

API Specification

API Specification: Drop

Version: 1.0 **Date:** 2026-02-09 **Author:** dev agent (Ollama) + John (AI Director) **Base URL:** `/api`
(Next.js API Routes) **Auth:** JWT in httpOnly cookie (jose library) **Database:** PostgreSQL 16 via
Drizzle ORM (ADR-014; `better-sqlite3` removed 2026-03-03)

1. Overview

API Style: REST **Format:** JSON **Rate Limiting:** 60 req/min per IP (standard), 10 req/min for auth endpoints **Auth mechanism:** JWT token set as httpOnly, secure, sameSite=strict cookie

2. Authentication

POST /api/auth/register

Description: Register new user account

Request:

```
{
  "email": "amir@example.com",
  "password": "min8chars",
  "firstName": "Amir",
  "lastName": "Hadžić",
  "phone": "+4712345678"
}
```

Response 201:

```
{
  "data": {
    "id": "usr_abc123",
    "email": "amir@example.com",
    "firstName": "Amir",
    "lastName": "Hadžić",
    "kycStatus": "pending",
    "createdAt": "2026-02-09T10:00:00Z"
  }
}
```

Sets httpOnly JWT cookie

Errors: 400 (validation), 409 (email exists)

POST /api/auth/login

Description: Login and receive JWT cookie

Request:

```
{
  "email": "amir@example.com",
  "password": "min8chars"
}
```

Response 200:

```
{
  "data": {
    "id": "usr_abc123",
    "email": "amir@example.com",
    "firstName": "Amir",
    "lastName": "Hadžić",
    "kycStatus": "approved"
  }
}
```

Sets httpOnly JWT cookie (24h expiry). Note: balance is NOT returned here — use /api/bank-accounts (AISP) to read bank balance.

Errors: 401 (wrong credentials), 423 (account locked)

POST /api/auth/logout

Description: Clear JWT cookie **Auth:** Required

Response 200:

```
{ "message": "Logged out" }
```

Clears httpOnly cookie

GET /api/auth/me

Description: Get current user from JWT **Auth:** Required

Response 200:

```
{
  "data": {
    "id": "usr_abc123",
    "email": "amir@example.com",
    "firstName": "Amir",
    "lastName": "Hadžić",
    "kycStatus": "approved",
    "createdAt": "2026-02-09T10:00:00Z"
  }
}
```

3. Bank Accounts (AISP — Pass-through)

“ **Pass-through model:** Drop never holds customer money. Balance is read from user's real bank account via Open Banking (AISP). Payments are initiated via

PISP from user's bank.

GET /api/bank-accounts

Description: Get linked bank accounts and balances via AISP (Open Banking) **Auth:** Required (BankID consent)

Response 200:

```
{
  "data": [
    {
      "id": "ba_1",
      "bankName": "SpareBank 1",
      "accountNumber": "*****1234",
      "balance": 12450.00,
      "currency": "NOK",
      "lastSynced": "2026-02-09T10:00:00Z"
    }
  ]
}
```

Note: Balance is a cached AISP read from the user's actual bank account. Drop does not store or manage this balance.

~~GET /api/users/balance — REMOVED~~

~~POST /api/users/top-up — REMOVED~~

“ These endpoints were part of the old wallet model and have been removed. In the pass-through model, there is no wallet to check or top up. Use `/api/bank-accounts` to read bank balances via AISP.

4. Recipients

GET /api/recipients

Description: List user's saved recipients **Auth:** Required **Query:** `?page=1&limit=20`

Response 200:

```
{
  "data": [
    {
      "id": "rec_1",
      "name": "Mama Jasmina",
      "country": "RS",
      "countryName": "Serbia",
      "currency": "RSD",
      "bankAccount": "*****1234",
      "createdAt": "2026-02-01T10:00:00Z"
    }
  ],
  "pagination": {
    "page": 1,
    "limit": 20,
    "total": 3
  }
}
```

POST /api/recipients

Description: Add new recipient **Auth:** Required

Request:

```
{
  "name": "Mama Jasmina",
  "country": "RS",
  "currency": "RSD",
  "bankAccount": "265100000012345678",
  "bankName": "Banca Intesa"
}
```

