CrymadX Mobile App - Backend API Documentation

For Backend Developer - Complete Endpoints & Integration Guide

Document Version: 1.0 **App Name:** CrymadX **Platform:** Flutter (iOS & Android) **Date Generated:** January 15, 2026

Table of Contents

1. Overview
2. API Configuration
3. Authentication Endpoints
4. User Management Endpoints
5. Wallet & Transaction Endpoints
6. Trading Endpoints
7. P2P Trading Endpoints
8. Earn & Staking Endpoints
9. NFT Marketplace Endpoints
10. Fiat On-Ramp Endpoints
11. All Screens & Their Requirements
12. All Buttons & Actions
13. Form Fields & Validation
14. Data Models
15. WebSocket Events
16. Error Codes
17. Security Requirements

1. Overview

CrymadX is a comprehensive cryptocurrency trading platform with the following features:

- User authentication with 2FA support
- Wallet management (deposits, withdrawals, transfers)
- Spot trading with order book
- P2P trading marketplace
- Staking and earning features
- NFT marketplace
- Fiat on-ramp integration
- KYC verification system
- Referral and rewards program

Tech Stack Requirements

- RESTful API with JSON responses
- JWT authentication with refresh tokens
- WebSocket for real-time data (prices, order book)
- File upload support for KYC documents
- Rate limiting and request throttling

2. API Configuration

Base URLs

```
Development: http://localhost:8000/api/v1
Production:  https://api.crymadx.com/api/v1
```

Request Headers

```
Content-Type: application/json
Authorization: Bearer {accessToken}
X-Device-Id: {deviceId}
X-App-Version: {appVersion}
```

Response Format

```json
{
  "success": true,
  "data": { ... },
  "message": "Success message",
  "timestamp": "2026-01-15T10:30:00Z"
}
```

Error Response Format

```json
{
  "success": false,
  "error": {
    "code": "ERROR_CODE",
    "message": "Human readable message",
    "details": { ... }
  },
  "timestamp": "2026-01-15T10:30:00Z"
}
```

Pagination Format

```json
{
  "data": [...],
  "pagination": {
    "page": 1,
    "limit": 20,
    "total": 100,
    "totalPages": 5,
    "hasNext": true,
    "hasPrevious": false
  }
}
```

---

3. Authentication Endpoints

3.1 User Registration

```
POST /auth/register
```

Request Body:

```
{
  "email": "user@example.com",
  "password": "SecurePass123!",
  "name": "John Doe",
  "referralCode": "CRYMADX2024"  // Optional
}
```

Response:

```
{
  "success": true,
  "data": {
    "accessToken": "eyJhbGciOiJIUzI1NiIs...",
    "refreshToken": "eyJhbGciOiJIUzI1NiIs...",
    "expiresIn": 3600,
    "user": {
      "id": "usr_123456",
      "email": "user@example.com",
      "name": "John Doe",
      "emailVerified": false,
      "kycLevel": 0,
      "referralCode": "USER123ABC"
    }
  }
}
```

Validation Rules:

- Email: Valid email format, unique
- Password: Min 8 chars, 1 uppercase, 1 lowercase, 1 number, 1 special char
- Name: Min 2 chars, max 100 chars

3.2 User Login

```
POST /auth/login
```

Request Body:

```
{
  "email": "user@example.com",
  "password": "SecurePass123!",
  "rememberMe": true,
  "deviceId": "device_abc123"
}
```

Response (without 2FA):

```
{
  "success": true,
  "data": {
    "accessToken": "eyJhbGciOiJIUzI1NiIs...",
    "refreshToken": "eyJhbGciOiJIUzI1NiIs...",
    "expiresIn": 3600,
    "user": { ... }
  }
}
```

Response (with 2FA enabled):

```
{
  "success": true,
  "data": {
    "requires2FA": true,
    "tempToken": "temp_token_for_2fa_verification"
  }
}
```

3.3 Verify 2FA During Login

```
POST /auth/2fa/verify
```

Request Body:

```
{
  "tempToken": "temp_token_for_2fa_verification",
  "code": "123456"
}
```

Response:

```
{
  "success": true,
  "data": {
    "accessToken": "eyJhbGciOiJIUzI1NiIs...",
    "refreshToken": "eyJhbGciOiJIUzI1NiIs...",
    "expiresIn": 3600,
    "user": { ... }
  }
}
```

3.4 Logout

```
POST /auth/logout
```

Headers: Authorization required

Response:

```json
{
  "success": true,
  "message": "Successfully logged out"
}
```

3.5 Refresh Token

```
POST /auth/refresh
```

Request Body:

```json
{
  "refreshToken": "eyJhbGciOiJIUzI1NiIs..."
}
```

Response:

```json
{
  "success": true,
  "data": {
    "accessToken": "new_access_token",
    "refreshToken": "new_refresh_token",
    "expiresIn": 3600
  }
}
```

3.6 Forgot Password

```
POST /auth/forgot-password
```

Request Body:

```json
{
  "email": "user@example.com"
}
```

Response:

```json
{
  "success": true,
  "message": "Password reset email sent"
}
```

3.7 Reset Password

```
POST /auth/reset-password
```

Request Body:

```json
{
  "token": "reset_token_from_email",
  "newPassword": "NewSecurePass123!"
}
```

3.8 Verify Email

```
POST /auth/verify-email
```

Request Body:

```json
{
  "code": "123456"
}
```

3.9 Resend Verification Email

```
POST /auth/resend-verification
```

Headers: Authorization required

3.10 Enable 2FA

```
POST /auth/2fa/enable
```

Headers: Authorization required

Response:

```json
{
  "success": true,
  "data": {
    "secret": "JBSWY3DPEHPK3PXP",
    "qrCode": "data:image/png;base64,..."
  }
}
```

3.11 Confirm 2FA Setup

```
POST /auth/2fa/confirm
```

Request Body:

```
{
  "code": "123456"
}
```

3.12 Disable 2FA

```
POST /auth/2fa/disable
```

Request Body:

```
{
  "code": "123456"
}
```

4. User Management Endpoints

4.1 Get User Profile

```
GET /user/profile
```

Response:

```
{
  "success": true,
  "data": {
    "id": "usr_123456",
    "email": "user@example.com",
    "name": "John Doe",
    "phone": "+1234567890",
    "avatar": "https://cdn.crymadx.com/avatars/user123.jpg",
    "country": "US",
    "language": "en",
    "currency": "USD",
    "kycLevel": 2,
    "kycStatus": "approved",
    "status": "active",
    "twoFactorEnabled": true,
    "emailVerified": true,
    "phoneVerified": false,
    "referralCode": "USER123ABC",
    "referredBy": "CRYMADX2024",
    "createdAt": "2026-01-01T00:00:00Z",
    "lastLogin": "2026-01-15T10:30:00Z"
  }
}
```

