC_FF_VIRTUAL_CARDS	false	Virtual card issuance
physicalCards	NEXT_PUBLIC_FF_PHYSICAL_CARDS	false	Physical card ordering
cardDetails	NEXT_PUBLIC_FF_CARD_DETAILS	false	View full card details

Flag Name	Env Var	Default	Description
cardFreeze	NEXT_PUBLIC_FF_CARD_FREEZE	false	Card freeze/unfreeze
cardPin	NEXT_PUBLIC_FF_CARD_PIN	false	Card PIN management
spendingLimits	NEXT_PUBLIC_FF_SPENDING_LIMITS	false	Card spending limits
notifications	NEXT_PUBLIC_FF_NOTIFICATIONS	true	Push notifications
merchantDashboard	NEXT_PUBLIC_FF_MERCHANT_DASHBOARD	true	Merchant dashboard

Source: `feature-flags.ts:27-36`

Server-Side API

Function	Return Type	Description
`isEnabled(flag)`	`boolean`	Check if a flag is enabled
`getAllFlags()`	`FeatureFlags`	Get all flags with current values
`featureGate(flag)`	`NextResponse | null`	API middleware: returns 404 response if disabled, null if enabled

`featureGate` usage in routes:

```
// In any route handler:
const gate = featureGate("physicalCards");
if (gate) return gate;  // Returns 404 with "Feature not available"
```

Source: `feature-flags.ts:80-88`

Routes using `featureGate`:

Route	Flag
POST /api/cards/[id]/physical	`physicalCards`
POST /api/cards/[id]/pin	`cardPin`
GET /api/cards/[id]/limits	`spendingLimits`
PUT /api/cards/[id]/limits	`spendingLimits`
GET /api/notifications	`notifications`
PATCH /api/notifications	`notifications`

Client-Side API

Function	Return Type	Description
`useFeatureFlag(flag)`	`boolean`	React hook for a single flag
`useFeatureFlags()`	`FeatureFlags`	React hook for all flags

These work because `NEXT_PUBLIC_*` env vars are inlined at build time — no server roundtrip needed.

Source: `feature-flags.ts:94-114`

Feature Tracking System

Source: `features.ts`

A separate system for tracking **implementation progress** of Drop features. Not runtime flags — this is a development tracking tool.

Feature Interface

```
interface Feature {
  id: string;              // e.g., "auth-001"
  category: string;        // e.g., "Authentication"
  name: string;            // e.g., "User Registration"
  description: string;
  status: "pending" | "in_progress" | "passing" | "failing";
  priority: number;        // 1 = highest
  dependencies: string[];  // IDs of prerequisite features
  acceptanceCriteria: string[];
  implementedAt?: string;  // ISO date
  testedAt?: string;       // ISO date
}
```

Feature Categories and Status

Category	Total	Passing	Pending	Notes

Authentication	4	3	1 (Biometric Login)	
KYC	1	1	0	
Banking	6	5	1	bank-006 (Top-up via Card) is FUTURE — incompatible with pass-through model
Cards	4	4	0	FUTURE — all card features are gated behind feature flags (default: false)
Notifications	1	0	1 (Push Notifications)	

All Features

ID	Name	Status	Priority	Dependencies	Notes
auth-001	User Registration	passing	1	-	
auth-002	PIN Login	passing	1	auth-001	
auth-003	Logout	passing	2	auth-002	
auth-004	Biometric Login	pending	3	auth-002	
kyc-001	Identity Verification	passing	1	auth-001	
bank-001	IBAN Generation	passing	1	kyc-001	
bank-002	Balance Display	passing	1	bank-001	AISP read-only
bank-003	Send Money	passing	1	bank-002	PISP from user's bank
bank-004	Receive Money	passing	1	bank-001	
bank-005	Transaction History	passing	2	bank-003, bank-004	
bank-006	Top-up via Card	passing	2	bank-001	**FUTURE** — no wallet in pass-through model
card-001	Virtual Card Issuance	passing	1	kyc-001	**FUTURE** — feature-flagged
card-002	Card Freeze/Unfreeze	passing	2	card-001	**FUTURE** — feature-flagged
card-003	Card Transactions	passing	1	card-001	**FUTURE** — feature-flagged
card-004	Physical Card Order	passing	3	card-001	**FUTURE** — feature-flagged

ID	Name	Status	Priority	Dependencies	Notes
notif-001	Push Notifications	pending	3	auth-001	

Helper Functions

Function	Description
`getFeaturesByStatus(status)`	Filter features by status
`getFeaturesByCategory(category)`	Filter features by category
`getFeatureStats()`	Get counts: total, passing, pending, inProgress, failing, percentComplete
`getReadyFeatures()`	Features whose dependencies are all `passing`
`printFeatureReport()`	Formatted text report

Source: `features.ts:284-357`

Environment Variable Summary

Variable	Purpose	Default
NEXT_PUBLIC_FF_VIRTUAL_CARDS	Enable virtual cards	false
NEXT_PUBLIC_FF_PHYSICAL_CARDS	Enable physical cards	false
NEXT_PUBLIC_FF_CARD_DETAILS	Enable card detail view	false
NEXT_PUBLIC_FF_CARD_FREEZE	Enable card freeze	false
NEXT_PUBLIC_FF_CARD_PIN	Enable card PIN	false
NEXT_PUBLIC_FF_SPENDING_LIMITS	Enable spending limits	false
NEXT_PUBLIC_FF_NOTIFICATIONS	Enable notifications	true
NEXT_PUBLIC_FF_MERCHANT_DASHBOARD	Enable merchant dashboard	true
NEXT_PUBLIC_SERVICE_MODE	mock or production	mock
DATABASE_URL	PostgreSQL 16 connection string	Required (no SQLite fallback)
JWT_SECRET	JWT signing secret	dev-only fallback
NEXT_PUBLIC_APP_URL	App URL for CSRF	-
SEED_DEMO	Enable demo data in staging	-

Revision #5
Created 2026-02-18 08:44:22 UTC by John

Updated 2026-05-23 10:57:46 UTC by John