Response 201:

```
{
  "data": {
    "id": "rec_new",
    "name": "Mama Jasmina",
    "country": "RS",
    "currency": "RSD",
    "bankAccount": "*****5678",
    "createdAt": "2026-02-09T10:00:00Z"
  }
}
```

Errors: 400 (validation), 422 (unsupported country)

DELETE /api/recipients/:id

Description: Remove recipient **Auth:** Required **Response 204:** No content

5. Exchange Rates

GET /api/rates

Description: Get current exchange rates for all corridors **Auth:** Not required

Response 200:

```
{
  "data": {
    "baseCurrency": "NOK",
    "rates": {
      "RSD": 11.70,
      "BAM": 1.04,
      "PLN": 0.41,
      "PKR": 26.80,
      "TRY": 3.45,
      "EUR": 0.089
    },
    "updatedAt": "2026-02-09T10:00:00Z"
  }
}
```

```
}  
}
```

GET /api/rates/:currency

Description: Get rate for specific currency pair **Auth:** Not required

Response 200:

```
{  
  "data": {  
    "from": "NOK",  
    "to": "RSD",  
    "rate": 11.70,  
    "fee": 0.005,  
    "updatedAt": "2026-02-09T10:00:00Z"  
  }  
}
```

Errors: 404 (unsupported currency)

6. Transactions — Remittance

POST /api/transactions/remittance

Description: Create new remittance transfer **Auth:** Required (KYC must be approved)

Request:

```
{  
  "recipientId": "rec_1",  
  "amount": 2000.00,  
  "currency": "NOK"  
}
```

Response 201:

```
{
  "data": {
    "id": "tx_rem_123",
    "type": "remittance",
    "status": "processing",
    "sendAmount": 2000.00,
    "sendCurrency": "NOK",
    "receiveAmount": 23400.00,
    "receiveCurrency": "RSD",
    "exchangeRate": 11.70,
    "fee": 10.00,
    "feePercent": 0.5,
    "total": 2010.00,
    "recipientName": "Mama Jasmina",
    "recipientCountry": "RS",
    "eta": "1-2 business days",
    "createdAt": "2026-02-09T10:00:00Z"
  }
}
```

Errors:

- 400 — invalid amount (min 100 NOK, max 50000 NOK)
- 402 — insufficient balance
- 403 — KYC not approved
- 404 — recipient not found
- 422 — unsupported corridor

7. Transactions — QR Payment

POST /api/transactions/qr-payment

Description: Pay a merchant via QR code **Auth:** Required

Request:

```
{
  "merchantId": "mer_1",
  "amount": 129.00
}
```



```
}
```

Response 201:

```
{
  "data": {
    "id": "tx_qr_456",
    "type": "qr_payment",
    "status": "completed",
    "amount": 129.00,
    "currency": "NOK",
    "fee": 1.29,
    "feePercent": 1.0,
    "merchantName": "Ahmetov Kebab",
    "merchantId": "mer_1",
    "createdAt": "2026-02-09T14:23:00Z"
  }
}
```

Errors:

- 400 — invalid amount (min 1 NOK)
- 402 — insufficient balance
- 404 — merchant not found

8. Transactions — List

GET /api/transactions

Description: List user's transactions (both remittance and QR) **Auth:** Required **Query:**

?page=1&limit=20&type=remittance|qr_payment&status=completed|processing|failed

Response 200:

```
{
  "data": [
    {
      "id": "tx_rem_123",
      "type": "remittance",

```

```
    "status": "completed",
    "amount": -2000.00,
    "currency": "NOK",
    "recipientName": "Mama Jasmina",
    "createdAt": "2026-02-09T10:00:00Z"
  },
  {
    "id": "tx_qr_456",
    "type": "qr_payment",
    "status": "completed",
    "amount": -129.00,
    "currency": "NOK",
    "merchantName": "Ahmetov Kebab",
    "createdAt": "2026-02-09T14:23:00Z"
  }
],
"pagination": {
  "page": 1,
  "limit": 20,
  "total": 47
}
}
```