4.2 Update User Profile

```
PUT /user/profile
```

Request Body:

```
{
  "name": "John Smith",
  "phone": "+1234567890",
  "country": "US",
  "language": "en",
  "currency": "USD"
}
```

4.3 Upload Avatar

```
POST /user/profile/avatar
Content-Type: multipart/form-data
```

Request: Form data with `avatar` file field

Response:

```
{
  "success": true,
  "data": {
    "avatarUrl": "https://cdn.crymadx.com/avatars/user123_new.jpg"
  }
}
```

4.4 Change Password

```
POST /user/profile/password
```

Request Body:

```
{
  "currentPassword": "OldPass123!",
  "newPassword": "NewPass456!"
}
```

4.5 Get KYC Status

```
GET /user/kyc
```

Response:

```json
{
  "success": true,
  "data": {
    "level": 2,
    "status": "approved",
    "documents": [
      {
        "id": "doc_123",
        "type": "passport",
        "status": "approved",
        "uploadedAt": "2026-01-10T10:00:00Z"
      },
      {
        "id": "doc_124",
        "type": "selfie",
        "status": "approved",
        "uploadedAt": "2026-01-10T10:05:00Z"
      }
    ],
    "submittedAt": "2026-01-10T10:00:00Z",
    "approvedAt": "2026-01-11T14:00:00Z",
    "rejectionReason": null
  }
}
```

4.6 Submit KYC Documents

```
POST /user/kyc/documents
Content-Type: multipart/form-data
```

Request Fields:

- `type` : "passport" | "drivers_license" | "national_id" | "selfie" | "proof_of_address"
- `file` : Document image file (JPG, PNG, PDF)
- `side` : "front" | "back" (for ID documents)

Response:

```json
{
  "success": true,
  "data": {
    "id": "doc_125",
    "type": "passport",
    "side": "front",
    "status": "pending",
    "uploadedAt": "2026-01-15T10:30:00Z"
  }
}
```

4.7 Submit KYC for Review

```
POST /user/kyc/submit
```

Request Body:

```
{
  "firstName": "John",
  "lastName": "Doe",
  "dateOfBirth": "1990-01-15",
  "nationality": "US",
  "address": "123 Main Street",
  "city": "New York",
  "postalCode": "10001",
  "country": "US"
}
```

4.8 Get Referral Stats

```
GET /user/referrals
```

Response:

```
{
  "success": true,
  "data": {
    "referralCode": "USER123ABC",
    "referralLink": "https://crymadx.com/ref/USER123ABC",
    "totalReferrals": 12,
    "activeReferrals": 8,
    "totalEarnings": "240.00",
    "pendingEarnings": "50.00",
    "earningsHistory": [
      {
        "id": "ref_001",
        "referredUser": "u***@email.com",
        "amount": "20.00",
        "currency": "USDT",
        "status": "paid",
        "createdAt": "2026-01-10T10:00:00Z"
      }
    ],
    "referrals": [
      {
        "id": "ref_usr_001",
        "email": "j***@email.com",
        "status": "active",
        "tradingVolume": "5000.00",
        "commission": "50.00",
        "joinedAt": "2026-01-05T10:00:00Z"
      }
    ]
  }
}
```

4.9 Get Rewards

```
GET /user/rewards
```

Query Parameters:

- `status` : "available" | "claimed" | "expired" (optional)

Response:

```
{
  "success": true,
  "data": {
    "totalPoints": 575,
    "availableRewards": [
      {
        "id": "reward_001",
        "type": "welcome_bonus",
        "title": "Welcome Bonus",
        "description": "Sign up and complete your first verification",
        "amount": "10.00",
        "currency": "USDT",
        "points": 100,
        "status": "completed",
        "expiresAt": "2026-02-15T00:00:00Z",
        "createdAt": "2026-01-01T00:00:00Z"
      },
      {
        "id": "reward_002",
        "type": "first_trade",
        "title": "First Trade",
        "description": "Complete your first trade",
        "amount": "5.00",
        "currency": "USDT",
        "points": 50,
        "status": "available",
        "expiresAt": null,
        "createdAt": "2026-01-01T00:00:00Z"
      }
    ]
  }
}
```

4.10 Claim Reward

```
POST /user/rewards/{rewardId}/claim
```

Response:

```
{
  "success": true,
  "data": {
    "id": "reward_001",
    "amount": "10.00",
    "currency": "USDT",
    "status": "claimed",
    "claimedAt": "2026-01-15T10:30:00Z"
  }
}
```

4.11 Get Support Tickets

```
GET /user/support/tickets
```

Query Parameters:

- `status` : "open" | "in_progress" | "resolved" | "closed" (optional)
- `page` : Page number (default: 1)
- `limit` : Items per page (default: 20)

Response:

```
{
  "success": true,
  "data": {
    "tickets": [
      {
        "id": "TKT-001234",
        "subject": "Deposit not credited",
        "category": "deposit_withdrawal",
        "status": "in_progress",
        "priority": "high",
        "createdAt": "2026-01-14T09:00:00Z",
        "updatedAt": "2026-01-14T14:00:00Z",
        "lastMessageAt": "2026-01-14T14:00:00Z"
      }
    ]
  },
  "pagination": { ... }
}
```

4.12 Get Ticket Details

```
GET /user/support/tickets/{ticketId}
```

Response:

```
{
  "success": true,
  "data": {
    "id": "TKT-001234",
    "subject": "Deposit not credited",
    "category": "deposit_withdrawal",
    "status": "in_progress",
    "priority": "high",
    "createdAt": "2026-01-14T09:00:00Z",
    "messages": [
      {
        "id": "msg_001",
        "content": "My BTC deposit has not been credited after 2 hours",
        "sender": "user",
        "createdAt": "2026-01-14T09:00:00Z"
      },
      {
        "id": "msg_002",
        "content": "We are investigating this issue. Please provide your transaction hash.",
        "sender": "support",
        "agentName": "Support Team",
        "createdAt": "2026-01-14T14:00:00Z"
      }
    ],
    "attachments": []
  }
}
```

4.13 Create Support Ticket

```
POST /user/support/tickets
```

Request Body:

```
{
  "subject": "Unable to withdraw",
  "category": "deposit_withdrawal",
  "message": "I am getting an error when trying to withdraw BTC",
  "priority": "high"
}
```

Category Options:

- `deposit_withdrawal`
- `trading`
- `kyc`
- `account_security`
- `technical`
- `other`

Priority Options:

- `low`
- `medium`
- `high`

4.14 Reply to Ticket

```
POST /user/support/tickets/{ticketId}/reply
```

Request Body:

```
{
  "message": "Here is the transaction hash: 0x123abc..."
}
```

