

API Documentation

This document provides comprehensive documentation for the CapitalFlow Portal API endpoints.

Base URL

```
Development: http://localhost:3000
Production: https://your-domain.com
```

Authentication

The API uses NextAuth.js for authentication. Most endpoints require authentication.

Authentication Flow

1. Sign Up

```
````javascript
POST /api/auth/signup
Content-Type: application/json

{
 "email": "user@example.com",
 "password": "securePassword123",
 "name": "John Doe",
 "companyName": "Acme Corp"
}
````
```

1. Sign In

```
````javascript
POST /api/auth/signin
Content-Type: application/json

{
 "email": "user@example.com",
 "password": "securePassword123"
}
````
```

1. Sign Out

```
javascript
POST /api/auth/signout
```

Dashboard Endpoints

Get Dashboard Data

```
GET /api/dashboard
Authorization: Bearer <token>
```

Response:

```
{
  "success": true,
  "data": {
    "totalBalance": 125000.50,
    "activeDeals": 3,
    "monthlyPayment": 2547.89,
    "nextPaymentDate": "2024-02-15",
    "deals": [
      {
        "id": "deal_123",
        "dealName": "Business Expansion Loan",
        "principalAmount": 100000,
        "remainingBalance": 85000,
        "interestRate": 6.25,
        "status": "ACTIVE"
      }
    ],
    "recentTransactions": [
      {
        "id": "txn_456",
        "amount": 2547.89,
        "description": "Monthly Payment",
        "date": "2024-01-15",
        "type": "PAYMENT"
      }
    ]
  }
}
```



Transaction Endpoints

Get Transactions

```
GET /api/transactions?page=1&limit=20&type=PAYMENT
Authorization: Bearer <token>
```

Query Parameters:

- `page` (optional): Page number (default: 1)
- `limit` (optional): Items per page (default: 20, max: 100)
- `type` (optional): Filter by transaction type
- `startDate` (optional): Start date filter (ISO 8601)
- `endDate` (optional): End date filter (ISO 8601)

Response:

```
{
  "success": true,
  "data": {
    "transactions": [
      {
        "id": "txn_123",
        "amount": 2547.89,
        "description": "Monthly Payment - Deal CF240001",
        "merchantName": "CapitalFlow",
        "category": "loan_payment",
        "transactionDate": "2024-01-15T10:30:00Z",
        "status": "COMPLETED",
        "type": "PAYMENT",
        "dealId": "deal_123"
      }
    ],
    "pagination": {
      "currentPage": 1,
      "totalPages": 5,
      "totalItems": 95,
      "hasNext": true,
      "hasPrevious": false
    }
  }
}
```

Create Transaction

POST /api/transactions
 Authorization: Bearer <token>
 Content-Type: application/json

```
{
  "dealId": "deal_123",
  "amount": 1000.00,
  "description": "Extra Payment",
  "type": "PAYMENT",
  "category": "loan_payment"
}
```

Statement Endpoints

Get Statements

GET /api/statements?year=2024&month=1
 Authorization: Bearer <token>

Query Parameters:

- `year` (optional): Filter by year
- `month` (optional): Filter by month (1-12)

Response:

```
{
  "success": true,
  "data": {
    "statements": [
      {
        "id": "stmt_123",
        "statementDate": "2024-01-31",
        "month": 1,
        "year": 2024,
        "totalCredits": 5000.00,
        "totalDebits": 2547.89,
        "netAmount": 2452.11,
        "fileUrl": "/api/statements/stmt_123/download"
      }
    ]
  }
}
```

Download Statement

```
GET /api/statements/[id]/download
Authorization: Bearer <token>
```

Response: PDF file download



Deal Endpoints

Get Deals

```
GET /api/deals?status=ACTIVE
Authorization: Bearer <token>
```

Query Parameters:

- `status` (optional): Filter by deal status (ACTIVE, PENDING, CLOSED, PAID_OFF)
- `productType` (optional): Filter by product type