GET /api/transactions/:id

Description: Get transaction detail **Auth:** Required

Response 200:

```
{
  "data": {
    "id": "tx_rem_123",
    "type": "remittance",
    "status": "completed",
    "sendAmount": 2000.00,
    "sendCurrency": "NOK",
    "receiveAmount": 23400.00,
    "receiveCurrency": "RSD",
    "exchangeRate": 11.70,
  }
}
```

```
"fee": 10.00,  
"total": 2010.00,  
"recipientName": "Mama Jasmina",  
"recipientCountry": "RS",  
"createdAt": "2026-02-09T10:00:00Z",  
"completedAt": "2026-02-10T14:30:00Z"  
}  
}
```

9. Merchants

POST /api/merchants/register

Description: Register as a merchant **Auth:** Required

Request:

```
{  
  "businessName": "Ahmetov Kebab",  
  "orgNumber": "923456789",  
  "address": "Grønland 12, Oslo",  
  "bankAccount": "1234.56.78901"  
}
```

Response 201:

```
{  
  "data": {  
    "id": "mer_1",  
    "businessName": "Ahmetov Kebab",  
    "orgNumber": "923456789",  
    "qrCode": "drop://pay/mer_1",  
    "status": "active",  
    "feeRate": 0.01,  
    "createdAt": "2026-02-09T10:00:00Z"  
  }  
}
```

Errors: 400 (validation), 409 (org number exists)

GET /api/merchants/dashboard

Description: Get merchant stats **Auth:** Required (merchant role) **Query:**

?period=today|week|month

Response 200:

```
{
  "data": {
    "period": "today",
    "revenue": 4350.00,
    "transactionCount": 12,
    "fees": 43.50,
    "netRevenue": 4306.50,
    "nextPayout": 4306.50,
    "payoutTime": "17:00"
  }
}
```

GET /api/merchants/transactions

Description: List merchant's received payments **Auth:** Required (merchant role) **Query:**

?page=1&limit=20&date=2026-02-09

Response 200:

```
{
  "data": [
    {
      "id": "tx_qr_456",
      "customerName": "Amir K.",
      "amount": 129.00,
      "fee": 1.29,
      "net": 127.71,
      "status": "completed",
      "time": "14:23"
    }
  ]
}
```

```
1,
"pagination": {
  "page": 1,
  "limit": 20,
  "total": 12
}
}
```

GET /api/merchants/qr

Description: Get merchant's QR code data **Auth:** Required (merchant role)

Response 200:

```
{
  "data": {
    "merchantId": "mer_1",
    "businessName": "Ahmetov Kebab",
    "qrValue": "drop://pay/mer_1",
    "address": "Grønland 12, Oslo"
  }
}
```

10. Database Schema (PostgreSQL 16 — 19 tables)

Source: `src/shared/db/schema.ts` (Drizzle ORM schema — PostgreSQL 16, all environments; `better-sqlite3` removed per ADR-014)

Core Tables (12)

“ **Note:** The SQL below is a historical snapshot from the original spec (SQLite syntax). The authoritative schema is `src/shared/db/schema.ts` (Drizzle ORM, PostgreSQL 16). Use `make db-push` to apply schema changes.

```

-- Users (NO balance field – pass-through model)
CREATE TABLE users (
  id TEXT PRIMARY KEY,
  email TEXT UNIQUE NOT NULL,
  password_hash TEXT NOT NULL,
  first_name TEXT NOT NULL,
  last_name TEXT NOT NULL,
  phone TEXT,
  date_of_birth TEXT,
  kyc_status TEXT DEFAULT 'pending' CHECK(kyc_status IN ('pending','approved','rejected')),
  role TEXT DEFAULT 'user' CHECK(role IN ('user','merchant')),
  risk_level TEXT DEFAULT 'low' CHECK(risk_level IN ('low','medium','high')),
  pep_status TEXT DEFAULT 'not_checked' CHECK(pep_status IN
('not_checked','clear','match','pending_review')),
  sanctions_cleared INTEGER DEFAULT 0,
  kyc_method TEXT CHECK(kyc_method IN ('bankid','document','simplified')),
  kyc_verified_at TEXT,
  national_id_hash TEXT,
  deleted_at TEXT,
  created_at TEXT DEFAULT (datetime('now'))
);

-- Bank accounts (balance is cached AISP read, NOT held by Drop)
CREATE TABLE bank_accounts (
  id TEXT PRIMARY KEY,
  user_id TEXT NOT NULL REFERENCES users(id),
  bank_name TEXT NOT NULL,
  account_number TEXT NOT NULL,
  iban TEXT,
  balance REAL DEFAULT 0, -- Cached AISP-read balance (read-only in production)
  balance_synced_at TEXT, -- When balance was last synced from bank via AISP
  currency TEXT DEFAULT 'NOK',
  is_primary INTEGER DEFAULT 0,
  connected_at TEXT DEFAULT (datetime('now'))
);

CREATE TABLE recipients (
  id TEXT PRIMARY KEY,
  user_id TEXT NOT NULL REFERENCES users(id),