4.15 Get/Update Notification Settings

```
GET /user/notifications
PUT /user/notifications
```

Request/Response Body:

```
{
  "emailNotifications": true,
  "pushNotifications": true,
  "priceAlerts": true,
  "tradeNotifications": true,
  "securityAlerts": true,
  "marketingEmails": false
}
```

4.16 Get Payment Methods

```
GET /user/payment-methods
```

Response:

```json
{
  "success": true,
  "data": {
    "paymentMethods": [
      {
        "id": "pm_001",
        "type": "card",
        "brand": "visa",
        "last4": "4532",
        "expiryMonth": 12,
        "expiryYear": 2028,
        "holderName": "JOHN DOE",
        "isDefault": true,
        "isVerified": true,
        "createdAt": "2026-01-01T00:00:00Z"
      },
      {
        "id": "pm_002",
        "type": "bank_account",
        "bankName": "Chase Bank",
        "accountType": "checking",
        "last4": "6789",
        "holderName": "John Doe",
        "isDefault": false,
        "isVerified": true,
        "createdAt": "2026-01-05T00:00:00Z"
      }
    ]
  }
}
```

4.17 Add Payment Method

```
POST /user/payment-methods
```

Request Body (Card):

```json
{
  "type": "card",
  "cardNumber": "4532123456781234",
  "expiryMonth": 12,
  "expiryYear": 2028,
  "cvv": "123",
  "holderName": "JOHN DOE"
}
```

Request Body (Bank Account):

```
{
  "type": "bank_account",
  "bankName": "Chase Bank",
  "accountNumber": "123456789",
  "routingNumber": "021000021",
  "accountType": "checking",
  "holderName": "John Doe"
}
```

4.18 Delete Payment Method

```
DELETE /user/payment-methods/{paymentMethodId}
```

4.19 Set Default Payment Method

```
PUT /user/payment-methods/{paymentMethodId}/default
```

---

5. Wallet & Transaction Endpoints

5.1 Get All Balances

```
GET /wallet/balances
```

Query Parameters:

- `accountType` : "funding" | "trading" | "all" (default: "all")

Response:

```
{
  "success": true,
  "data": {
    "totalUsdValue": "15420.50",
    "balances": [
      {
        "currency": "Bitcoin",
        "symbol": "BTC",
        "available": "0.5",
        "locked": "0.1",
        "total": "0.6",
        "usdValue": "12000.00",
        "accountType": "funding"
      },
      {
        "currency": "Tether",
        "symbol": "USDT",
        "available": "1500.00",
        "locked": "0.00",
        "total": "1500.00",
        "usdValue": "1500.00",
        "accountType": "funding"
      },
      {
        "currency": "Ethereum",
        "symbol": "ETH",
        "available": "2.5",
        "locked": "0.0",
        "total": "2.5",
        "usdValue": "1920.50",
        "accountType": "trading"
      }
    ]
  }
}
```

5.2 Get Specific Balance

```
GET /wallet/balances/{currency}
```

Example: `GET /wallet/balances/BTC`

Response:

```
{
  "success": true,
  "data": {
    "currency": "Bitcoin",
    "symbol": "BTC",
    "available": "0.5",
    "locked": "0.1",
    "total": "0.6",
    "usdValue": "12000.00",
    "networks": ["BTC", "ERC20", "BEP20"],
    "canDeposit": true,
    "canWithdraw": true
  }
}
```

5.3 Get Deposit Address

```
GET /wallet/deposit
```

Query Parameters:

- `currency` : Currency symbol (e.g., "BTC", "ETH", "USDT")
- `network` : Network type (e.g., "BTC", "ERC20", "BEP20", "TRC20")

Response:

```
{
  "success": true,
  "data": {
    "currency": "BTC",
    "network": "BTC",
    "address": "bc1qxy2kgdygjrsqtzq2n0yrf2493p83kkfjhx0wlh",
    "memo": null,
    "qrCode": "data:image/png;base64,...",
    "minDeposit": "0.0001",
    "confirmations": 3,
    "estimatedTime": "30-60 minutes"
  }
}
```

5.4 Get Available Networks for Currency

```
GET /wallet/networks/{currency}
```

Response:

```
{
  "success": true,
  "data": {
    "currency": "USDT",
    "networks": [
      {
        "network": "ERC20",
        "name": "Ethereum",
        "minDeposit": "10",
        "minWithdraw": "20",
        "withdrawFee": "5",
        "depositEnabled": true,
        "withdrawEnabled": true,
        "confirmations": 12,
        "estimatedTime": "10-15 minutes"
      },
      {
        "network": "TRC20",
        "name": "Tron",
        "minDeposit": "1",
        "minWithdraw": "10",
        "withdrawFee": "1",
        "depositEnabled": true,
        "withdrawEnabled": true,
        "confirmations": 20,
        "estimatedTime": "3-5 minutes"
      },
      {
        "network": "BEP20",
        "name": "BNB Chain",
        "minDeposit": "5",
        "minWithdraw": "10",
        "withdrawFee": "0.5",
        "depositEnabled": true,
        "withdrawEnabled": true,
        "confirmations": 15,
        "estimatedTime": "5-10 minutes"
      }
    ]
  }
}
```

5.5 Request Withdrawal

```
POST /wallet/withdraw
```

Request Body:

```
{
  "currency": "BTC",
  "network": "BTC",
  "address": "bc1qxy2kgdygjrsqtzq2n0yrf2493p83kkfjhx0wlh",
  "amount": "0.1",
  "memo": null,
  "twoFaCode": "123456"
}
```

Response:

```
{
  "success": true,
  "data": {
    "id": "wd_123456",
    "currency": "BTC",
    "network": "BTC",
    "address": "bc1qxy2kgdygjrsqtzq2n0yrf2493p83kkfjhx0wlh",
    "amount": "0.1",
    "fee": "0.0005",
    "netAmount": "0.0995",
    "status": "pending",
    "createdAt": "2026-01-15T10:30:00Z"
  }
}
```

5.6 Get Withdrawal Fee

```
GET /wallet/withdraw/fee
```

Query Parameters:

- `currency` : Currency symbol
- `network` : Network type
- `amount` : Withdrawal amount

Response:

```json
{
  "success": true,
  "data": {
    "currency": "BTC",
    "network": "BTC",
    "fee": "0.0005",
    "minWithdraw": "0.001",
    "maxWithdraw": "10",
    "available": "0.5"
  }
}
```

5.7 Internal Transfer

```
POST /wallet/transfer
```

Request Body:

```json
{
  "fromAccount": "funding",
  "toAccount": "trading",
  "currency": "USDT",
  "amount": "500"
}
```

Account Types:

- `funding` : Main wallet for deposits/withdrawals
- `trading` : Trading account for spot trading

