# **Guide Complet des API Mosaïque Capital**

## **📋 Table des Matières**

1. [Configuration de Base](https://claude.ai/chat/43d5993e-81fe-4c30-aa6e-c3ae6ecebc01#configuration-de-base)
2. [Authentification & Autorisation](https://claude.ai/chat/43d5993e-81fe-4c30-aa6e-c3ae6ecebc01#authentification--autorisation)
3. [Authentification à Deux Facteurs (MFA)](https://claude.ai/chat/43d5993e-81fe-4c30-aa6e-c3ae6ecebc01#authentification-%C3%A0-deux-facteurs-mfa)
4. [Gestion des Actifs](https://claude.ai/chat/43d5993e-81fe-4c30-aa6e-c3ae6ecebc01#gestion-des-actifs)
5. [Valorisations d'Actifs](https://claude.ai/chat/43d5993e-81fe-4c30-aa6e-c3ae6ecebc01#valorisations-dactifs)
6. [Portfolio & Patrimoine](https://claude.ai/chat/43d5993e-81fe-4c30-aa6e-c3ae6ecebc01#portfolio--patrimoine)
7. [Services Bancaires](https://claude.ai/chat/43d5993e-81fe-4c30-aa6e-c3ae6ecebc01#services-bancaires)
8. [Webhooks Bancaires](https://claude.ai/chat/43d5993e-81fe-4c30-aa6e-c3ae6ecebc01#webhooks-bancaires)
9. [Codes d'Erreur](https://claude.ai/chat/43d5993e-81fe-4c30-aa6e-c3ae6ecebc01#codes-derreur)

## **🔧 Configuration de Base**

### **URL de Base**

http://localhost:9999

### **Headers Communs**

const headers = {

'Content-Type': 'application/json',

'Authorization': 'Bearer ' + token, // Pour les endpoints authentifiés

'Accept': 'application/json'

};

## **🔐 Authentification & Autorisation**

### **1. Inscription d'un utilisateur**

POST /api/auth/signup

**Body:**

{

"username": "john\_doe",

"email": "john@example.com",

"password": "motDePasseSecurise123"

}

**Réponse (201):**

{

"message": "Utilisateur enregistré avec succès",

"username": "john\_doe",

"status": "created"

}

**Exemple JavaScript:**

async function signup(userData) {

const response = await fetch('/api/auth/signup', {

method: 'POST',

headers: { 'Content-Type': 'application/json' },

body: JSON.stringify(userData)

});

if (!response.ok) {

throw new Error('Erreur lors de l\'inscription');

}

return await response.json();

}

### **2. Connexion**

POST /api/auth/login

**Body:**

{

"username": "john\_doe",

"password": "motDePasseSecurise123",

"mfaCode": "123456" // Optionnel si MFA activé

}

**Réponse (200):**

{

"accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",

"refreshToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",

"tokenType": "Bearer",

"id": 1,

"username": "john\_doe",

"mfaEnabled": false

}

### **3. Rafraîchissement de token**

POST /api/auth/refresh

**Body:**

{

"refreshToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."

}

### **4. Déconnexion**

POST /api/auth/logout

**Headers:** Authorization: Bearer {token}

### **5. Informations utilisateur actuel**

GET /api/auth/me

**Headers:** Authorization: Bearer {token}

**Réponse (200):**

{

"id": 1,

"username": "john\_doe",

"email": "john@example.com",

"mfaEnabled": false,

"roles": ["ROLE\_USER"],

"active": true,

"createdAt": "2024-01-15T10:00:00",

"lastLoginAt": "2024-01-20T14:30:00"

}

## **🔒 Authentification à Deux Facteurs (MFA)**

### **1. Configuration MFA**

POST /api/mfa/setup

**Headers:** Authorization: Bearer {token}

**Réponse (200):**

{

"secret": "JBSWY3DPEHPK3PXP",

"qrCodeUrl": "otpauth://totp/MosaiqueCapital:john\_doe?secret=...",

"message": "Configuration MFA générée. Scannez le QR code...",

"instructions": "Utilisez Google Authenticator, Authy..."