```

```
name TEXT NOT NULL,  
country TEXT NOT NULL,  
currency TEXT NOT NULL,  
bank_account TEXT NOT NULL,  
bank_name TEXT,  
created_at TEXT DEFAULT (datetime('now'))  
);
```

```
CREATE TABLE merchants (  
  id TEXT PRIMARY KEY,  
  user_id TEXT NOT NULL REFERENCES users(id),  
  business_name TEXT NOT NULL,  
  org_number TEXT UNIQUE NOT NULL,  
  address TEXT,  
  bank_account TEXT NOT NULL,  
  fee_rate REAL DEFAULT 0.01,  
  status TEXT DEFAULT 'active',  
  created_at TEXT DEFAULT (datetime('now'))  
);
```

```
CREATE TABLE transactions (  
  id TEXT PRIMARY KEY,  
  user_id TEXT NOT NULL REFERENCES users(id),  
  type TEXT NOT NULL CHECK(type IN ('remittance','qr_payment')),  
  status TEXT DEFAULT 'processing' CHECK(status IN ('processing','completed','failed')),  
  amount REAL NOT NULL,  
  currency TEXT DEFAULT 'NOK',  
  fee REAL DEFAULT 0,  
  recipient_id TEXT REFERENCES recipients(id),  
  merchant_id TEXT REFERENCES merchants(id),  
  send_amount REAL,  
  send_currency TEXT,  
  receive_amount REAL,  
  receive_currency TEXT,  
  exchange_rate REAL,  
  purpose_code TEXT,  
  created_at TEXT DEFAULT (datetime('now')),  
  completed_at TEXT  
);
```

```
CREATE TABLE exchange_rates (  
  id INTEGER PRIMARY KEY AUTOINCREMENT, -- SERIAL for PostgreSQL  
  from_currency TEXT DEFAULT 'NOK',  
  to_currency TEXT NOT NULL,  
  rate REAL NOT NULL,  
  updated_at TEXT DEFAULT (datetime('now'))  
);
```

```
CREATE TABLE cards (  
  id TEXT PRIMARY KEY,  
  user_id TEXT NOT NULL REFERENCES users(id),  
  type TEXT DEFAULT 'virtual' CHECK(type IN ('virtual','physical')),  
  last_four TEXT NOT NULL,  
  token_ref TEXT,  
  expiry TEXT NOT NULL,  
  status TEXT DEFAULT 'active' CHECK(status IN ('active','frozen','cancelled')),  
  shipping_address TEXT,  
  pin_hash TEXT,  
  created_at TEXT DEFAULT (datetime('now'))  
);
```

```
CREATE TABLE sessions (  
  id TEXT PRIMARY KEY,  
  user_id TEXT NOT NULL REFERENCES users(id),  
  token_hash TEXT NOT NULL,  
  created_at TEXT DEFAULT (datetime('now')),  
  expires_at TEXT NOT NULL,  
  revoked INTEGER DEFAULT 0  
);
```

```
CREATE TABLE notifications (  
  id TEXT PRIMARY KEY,  
  user_id TEXT NOT NULL REFERENCES users(id),  
  type TEXT NOT NULL,  
  title TEXT NOT NULL,  
  body TEXT NOT NULL,  
  read INTEGER DEFAULT 0,  
  created_at TEXT DEFAULT (datetime('now'))
```



```

);

CREATE TABLE settings (
  user_id TEXT PRIMARY KEY REFERENCES users(id),
  currency TEXT DEFAULT 'NOK',
  language TEXT DEFAULT 'nb',
  push_enabled INTEGER DEFAULT 1,
  email_enabled INTEGER DEFAULT 1,
  updated_at TEXT DEFAULT (datetime('now'))
);

CREATE TABLE spending_limits (
  id TEXT PRIMARY KEY,
  user_id TEXT NOT NULL REFERENCES users(id),
  card_id TEXT REFERENCES cards(id),
  limit_type TEXT NOT NULL,
  amount REAL NOT NULL,
  currency TEXT DEFAULT 'NOK',
  created_at TEXT DEFAULT (datetime('now'))
);

CREATE TABLE rate_limits (
  key TEXT PRIMARY KEY,
  count INTEGER NOT NULL,
  reset_at INTEGER NOT NULL
);