Response:

```json
{
  "success": true,
  "data": {
    "id": "tf_123456",
    "fromAccount": "funding",
    "toAccount": "trading",
    "currency": "USDT",
    "amount": "500",
    "status": "completed",
    "createdAt": "2026-01-15T10:30:00Z"
  }
}
```

5.8 Get Transaction History

```
GET /wallet/transactions
```

Query Parameters:

- `type` : "deposit" | "withdrawal" | "transfer" | "trade" | "earn" | "p2p" (optional)
- `currency` : Currency symbol (optional)
- `status` : "pending" | "completed" | "failed" | "cancelled" (optional)
- `startDate` : ISO date string (optional)
- `endDate` : ISO date string (optional)
- `page` : Page number (default: 1)
- `limit` : Items per page (default: 20)

Response:

```json
{
  "success": true,
  "data": {
    "transactions": [
      {
        "id": "tx_001",
        "type": "deposit",
        "currency": "BTC",
        "amount": "0.5",
        "usdValue": "10000.00",
        "status": "completed",
        "txHash": "0x123abc...",
        "network": "BTC",
        "address": "bc1q...",
        "fee": "0",
        "confirmations": 6,
        "createdAt": "2026-01-14T10:00:00Z",
        "completedAt": "2026-01-14T11:00:00Z"
      },
      {
        "id": "tx_002",
        "type": "withdrawal",
        "currency": "ETH",
        "amount": "1.0",
        "usdValue": "800.00",
        "status": "pending",
        "txHash": null,
        "network": "ERC20",
        "address": "0xabc...",
        "fee": "0.005",
        "createdAt": "2026-01-15T09:00:00Z",
        "completedAt": null
      },
      {
        "id": "tx_003",
        "type": "transfer",
        "currency": "USDT",
        "amount": "500",
        "usdValue": "500.00",
        "status": "completed",
        "fromAccount": "funding",
        "toAccount": "trading",
        "createdAt": "2026-01-15T10:30:00Z",
        "completedAt": "2026-01-15T10:30:00Z"
      }
    ]
  },
  "pagination": { ... }
}
```

6. Trading Endpoints

6.1 Get All Market Tickers

```
GET /trading/tickers
```

Response:

```
{
  "success": true,
  "data": {
    "tickers": [
      {
        "symbol": "BTCUSDT",
        "baseAsset": "BTC",
        "quoteAsset": "USDT",
        "price": "42500.00",
        "change24h": "1250.00",
        "changePercent24h": "3.02",
        "high24h": "43000.00",
        "low24h": "41000.00",
        "volume24h": "12500.5",
        "quoteVolume24h": "531271250.00",
        "openPrice": "41250.00",
        "bidPrice": "42499.00",
        "askPrice": "42501.00",
        "lastTradeTime": "2026-01-15T10:30:00Z"
      }
    ]
  }
}
```

6.2 Get Specific Ticker

```
GET /trading/tickers/{symbol}
```

Example: `GET /trading/tickers/BTCUSDT`

6.3 Get Order Book

```
GET /trading/orderbook/{symbol}
```

Query Parameters:

- `limit` : Number of levels (default: 20, max: 100)

Response:

```json
{
  "success": true,
  "data": {
    "symbol": "BTCUSDT",
    "bids": [
      ["42499.00", "0.5"],
      ["42498.00", "1.2"],
      ["42497.00", "0.8"]
    ],
    "asks": [
      ["42501.00", "0.3"],
      ["42502.00", "0.9"],
      ["42503.00", "1.5"]
    ],
    "timestamp": "2026-01-15T10:30:00Z"
  }
}
```

6.4 Get Candlestick/Kline Data

```
GET /trading/klines/{symbol}
```

Query Parameters:

- `interval` : "1m" | "5m" | "15m" | "30m" | "1h" | "4h" | "1d" | "1w"
- `limit` : Number of candles (default: 100, max: 1000)
- `startTime` : Unix timestamp (optional)
- `endTime` : Unix timestamp (optional)

Response:

```json
{
  "success": true,
  "data": {
    "symbol": "BTCUSDT",
    "interval": "1h",
    "klines": [
      {
        "openTime": 1705312800000,
        "open": "42000.00",
        "high": "42500.00",
        "low": "41800.00",
        "close": "42300.00",
        "volume": "150.5",
        "closeTime": 1705316400000,
        "quoteVolume": "6339150.00"
      }
    ]
  }
}
```

6.5 Get Recent Trades

```
GET /trading/trades/{symbol}
```

Query Parameters:

- `limit` : Number of trades (default: 50, max: 500)

Response:

```
{
  "success": true,
  "data": {
    "trades": [
      {
        "id": "trade_001",
        "symbol": "BTCUSDT",
        "price": "42500.00",
        "quantity": "0.05",
        "quoteQuantity": "2125.00",
        "side": "buy",
        "timestamp": "2026-01-15T10:30:00Z"
      }
    ]
  }
}
```

6.6 Create Order

```
POST /trading/orders
```

Request Body (Market Order):

```
{
  "symbol": "BTCUSDT",
  "side": "buy",
  "type": "market",
  "quantity": "0.1"
}
```

Request Body (Limit Order):

```
{
  "symbol": "BTCUSDT",
  "side": "sell",
  "type": "limit",
  "quantity": "0.1",
  "price": "43000.00"
}
```

Request Body (Stop-Limit Order):

```
{
  "symbol": "BTCUSDT",
  "side": "sell",
  "type": "stop_limit",
  "quantity": "0.1",
  "price": "42000.00",
  "stopPrice": "42500.00"
}
```

Order Types:

- `market` : Execute immediately at market price
- `limit` : Execute at specified price or better
- `stop_limit` : Trigger limit order when stop price is reached

Order Sides:

- `buy` : Buy base asset
- `sell` : Sell base asset

Response:

```
{
  "success": true,
  "data": {
    "id": "ord_123456",
    "symbol": "BTCUSDT",
    "side": "buy",
    "type": "limit",
    "status": "open",
    "price": "43000.00",
    "quantity": "0.1",
    "filledQuantity": "0",
    "stopPrice": null,
    "createdAt": "2026-01-15T10:30:00Z",
    "updatedAt": "2026-01-15T10:30:00Z"
  }
}
```

6.7 Get User Orders

```
GET /trading/orders
```

Query Parameters:

- `symbol` : Trading pair (optional)
- `status` : "open" | "filled" | "cancelled" | "partially_filled" (optional)
- `side` : "buy" | "sell" (optional)
- `page` : Page number
- `limit` : Items per page

6.8 Get Order Details

```
GET /trading/orders/{orderId}
```

6.9 Cancel Order

```
DELETE /trading/orders/{orderId}
```

Response:

```
{
  "success": true,
  "data": {
    "id": "ord_123456",
    "status": "cancelled",
    "cancelledAt": "2026-01-15T10:35:00Z"
  }
}
```