Response:

```
{
  "success": true,
  "data": {
    "deals": [
      {
        "id": "deal_123",
        "dealNumber": "CF240001",
        "dealName": "Business Expansion Loan",
        "productType": "TERM_LOAN",
        "principalAmount": 100000,
        "remainingBalance": 85000,
        "interestRate": 6.25,
        "termMonths": 60,
        "monthlyPayment": 2547.89,
        "nextPaymentDate": "2024-02-15",
        "maturityDate": "2029-01-15",
        "status": "ACTIVE"
      }
    ]
  }
}
```

Get Deal Details

```
GET /api/deals/[id]
Authorization: Bearer <token>
```

Response:

```
{
  "success": true,
  "data": {
    "deal": {
      "id": "deal_123",
      "dealNumber": "CF240001",
      "dealName": "Business Expansion Loan",
      "productType": "TERM_LOAN",
      "principalAmount": 100000,
      "remainingBalance": 85000,
      "interestRate": 6.25,
      "termMonths": 60,
      "monthlyPayment": 2547.89,
      "nextPaymentDate": "2024-02-15",
      "maturityDate": "2029-01-15",
      "status": "ACTIVE",
      "paymentHistory": [
        {
          "date": "2024-01-15",
          "amount": 2547.89,
          "principal": 1464.56,
          "interest": 1083.33,
          "balance": 85000
        }
      ]
    }
  }
}
```

Application Endpoints

Submit Application

```
POST /api/applications
Authorization: Bearer <token>
Content-Type: application/json
```

```
{
  "productType": "TERM_LOAN",
  "requestedAmount": 100000,
  "purposeOfLoan": "Business expansion",
  "businessName": "Acme Corp",
  "businessType": "LLC",
  "yearsInBusiness": 5,
  "annualRevenue": 500000,
  "monthlyRevenue": 41667,
  "creditScore": 720,
  "preferredTerm": 60,
  "contactName": "John Doe",
  "contactEmail": "john@acme.com",
  "contactPhone": "+1-555-123-4567",
  "businessAddress": "123 Main St",
  "businessCity": "New York",
  "businessState": "NY",
  "businessZip": "10001"
}
```

Response:

```
{
  "success": true,
  "data": {
    "applicationId": "app_123",
    "applicationNumber": "APP240001",
    "status": "PENDING",
    "submissionDate": "2024-01-15T10:30:00Z",
    "message": "Application submitted successfully"
  }
}
```

Get Applications

```
GET /api/applications?status=PENDING
Authorization: Bearer <token>
```

Response:

```
{
  "success": true,
  "data": {
    "applications": [
      {
        "id": "app_123",
        "applicationNumber": "APP240001",
        "productType": "TERM_LOAN",
        "requestedAmount": 100000,
        "status": "PENDING",
        "submissionDate": "2024-01-15T10:30:00Z",
        "businessName": "Acme Corp"
      }
    ]
  }
}
```

Prepayment Endpoints

Calculate Prepayment

POST /api/prepayment/calculate
 Authorization: Bearer <token>
 Content-Type: application/json

```
{
  "dealId": "deal_123",
  "prepaymentAmount": 25000
}
```

Response:

```
{
  "success": true,
  "data": {
    "currentBalance": 85000,
    "prepaymentAmount": 25000,
    "newBalance": 60000,
    "interestSavings": 8750,
    "monthsReduced": 18,
    "newMaturityDate": "2027-07-15"
  }
}
```

Submit Prepayment Request

POST /api/prepayment/request
 Authorization: Bearer <token>
 Content-Type: application/json

```
{
  "dealId": "deal_123",
  "requestedAmount": 25000
}
```

Notification Endpoints

Get Notifications

```
GET /api/notifications?read=false
Authorization: Bearer <token>
```

Response:

```
{
  "success": true,
  "data": {
    "notifications": [
      {
        "id": "notif_123",
        "type": "PAYMENT_REMINDER",
        "priority": "HIGH",
        "title": "Payment Due Soon",
        "message": "Your payment of $2,547.89 is due on February 15, 2024",
        "isRead": false,
        "actionUrl": "/deals/deal_123",
        "createdAt": "2024-01-10T10:30:00Z"
      }
    ]
  }
}
```