}

### **2. Téléchargement QR Code**

GET /api/mfa/qrcode

**Headers:** Authorization: Bearer {token}

**Réponse:** Image PNG

### **3. Vérification et activation MFA**

POST /api/mfa/verify

**Headers:** Authorization: Bearer {token}

**Body:**

{

"code": "123456"

}

### **4. Statut MFA**

GET /api/mfa/status

**Headers:** Authorization: Bearer {token}

**Réponse (200):**

{

"enabled": true,

"message": "L'authentification à deux facteurs est activée"

}

### **5. Désactivation MFA**

POST /api/mfa/disable

**Headers:** Authorization: Bearer {token}

**Body:**

{

"currentMfaCode": "123456"

}

## **💰 Gestion des Actifs**

### **1. Liste de tous les actifs**

GET /api/assets

**Headers:** Authorization: Bearer {token}

**Réponse (200):**

[

{

"id": 1,

"name": "Résidence Principale",

"description": "Appartement 3 pièces Paris 15e",

"type": "REAL\_ESTATE\_RESIDENTIAL",

"currentValue": 450000.00,

"currency": "EUR",

"createdAt": "2024-01-15T10:00:00",

"updatedAt": "2024-01-20T14:30:00"

}

]

### **2. Récupération d'un actif par ID**

GET /api/assets/{id}

**Headers:** Authorization: Bearer {token}

### **3. Actifs par type**

GET /api/assets/type/{type}?includeSubTypes=true

**Paramètres:**

* type: Type d'actif (ex: REAL\_ESTATE, FINANCIAL, CASH)
* includeSubTypes: boolean (inclure les sous-types)

**Types disponibles:**

REAL\_ESTATE, REAL\_ESTATE\_RESIDENTIAL, REAL\_ESTATE\_RENTAL,

FINANCIAL, FINANCIAL\_STOCKS, FINANCIAL\_BONDS,

CASH, CASH\_CURRENT, CASH\_SAVINGS,

CRYPTOCURRENCY, OTHERS, LIABILITIES

### **4. Création d'un actif**

POST /api/assets

**Headers:** Authorization: Bearer {token}

**Body:**

{

"name": "Livret A",

"description": "Épargne sécurisée",

"type": "CASH\_SAVINGS",

"currentValue": 15000.00,

"currency": "EUR"

}

### **5. Mise à jour d'un actif**

PUT /api/assets/{id}

**Headers:** Authorization: Bearer {token}

**Body:** Même structure que création

### **6. Suppression d'un actif**

DELETE /api/assets/{id}

**Headers:** Authorization: Bearer {token}

## **📈 Valorisations d'Actifs**

### **1. Valorisations d'un actif**

GET /api/valuations/asset/{assetId}

**Headers:** Authorization: Bearer {token}

### **2. Valorisations par période**

GET /api/valuations/asset/{assetId}/range?startDate=2024-01-01&endDate=2024-01-31

**Headers:** Authorization: Bearer {token}

### **3. Création d'une valorisation**

POST /api/valuations

**Headers:** Authorization: Bearer {token}

**Body:**

{

"assetId": 1,

"valuationDate": "2024-01-20",

"value": 455000.00,

"currency": "EUR",

"source": "Estimation notaire"

}

### **4. Suppression d'une valorisation**

DELETE /api/valuations/{id}

**Headers:** Authorization: Bearer {token}

## **🏦 Portfolio & Patrimoine**

### **1. Résumé du portfolio**

GET /api/portfolio/summary

**Headers:** Authorization: Bearer {token}

**Réponse (200):**

{

"totalPatrimony": 675000.00

}

### **2. Distribution des actifs**

GET /api/portfolio/distribution

**Headers:** Authorization: Bearer {token}

**Réponse (200):**

[

{

"type": "REAL\_ESTATE",

"label": "Immobilier",

"total": 450000.00

},

{

"type": "CASH",

"label": "Liquidités",

"total": 225000.00

}

]

## **🏧 Services Bancaires**

### **1. Providers bancaires disponibles**

GET /api/banking/providers

**Headers:** Authorization: Bearer {token}

**Réponse (200):**

[

{

"code": "tink",

"name": "Tink (ex-Nordigen) - GRATUIT",

"logo": "/images/providers/tink.png",

"available": true,

"requiredFields": ["login", "password"],

"description": "🎉 GRATUIT : 100 connexions/mois ! 2500+ banques européennes.",

"strongAuthentication": true

}

]

### **2. Liste des connexions bancaires**

GET /api/banking/connections

**Headers:** Authorization: Bearer {token}

### **3. Initiation d'une connexion bancaire**

POST /api/banking/connections

**Headers:** Authorization: Bearer {token}

**Body:**

{

"provider": "tink",

"bankCredentials": {

"login": "user@banque.fr",

"password": "motdepasse",

"additionalField1": null,

"additionalField2": null,

"extraFields": {}

}

}

### **4. Confirmation d'une connexion**

POST /api/banking/connections/{id}/confirm

**Headers:** Authorization: Bearer {token}

**Body:**

{

"confirmationCode": "123456"