6.10 Cancel All Orders

```
DELETE /trading/orders
```

Query Parameters:

- `symbol` : Trading pair (required)

6.11 Get Trade History

```
GET /trading/history
```

Query Parameters:

- `symbol` : Trading pair (optional)
- `startTime` : Unix timestamp (optional)
- `endTime` : Unix timestamp (optional)
- `page` : Page number
- `limit` : Items per page

Response:

```json
{
  "success": true,
  "data": {
    "trades": [
      {
        "id": "trade_usr_001",
        "orderId": "ord_123456",
        "symbol": "BTCUSDT",
        "side": "buy",
        "price": "42500.00",
        "quantity": "0.1",
        "quoteQuantity": "4250.00",
        "fee": "4.25",
        "feeCurrency": "USDT",
        "timestamp": "2026-01-15T10:30:00Z"
      }
    ]
  },
  "pagination": { ... }
}
```

6.12 Convert Crypto (Instant Swap)

```
POST /trading/convert
```

Request Body:

```json
{
  "fromCurrency": "BTC",
  "toCurrency": "ETH",
  "fromAmount": "0.1"
}
```

Response:

```json
{
  "success": true,
  "data": {
    "id": "conv_123456",
    "fromCurrency": "BTC",
    "toCurrency": "ETH",
    "fromAmount": "0.1",
    "toAmount": "1.75",
    "rate": "17.5",
    "fee": "0.001",
    "status": "completed",
    "createdAt": "2026-01-15T10:30:00Z"
  }
}
```

6.13 Get Convert Quote

```
GET /trading/convert/quote
```

Query Parameters:

- `fromCurrency` : Source currency
- `toCurrency` : Destination currency
- `fromAmount` : Amount to convert

Response:

```
{
  "success": true,
  "data": {
    "fromCurrency": "BTC",
    "toCurrency": "ETH",
    "fromAmount": "0.1",
    "toAmount": "1.75",
    "rate": "17.5",
    "fee": "0.001",
    "expiresAt": "2026-01-15T10:35:00Z"
  }
}
```

---

7. P2P Trading Endpoints

7.1 Get P2P Ads/Listings

```
GET /p2p/ads
```

Query Parameters:

- `crypto` : Crypto currency (e.g., "USDT", "BTC")
- `fiat` : Fiat currency (e.g., "NGN", "USD", "EUR")
- `side` : "buy" | "sell"
- `paymentMethod` : Payment method filter (optional)
- `amount` : Filter by amount (optional)
- `page` : Page number

- `limit` : Items per page (default: 20)

Response:

```
{
  "success": true,
  "data": {
    "ads": [
      {
        "id": "ad_001",
        "merchant": {
          "id": "merchant_001",
          "name": "CryptoKing",
          "avatar": "https://...",
          "completedOrders": 1250,
          "completionRate": 98.5,
          "rating": 4.9,
          "isVerified": true,
          "isOnline": true
        },
        "crypto": "USDT",
        "fiat": "NGN",
        "side": "sell",
        "price": "1650.00",
        "minAmount": "10000",
        "maxAmount": "500000",
        "available": "5000",
        "paymentMethods": ["bank_transfer", "opay", "palmpay"],
        "timeLimit": 15,
        "terms": "Fast payment required",
        "createdAt": "2026-01-15T10:00:00Z"
      }
    ]
  },
  "pagination": { ... }
}
```

7.2 Create P2P Order

```
POST /p2p/orders
```

Request Body:

```
{
  "adId": "ad_001",
  "amount": "100000",
  "paymentMethod": "bank_transfer"
}
```

Response:

```
{
  "success": true,
  "data": {
    "id": "p2p_ord_001",
    "adId": "ad_001",
    "merchant": { ... },
    "crypto": "USDT",
    "fiat": "NGN",
    "side": "buy",
    "price": "1650.00",
    "cryptoAmount": "60.60",
    "fiatAmount": "100000",
    "paymentMethod": "bank_transfer",
    "paymentDetails": {
      "bankName": "GTBank",
      "accountNumber": "0123456789",
      "accountName": "CRYPTO KING LTD"
    },
    "status": "pending_payment",
    "timeLimit": 15,
    "expiresAt": "2026-01-15T10:45:00Z",
    "createdAt": "2026-01-15T10:30:00Z"
  }
}
```

7.3 Get P2P Orders

```
GET /p2p/orders
```

Query Parameters:

- `status` : "pending_payment" | "paid" | "completed" | "cancelled" | "disputed" (optional)
- `page` : Page number
- `limit` : Items per page

7.4 Get P2P Order Details

```
GET /p2p/orders/{orderId}
```

7.5 Mark P2P Order as Paid

```
POST /p2p/orders/{orderId}/paid
```

Request Body:

```
{
  "paymentProof": "https://..." // Optional screenshot URL
}
```

7.6 Release P2P Order (Merchant)

```
POST /p2p/orders/{orderId}/release
```

7.7 Cancel P2P Order

```
POST /p2p/orders/{orderId}/cancel
```

Request Body:

```
{
  "reason": "Changed my mind"
}
```

7.8 Dispute P2P Order

```
POST /p2p/orders/{orderId}/dispute
```

Request Body:

```
{
  "reason": "Payment not received",
  "evidence": ["https://proof1.jpg", "https://proof2.jpg"]
}
```

---

8. Earn & Staking Endpoints

8.1 Get Earn Products

```
GET /earn/products
```

Query Parameters:

- `type` : "flexible" | "locked" (optional)
- `currency` : Currency filter (optional)

Response:

```
{
  "success": true,
  "data": {
    "products": [
      {
        "id": "earn_001",
        "type": "flexible",
        "currency": "USDT",
        "name": "USDT Flexible Savings",
        "apy": "5.5",
        "minAmount": "10",
        "maxAmount": "100000",
        "totalDeposited": "5000000",
        "availableQuota": "1000000",
        "tierRates": [
          {"min": "0", "max": "1000", "apy": "5.5"},
          {"min": "1000", "max": "10000", "apy": "5.0"},
          {"min": "10000", "max": "100000", "apy": "4.5"}
        ],
        "redemptionPeriod": 0,
        "status": "active"
      },
      {
        "id": "earn_002",
        "type": "locked",
        "currency": "BTC",
        "name": "BTC 30-Day Lock",
        "apy": "4.0",
        "minAmount": "0.001",
        "maxAmount": "10",
        "lockPeriod": 30,
        "totalDeposited": "500",
        "availableQuota": "100",
        "status": "active"
      }
    ]
  }
}
```