Mark Notification as Read

```
PATCH /api/notifications/[id]
Authorization: Bearer <token>
Content-Type: application/json
```

```
{
  "isRead": true
}
```

Health Check

System Health

```
GET /api/health
```

Response:

```
{
  "status": "healthy",
  "timestamp": "2024-01-15T10:30:00Z",
  "database": "connected",
  "version": "1.0.0"
}
```


Analytics Endpoints

Get Analytics Data

```
GET /api/analytics?period=monthly&year=2024
Authorization: Bearer <token>
```

Response:

```
{
  "success": true,
  "data": {
    "monthlyPayments": [
      {
        "month": "January",
        "amount": 2547.89,
        "principal": 1464.56,
        "interest": 1083.33
      }
    ],
    "transactionSummary": {
      "totalPayments": 30574.68,
      "totalFees": 150.00,
      "transactionCount": 12
    }
  }
}
```

Error Handling

Error Response Format

```
{
  "success": false,
  "error": {
    "code": "VALIDATION_ERROR",
    "message": "Invalid request parameters",
    "details": [
      {
        "field": "email",
        "message": "Email is required"
      }
    ]
  }
}
```

Common Error Codes

- `VALIDATION_ERROR` (400): Invalid request data
- `UNAUTHORIZED` (401): Authentication required
- `FORBIDDEN` (403): Insufficient permissions
- `NOT_FOUND` (404): Resource not found
- `CONFLICT` (409): Resource already exists
- `RATE_LIMIT_EXCEEDED` (429): Too many requests

- `INTERNAL_SERVER_ERROR` (500): Server error



Rate Limiting

API endpoints are rate-limited to prevent abuse:

- **General endpoints:** 100 requests per 15 minutes
- **Authentication endpoints:** 5 requests per 15 minutes
- **File downloads:** 10 requests per minute

Rate limit headers:

```
X-RateLimit-Limit: 100
X-RateLimit-Remaining: 95
X-RateLimit-Reset: 1609459200
```



Security

API Security Features

- JWT-based authentication
- CSRF protection
- Input validation and sanitization
- SQL injection prevention
- XSS protection
- Rate limiting
- Secure headers

Best Practices

1. Always use HTTPS in production
2. Validate all input data
3. Use parameterized queries
4. Implement proper error handling
5. Log security events
6. Regular security audits

SDK and Examples

JavaScript/TypeScript Example

```
const apiClient = {
  baseUrl: 'https://api.capitalflow.com',

  async request(endpoint, options = {}) {
    const response = await fetch(`${this.baseUrl}${endpoint}`, {
      headers: {
        'Content-Type': 'application/json',
        'Authorization': `Bearer ${localStorage.getItem('token')}`,
        ...options.headers,
      },
      ...options,
    });

    return response.json();
  },

  async getDashboard() {
    return this.request('/api/dashboard');
  },

  async getTransactions(params = {}) {
    const query = new URLSearchParams(params);
    return this.request(`/api/transactions?${query}`);
  },
};
```

Usage Example

```
// Get dashboard data
const dashboard = await apiClient.getDashboard();

// Get recent transactions
const transactions = await apiClient.getTransactions({
  limit: 10,
  type: 'PAYMENT'
});

// Submit application
const application = await apiClient.request('/api/applications', {
  method: 'POST',
  body: JSON.stringify({
    productType: 'TERM_LOAN',
    requestedAmount: 100000,
    // ... other fields
  }),
});
```

Webhooks

Webhook Events

- application.submitted
- application.approved

- application.rejected
- payment.received
- payment.failed
- deal.created
- deal.updated

Webhook Payload Example

```
{
  "event": "payment.received",
  "timestamp": "2024-01-15T10:30:00Z",
  "data": {
    "transactionId": "txn_123",
    "dealId": "deal_123",
    "amount": 2547.89,
    "status": "COMPLETED"
  }
}
```

For more information, please refer to the [README.md](#) (../README.md) or contact support.