```

Compliance Tables (7) — Added 2026-02-16

```

CREATE TABLE audit_log (
  id TEXT PRIMARY KEY,
  timestamp TEXT DEFAULT (datetime('now')),
  user_id TEXT REFERENCES users(id),
  action TEXT NOT NULL,
  resource_type TEXT,
  resource_id TEXT,
  details TEXT,
  ip_address TEXT,
  user_agent TEXT
);

```

```

);

CREATE TABLE aml_alerts (
    id TEXT PRIMARY KEY,
    user_id TEXT NOT NULL REFERENCES users(id),
    alert_type TEXT NOT NULL,
    severity TEXT NOT NULL CHECK(severity IN ('low','medium','high','critical')),
    transaction_id TEXT REFERENCES transactions(id),
    details TEXT,
    status TEXT DEFAULT 'open' CHECK(status IN
('open','investigating','resolved','escalated','filed')),
    reviewed_by TEXT,
    reviewed_at TEXT,
    created_at TEXT DEFAULT (datetime('now'))
);

CREATE TABLE str_reports (
    id TEXT PRIMARY KEY,
    user_id TEXT NOT NULL REFERENCES users(id),
    alert_id TEXT REFERENCES aml_alerts(id),
    report_type TEXT NOT NULL,
    status TEXT DEFAULT 'draft' CHECK(status IN ('draft','submitted','acknowledged')),
    filed_at TEXT,
    reference_number TEXT,
    details TEXT,
    created_at TEXT DEFAULT (datetime('now'))
);

CREATE TABLE screening_results (
    id TEXT PRIMARY KEY,
    user_id TEXT NOT NULL REFERENCES users(id),
    screening_type TEXT NOT NULL CHECK(screening_type IN ('pep','sanctions','adverse_media')),
    provider TEXT,
    result TEXT NOT NULL CHECK(result IN ('clear','match','potential_match','error')),
    match_details TEXT,
    screened_at TEXT DEFAULT (datetime('now'))
);

CREATE TABLE consents (

```

```

    id TEXT PRIMARY KEY,
    user_id TEXT NOT NULL REFERENCES users(id),
    consent_type TEXT NOT NULL,
    granted INTEGER NOT NULL DEFAULT 1,
    granted_at TEXT DEFAULT (datetime('now')),
    withdrawn_at TEXT,
    ip_address TEXT
);

CREATE TABLE data_access_requests (
    id TEXT PRIMARY KEY,
    user_id TEXT NOT NULL REFERENCES users(id),
    request_type TEXT NOT NULL CHECK(request_type IN
('export','erasure','rectification','restriction')),
    status TEXT DEFAULT 'pending' CHECK(status IN
('pending','processing','completed','rejected')),
    requested_at TEXT DEFAULT (datetime('now')),
    completed_at TEXT,
    download_url TEXT,
    notes TEXT
);

CREATE TABLE complaints (
    id TEXT PRIMARY KEY,
    user_id TEXT NOT NULL REFERENCES users(id),
    category TEXT NOT NULL,
    subject TEXT NOT NULL,
    description TEXT NOT NULL,
    status TEXT DEFAULT 'received' CHECK(status IN
('received','investigating','resolved','escalated')),
    resolution TEXT,
    created_at TEXT DEFAULT (datetime('now')),
    resolved_at TEXT
);