8.2 Subscribe to Earn Product

```
POST /earn/subscribe
```

Request Body:

```
{
  "productId": "earn_001",
  "amount": "1000"
}
```

Response:

```
{
  "success": true,
  "data": {
    "id": "sub_001",
    "productId": "earn_001",
    "currency": "USDT",
    "amount": "1000",
    "apy": "5.5",
    "type": "flexible",
    "lockPeriod": null,
    "status": "active",
    "startDate": "2026-01-15T10:30:00Z",
    "maturityDate": null,
    "accruedInterest": "0"
  }
}
```

8.3 Get My Earnings

```
GET /earn/positions
```

Response:

```
{
  "success": true,
  "data": {
    "totalValue": "1050.50",
    "totalEarnings": "50.50",
    "positions": [
      {
        "id": "sub_001",
        "productId": "earn_001",
        "currency": "USDT",
        "amount": "1000",
        "apy": "5.5",
        "type": "flexible",
        "accruedInterest": "50.50",
        "status": "active",
        "startDate": "2025-12-15T10:30:00Z",
        "canRedeem": true
      }
    ]
  }
}
```

8.4 Redeem Earnings

```
POST /earn/redeem
```

Request Body:

```
{
  "positionId": "sub_001",
  "amount": "500"
}
```

8.5 Get Staking Products

```
GET /staking/products
```

Response:

```json
{
  "success": true,
  "data": {
    "products": [
      {
        "id": "stake_001",
        "currency": "ETH",
        "name": "ETH 2.0 Staking",
        "type": "liquid",
        "apy": "4.5",
        "minStake": "0.01",
        "totalStaked": "50000",
        "validators": 150,
        "lockPeriod": 0,
        "unstakePeriod": 7,
        "status": "active"
      },
      {
        "id": "stake_002",
        "currency": "SOL",
        "name": "Solana Staking",
        "type": "lock",
        "apy": "7.2",
        "minStake": "1",
        "totalStaked": "100000",
        "validators": 200,
        "lockPeriod": 30,
        "unstakePeriod": 3,
        "status": "active"
      }
    ]
  }
}
```

8.6 Stake Crypto

```
POST /staking/stake
```

Request Body:

```json
{
  "productId": "stake_001",
  "amount": "1.5"
}
```

8.7 Get Staking Positions

```
GET /staking/positions
```

8.8 Unstake Crypto

```
POST /staking/unstake
```

Request Body:

```json
{
  "positionId": "stake_pos_001",
  "amount": "0.5"
}
```

---

9. NFT Marketplace Endpoints

9.1 Get NFT Listings

```
GET /nft/listings
```

Query Parameters:

- `collection` : Collection ID (optional)
- `minPrice` : Minimum price filter (optional)
- `maxPrice` : Maximum price filter (optional)
- `sort` : "price_asc" | "price_desc" | "recent" (optional)
- `page` : Page number
- `limit` : Items per page

Response:

```json
{
  "success": true,
  "data": {
    "listings": [
      {
        "id": "nft_001",
        "tokenId": "1234",
        "name": "Cosmic Ape #1234",
        "description": "A unique cosmic ape from the universe",
        "image": "https://...",
        "collection": {
          "id": "col_001",
          "name": "Cosmic Apes",
          "verified": true
        },
        "owner": {
          "id": "user_001",
          "name": "NFTCollector",
          "address": "0x..."
        },
        "price": "0.5",
        "currency": "ETH",
        "status": "listed",
        "createdAt": "2026-01-15T10:00:00Z"
      }
    ]
  },
  "pagination": { ... }
}
```

9.2 Get My NFTs

```
GET /nft/owned
```

9.3 Get NFT Collections

```
GET /nft/collections
```

Response:

```
{
  "success": true,
  "data": {
    "collections": [
      {
        "id": "col_001",
        "name": "Cosmic Apes",
        "description": "10,000 unique cosmic apes",
        "image": "https://...",
        "banner": "https://...",
        "verified": true,
        "totalItems": 10000,
        "owners": 5432,
        "floorPrice": "0.5",
        "volume24h": "150.5",
        "volumeTotal": "25000"
      }
    ]
  }
}
```

9.4 Buy NFT

```
POST /nft/buy
```

Request Body:

```json
{
  "listingId": "nft_001"
}
```

9.5 List NFT for Sale

```
POST /nft/list
```

Request Body:

```json
{
  "tokenId": "1234",
  "collectionId": "col_001",
  "price": "0.8",
  "currency": "ETH"
}
```

9.6 Cancel NFT Listing

```
DELETE /nft/listings/{listingId}
```

9.7 Get NFT Activity

```
GET /nft/activity
```

Query Parameters:

- `type` : "sale" | "listing" | "transfer" | "mint" (optional)
- `page` : Page number
- `limit` : Items per page

---

10. Fiat On-Ramp Endpoints

10.1 Get Fiat Providers

```
GET /fiat/providers
```

Query Parameters:

- `crypto` : Crypto to buy (e.g., "BTC")
- `fiat` : Fiat currency (e.g., "USD")
- `type` : "buy" | "sell"

Response:

```json
{
  "success": true,
  "data": {
    "providers": [
      {
        "id": "transak",
        "name": "Transak",
        "logo": "https://...",
        "feePercent": "1.5",
        "minAmount": "30",
        "maxAmount": "5000",
        "paymentMethods": ["card", "bank_transfer", "apple_pay"],
        "estimatedTime": "5-30 minutes",
        "rating": 4.5,
        "supported": true
      },
      {
        "id": "moonpay",
        "name": "MoonPay",
        "logo": "https://...",
        "feePercent": "2.0",
        "minAmount": "20",
        "maxAmount": "10000",
        "paymentMethods": ["card", "bank_transfer", "google_pay"],
        "estimatedTime": "10-30 minutes",
        "rating": 4.3,
        "supported": true
      }
    ]
  }
}
```

10.2 Get Fiat Quote

```
GET /fiat/quote
```

Query Parameters:

- `provider` : Provider ID
- `crypto` : Crypto currency
- `fiat` : Fiat currency
- `fiatAmount` : Amount in fiat
- `type` : "buy" | "sell"

Response:

```json
{
  "success": true,
  "data": {
    "provider": "transak",
    "crypto": "BTC",
    "fiat": "USD",
    "fiatAmount": "100",
    "cryptoAmount": "0.00235",
    "rate": "42553.19",
    "fee": "1.50",
    "totalCost": "101.50",
    "expiresAt": "2026-01-15T10:35:00Z"
  }
}
```

10.3 Create Fiat Order

```
POST /fiat/orders
```

Request Body:

```json
{
  "provider": "transak",
  "crypto": "BTC",
  "fiat": "USD",
  "fiatAmount": "100",
  "paymentMethod": "card",
  "type": "buy"
}
```

Response:

```json
{
  "success": true,
  "data": {
    "id": "fiat_ord_001",
    "provider": "transak",
    "redirectUrl": "https://transak.com/checkout?orderId=...",
    "status": "pending",
    "expiresAt": "2026-01-15T11:00:00Z"
  }
}
```

10.4 Get Fiat Order Status

```
GET /fiat/orders/{orderId}
```

11. All Screens & Their Requirements

#	Screen Name	Route	Backend Requirements
1	Splash	`/`	None (static)
2	Login	`/login`	Auth endpoints
3	Register	`/register`	Auth endpoints
4	Home	`/main/home`	Tickers, Balances
5	Markets	`/main/markets`	Tickers list
6	Trading	`/main/trade`	Tickers, Order book, Trades, Orders
7	Assets	`/main/assets`	Balances, Transaction history
8	Deposit	`/deposit`	Deposit address, Networks
9	Withdraw	`/withdraw`	Withdrawal API, Fee estimate
10	Transfer	`/transfer`	Internal transfer API
11	Transaction History	`/transactions`	Transaction history
12	P2P	`/p2p`	P2P ads, Orders
13	Convert	`/convert`	Convert quote, Convert API
14	Earn	`/earn`	Earn products, Positions
15	Stake	`/stake`	Staking products, Positions
16	NFT	`/nft`	NFT listings, Collections
17	Fiat	`/fiat`	Fiat providers, Quotes
18	Profile	`/profile`	User profile
19	KYC	`/kyc`	KYC status, Document upload
20	Rewards	`/rewards`	Rewards list, Claim
21	Referral	`/referral`	Referral stats
22	Notifications	`/notifications`	Notification settings
23	Support Tickets	`/tickets`	Support tickets
24	Payment Methods	`/payment-methods`	Payment methods CRUD
25	Coin Detail	`/coin-detail`	Balance, History for specific coin

12. All Buttons & Actions

Authentication

Button	Screen	API Call
Sign In	Login	`POST /auth/login`
Create Account	Register	`POST /auth/register`
Forgot Password	Login	`POST /auth/forgot-password`
Google Sign-In	Login/Register	OAuth integration
Logout	Profile	`POST /auth/logout`

Wallet Operations

Button	Screen	API Call
Deposit	Home/Assets	Navigate + `GET /wallet/deposit`
Withdraw	Home/Assets	Navigate + `POST /wallet/withdraw`
Transfer	Home/Assets	`POST /wallet/transfer`
Send	Home	Navigate to Transfer
Buy	Home	Navigate to Trading/Fiat
Copy Address	Deposit	Client-side
Max (amount)	Withdraw/Transfer	Client-side

Trading

Button	Screen	API Call
Buy Order	Trading	`POST /trading/orders`
Sell Order	Trading	`POST /trading/orders`
Cancel Order	Trading	`DELETE /trading/orders/{id}`
Favorite	Trading/Markets	Client-side storage
Convert	Convert	`POST /trading/convert`
Swap Currencies	Convert	Client-side

P2P

Button	Screen	API Call
Create Order	P2P	`POST /p2p/orders`
Mark Paid	P2P Order	`POST /p2p/orders/{id}/paid`

Release	P2P Order	`POST /p2p/orders/{id}/release`
Cancel	P2P Order	`POST /p2p/orders/{id}/cancel`
Dispute	P2P Order	`POST /p2p/orders/{id}/dispute`

Earn/Stake

Button	Screen	API Call
Subscribe	Earn	`POST /earn/subscribe`
Redeem	Earn	`POST /earn/redeem`
Stake	Stake	`POST /staking/stake`
Unstake	Stake	`POST /staking/unstake`

Profile/Settings

Button	Screen	API Call
Update Profile	Profile	`PUT /user/profile`
Change Password	Profile	`POST /user/profile/password`
Enable 2FA	Profile	`POST /auth/2fa/enable`
Disable 2FA	Profile	`POST /auth/2fa/disable`
Submit KYC	KYC	`POST /user/kyc/submit`
Upload Document	KYC	`POST /user/kyc/documents`
Claim Reward	Rewards	`POST /user/rewards/{id}/claim`
Copy Referral	Referral	Client-side
New Ticket	Support	`POST /user/support/tickets`
Add Payment	Payments	`POST /user/payment-methods`

---

13. Form Fields & Validation

Login Form

Field	Type	Validation
email	email	Required, valid email
password	password	Required, min 8 chars
rememberMe	boolean	Optional

Register Form

Field	Type	Validation
name	text	Required, 2-100 chars
email	email	Required, valid email, unique
password	password	Required, min 8, uppercase, lowercase, number, special
confirmPassword	password	Must match password
referralCode	text	Optional
termsAccepted	boolean	Required, must be true

Withdraw Form

Field	Type	Validation
currency	select	Required
network	select	Required
address	text	Required, valid address format
memo	text	Optional (required for some coins)
amount	number	Required, > min, <= available
twoFaCode	text	Required if 2FA enabled

KYC Form

Field	Type	Validation
firstName	text	Required
lastName	text	Required
dateOfBirth	date	Required, must be 18+
nationality	select	Required
documentType	select	Required
documentFront	file	Required, image/pdf, max 5MB
documentBack	file	Optional, image/pdf, max 5MB
selfie	file	Required, image, max 5MB
address	text	Required
city	text	Required
postalCode	text	Required
country	select	Required

Trading Order Form

Field	Type	Validation
symbol	text	Required
side	select	Required: "buy" or "sell"
type	select	Required: "market", "limit", "stop_limit"
quantity	number	Required, > 0
price	number	Required for limit orders
stopPrice	number	Required for stop-limit orders

Support Ticket Form

Field	Type	Validation
subject	text	Required, 5-100 chars
category	select	Required
message	textarea	Required, 10-2000 chars
priority	select	Required
attachment	file	Optional, max 10MB

---

14. Data Models

User

```
{
  id: string;
  email: string;
  name: string;
  phone?: string;
  avatar?: string;
  country?: string;
  language: string;
  currency: string;
  kycLevel: 0 | 1 | 2 | 3;
  kycStatus: 'none' | 'pending' | 'approved' | 'rejected';
  status: 'active' | 'suspended' | 'banned';
  twoFactorEnabled: boolean;
  emailVerified: boolean;
  phoneVerified: boolean;
  referralCode: string;
  referredBy?: string;
  createdAt: datetime;
  lastLogin: datetime;
}
```

WalletBalance

```
{
  currency: string;
  symbol: string;
  available: decimal;
  locked: decimal;
  total: decimal;
  usdValue: decimal;
  accountType: 'funding' | 'trading';
}
```

Transaction

```
{
  id: string;
  type: 'deposit' | 'withdrawal' | 'transfer' | 'trade' | 'earn' | 'p2p';
  currency: string;
  amount: decimal;
  usdValue: decimal;
  fee?: decimal;
  status: 'pending' | 'completed' | 'failed' | 'cancelled';
  txHash?: string;
  network?: string;
  address?: string;
  fromAccount?: string;
  toAccount?: string;
  confirmations?: number;
  createdAt: datetime;
  completedAt?: datetime;
}
```