```

Indexes

```

-- Core indexes
CREATE INDEX idx_transactions_user ON transactions(user_id);

```

```

CREATE INDEX idx_transactions_merchant ON transactions(merchant_id);
CREATE INDEX idx_recipients_user ON recipients(user_id);
CREATE INDEX idx_merchants_org ON merchants(org_number);
CREATE INDEX idx_bank_accounts_user ON bank_accounts(user_id);
CREATE INDEX idx_cards_user ON cards(user_id);
CREATE INDEX idx_sessions_user ON sessions(user_id);
CREATE INDEX idx_sessions_token ON sessions(token_hash);
CREATE INDEX idx_notifications_user ON notifications(user_id);
CREATE INDEX idx_spending_limits_user ON spending_limits(user_id);
CREATE INDEX idx_spending_limits_card ON spending_limits(card_id);

-- Compliance indexes
CREATE INDEX idx_audit_log_user ON audit_log(user_id);
CREATE INDEX idx_audit_log_timestamp ON audit_log(timestamp);
CREATE INDEX idx_audit_log_action ON audit_log(action);
CREATE INDEX idx_aml_alerts_user ON aml_alerts(user_id);
CREATE INDEX idx_aml_alerts_status ON aml_alerts(status);
CREATE INDEX idx_screening_user ON screening_results(user_id);
CREATE INDEX idx_consent_user ON consents(user_id);
CREATE INDEX idx_data_requests_user ON data_access_requests(user_id);
CREATE INDEX idx_complaints_user ON complaints(user_id);
CREATE INDEX idx_complaints_status ON complaints(status);

```

Note: Schema uses PostgreSQL 16 in all environments (development, CI, staging, production) via Drizzle ORM. The old dual-mode SQLite/PostgreSQL driver has been removed (ADR-014).

11. Common Error Format

All errors follow this format:

```

{
  "error": "error_code",
  "message": "Human readable message",
  "details": []
}

```

Code	Error	Description
400	bad_request	Malformed request body

Code	Error	Description
401	unauthorized	Missing or expired JWT
402	insufficient_balance	Not enough NOK in bank account (PISP will fail)
403	kyc_required	KYC must be approved
404	not_found	Resource not found
409	conflict	Duplicate (email, org number)
422	validation_error	Field validation failed
429	rate_limited	Too many requests
500	internal_error	Server error

12. Rate Limits

Endpoint Group	Limit	Window
Auth (login/register)	10/min	Per IP
Transactions (create)	30/min	Per user
Read endpoints	60/min	Per user
Exchange rates	120/min	Per IP

13. API Route File Structure (Next.js)

```
src/app/api/  
├─ auth/  
│   ├── register/route.ts  
│   ├── login/route.ts  
│   ├── logout/route.ts  
│   └─ me/route.ts  
├─ bank-accounts/  
│   └─ route.ts           (GET linked accounts via AISP)  
├─ recipients/  
│   ├── route.ts         (GET list, POST create)  
│   └─ [id]/route.ts     (DELETE)  
└─ transactions/
```

```
| └─ route.ts          (GET list)
| └─ [id]/route.ts     (GET detail)
| └─ remittance/route.ts (POST)
|   └─ qr-payment/route.ts (POST)
└─ merchants/
  | └─ register/route.ts
  | └─ dashboard/route.ts
  | └─ transactions/route.ts
  |   └─ qr/route.ts
└─ rates/
  | └─ route.ts          (GET all)
  |   └─ [currency]/route.ts (GET specific)
└─ lib/
  | └─ db.ts             (PostgreSQL/Drizzle connection)
  | └─ auth.ts           (JWT verify/sign)
  |   └─ middleware.ts   (auth middleware, rate limit)
```

Generated: 2026-02-09 by dev agent (Ollama) + John (orchestration) Status: Ready for implementation (Sprint 1)

Revision #7

Created 2026-02-18 08:44:18 UTC by John

Updated 2026-06-27 21:54:03 UTC by John