Order

```
{
  id: string;
  symbol: string;
  side: 'buy' | 'sell';
  type: 'market' | 'limit' | 'stop_limit';
  status: 'open' | 'filled' | 'partially_filled' | 'cancelled';
  price?: decimal;
  stopPrice?: decimal;
  quantity: decimal;
  filledQuantity: decimal;
  avgPrice?: decimal;
  fee?: decimal;
  feeCurrency?: string;
  createdAt: datetime;
  updatedAt: datetime;
}
```

Ticker

```
{
  symbol: string;
  baseAsset: string;
  quoteAsset: string;
  price: decimal;
  change24h: decimal;
  changePercent24h: decimal;
  high24h: decimal;
  low24h: decimal;
  volume24h: decimal;
  quoteVolume24h: decimal;
  bidPrice: decimal;
  askPrice: decimal;
  lastTradeTime: datetime;
}
```

P2PAd

```
{
  id: string;
  merchant: {
    id: string;
    name: string;
    avatar?: string;
    completedOrders: number;
    completionRate: decimal;
    rating: decimal;
    isVerified: boolean;
    isOnline: boolean;
  };
  crypto: string;
  fiat: string;
  side: 'buy' | 'sell';
  price: decimal;
  minAmount: decimal;
  maxAmount: decimal;
  available: decimal;
  paymentMethods: string[];
  timeLimit: number;
  terms?: string;
  createdAt: datetime;
}
```

SupportTicket

```
{
  id: string;
  subject: string;
  category: string;
  status: 'open' | 'in_progress' | 'resolved' | 'closed';
  priority: 'low' | 'medium' | 'high';
  messages: {
    id: string;
    content: string;
    sender: 'user' | 'support';
    agentName?: string;
    createdAt: datetime;
  }[];
  attachments: string[];
  createdAt: datetime;
  updatedAt: datetime;
}
```

---

15. WebSocket Events

Connection

```
URL: wss://api.crymadx.com/ws
Auth: ?token={accessToken}
```

Subscribe to Channels

```
{
  "action": "subscribe",
  "channels": ["ticker.BTCUSDT", "orderbook.BTCUSDT", "trades.BTCUSDT"]
}
```

Channel Events

Ticker Update:

```
{
  "channel": "ticker.BTCUSDT",
  "data": {
    "symbol": "BTCUSDT",
    "price": "42500.00",
    "change24h": "1250.00",
    ...
  }
}
```

Order Book Update:

```
{
  "channel": "orderbook.BTCUSDT",
  "data": {
    "bids": [...],
    "asks": [...],
    "timestamp": "2026-01-15T10:30:00Z"
  }
}
```

Trade Update:

```
{
  "channel": "trades.BTCUSDT",
  "data": {
    "id": "trade_001",
    "price": "42500.00",
    "quantity": "0.1",
    "side": "buy",
    "timestamp": "2026-01-15T10:30:00Z"
  }
}
```

User Order Update:

```
{
  "channel": "user.orders",
  "data": {
    "id": "ord_123",
    "status": "filled",
    ...
  }
}
```

Balance Update:

```
{
  "channel": "user.balance",
  "data": {
    "currency": "BTC",
    "available": "0.5",
    "locked": "0.1"
  }
}
```

---

16. Error Codes

Code	HTTP Status	Description
`AUTH_INVALID_CREDENTIALS`	401	Invalid email or password
`AUTH_TOKEN_EXPIRED`	401	Access token has expired
`AUTH_TOKEN_INVALID`	401	Invalid access token
`AUTH_2FA_REQUIRED`	403	2FA verification required
`AUTH_2FA_INVALID`	400	Invalid 2FA code
`USER_NOT_FOUND`	404	User does not exist
`USER_EMAIL_EXISTS`	409	Email already registered
`USER_NOT_VERIFIED`	403	Email not verified
`USER_SUSPENDED`	403	Account is suspended
`KYC_REQUIRED`	403	KYC verification required
`KYC_PENDING`	403	KYC is pending review
`WALLET_INSUFFICIENT_BALANCE`	400	Insufficient balance
`WALLET_INVALID_ADDRESS`	400	Invalid withdrawal address
`WALLET_MIN_AMOUNT`	400	Amount below minimum
`WALLET_MAX_AMOUNT`	400	Amount above maximum
`WALLET_WITHDRAWAL_DISABLED`	403	Withdrawals disabled for this currency
`ORDER_INVALID_SYMBOL`	400	Invalid trading pair
`ORDER_INVALID_QUANTITY`	400	Invalid order quantity
`ORDER_INVALID_PRICE`	400	Invalid order price
`ORDER_NOT_FOUND`	404	Order not found
`ORDER_ALREADY_FILLED`	400	Order already filled
`ORDER_ALREADY_CANCELLED`	400	Order already cancelled
`P2P_AD_NOT_FOUND`	404	P2P ad not found
`P2P_INSUFFICIENT_AMOUNT`	400	Amount exceeds available
`P2P_ORDER_EXPIRED`	400	P2P order has expired
`RATE_LIMIT_EXCEEDED`	429	Too many requests
`VALIDATION_ERROR`	400	Request validation failed

`INTERNAL_ERROR`	500	Internal server error

---

17. Security Requirements

Authentication

- JWT tokens with short expiry (1 hour)
- Refresh tokens with longer expiry (7-30 days)
- Token blacklisting on logout
- Device tracking for sessions

Password Requirements

- Minimum 8 characters
- At least 1 uppercase letter
- At least 1 lowercase letter
- At least 1 number
- At least 1 special character
- Bcrypt hashing with cost factor 12+

2FA Implementation

- TOTP (Time-based One-Time Password)
- Google Authenticator compatible
- Backup codes for recovery
- Required for sensitive operations

Rate Limiting

- Authentication endpoints: 5 requests/minute
- Trading endpoints: 100 requests/minute
- General API: 300 requests/minute
- WebSocket: 50 messages/second

Input Validation

- Sanitize all user inputs
- Validate addresses format per network
- Prevent SQL injection
- Prevent XSS attacks

Withdrawal Security

- Email confirmation for withdrawals
- 2FA required for large amounts
- 24-hour lock after password change
- Whitelist addresses option

Data Protection

- HTTPS only (TLS 1.3)
- Sensitive data encryption at rest
- PCI DSS compliance for card data
- GDPR compliance for EU users

Summary Statistics

- **Total API Endpoints:** 85+
- **Total Screens:** 25
- **Total Interactive Elements:** 100+
- **Total Form Fields:** 50+
- **WebSocket Channels:** 5+
- **Error Codes:** 30+

Document End

This document was generated for CrymadX Flutter Mobile App v1.0 For questions, contact the development team.