}

### **5. Synchronisation d'une connexion**

POST /api/banking/connections/{id}/sync

**Headers:** Authorization: Bearer {token}

**Réponse (200):**

{

"connectionId": 1,

"success": true,

"message": "Synchronisation réussie",

"accountsSynced": 2,

"transactionsSynced": 15,

"syncTimestamp": "2024-01-20T14:30:00",

"statistics": {

"newAccounts": 0,

"updatedAccounts": 2,

"newTransactions": 5,

"categorizedTransactions": 3

}

}

### **6. Santé d'une connexion**

GET /api/banking/connections/{id}/health

**Headers:** Authorization: Bearer {token}

### **7. Liste des comptes bancaires**

GET /api/banking/accounts

**Headers:** Authorization: Bearer {token}

**Réponse (200):**

[

{

"id": 1,

"accountId": "FR1420041010050500013M02606",

"name": "Compte Courant",

"type": "checking",

"balance": 2450.75,

"currency": "EUR",

"iban": "FR1420041010050500013M02606",

"lastSyncAt": "2024-01-20T14:30:00",

"createdAt": "2024-01-15T10:00:00",

"updatedAt": "2024-01-20T14:30:00"

}

]

### **8. Résumé financier global**

GET /api/banking/summary

**Headers:** Authorization: Bearer {token}

**Réponse (200):**

{

"totalBalance": 25450.75,

"totalAssets": 30000.00,

"totalLiabilities": 4549.25,

"totalAccounts": 3,

"balanceByType": {

"checking": 2450.75,

"savings": 25000.00,

"loan": -2000.00

},

"accountCountByType": {

"checking": 1,

"savings": 1,

"loan": 1

},

"monthlyIncome": 3500.00,

"monthlyExpenses": 2800.00,

"netWorth": 25450.75

}

### **9. Recherche de transactions**

POST /api/banking/transactions/search

**Headers:** Authorization: Bearer {token}

**Body:**

{

"accountId": 1,

"startDate": "2024-01-01",

"endDate": "2024-01-31",

"minAmount": -1000.00,

"maxAmount": 1000.00,

"category": "alimentation",

"type": "DEBIT",

"description": "MONOPRIX",

"page": 0,

"size": 50,

"sortBy": "transactionDate",

"sortDirection": "DESC"

}

### **10. Transactions d'un compte**

GET /api/banking/accounts/{accountId}/transactions?page=0&size=50&sortBy=transactionDate&sortDir=DESC

**Headers:** Authorization: Bearer {token}

### **11. Mise à jour catégorie d'une transaction**

PATCH /api/banking/transactions/{id}/category

**Headers:** Authorization: Bearer {token}

**Body:**

{

"category": "transport"

}

### **12. Statistiques par catégorie**

GET /api/banking/transactions/statistics/categories?startDate=2024-01-01&endDate=2024-01-31

**Headers:** Authorization: Bearer {token}

**Réponse (200):**

{

"period": {

"startDate": "2024-01-01",

"endDate": "2024-01-31"

},

"categoryExpenses": {

"alimentation": 456.78,

"transport": 123.45,

"loisirs": 89.12

},

"totalExpenses": 669.35,

"averageDailyExpense": 21.59

}

### **13. Statistiques de cash-flow**

GET /api/banking/transactions/statistics/cash-flow?startDate=2024-01-01&endDate=2024-01-31

**Headers:** Authorization: Bearer {token}

**Réponse (200):**

{

"period": {

"startDate": "2024-01-01",

"endDate": "2024-01-31"

},

"totalIncome": 3500.00,

"totalExpenses": 2800.00,

"netCashFlow": 700.00,

"savingsRate": 20.00

}

## **🔗 Webhooks Bancaires**

### **1. Statut des connexions Tink**

GET /api/banking/webhooks/tink/status/{connectionId}

**Réponse (200):**

{

"connectionId": "123",

"status": "ACTIVE",

"healthy": true,

"provider": "tink",

"message": "Connexion Tink active",

"timestamp": "2024-01-20T14:30:00"

}

### **2. URL d'autorisation Tink**

POST /api/banking/webhooks/tink/authorization-url

**Body:**

{

"requisitionId": "req\_123456",

"institutionId": "CREDIT\_AGRICOLE\_FR"

}

### **3. Test d'intégration Tink**

GET /api/banking/webhooks/tink/test

**Réponse (200):**

{

"provider": "tink",

"status": "AVAILABLE",

"pricing": "FREE - 100 connections/month",

"features": {

"accounts": true,

"transactions": true,

"balances": true,

"psd2\_native": true

},

"supported\_countries": ["FR", "DE", "ES", "IT"],

"institutions\_count": "2500+",

"message": "🎉 Tink est opérationnel et GRATUIT pour vos démos !",

"timestamp": "2024-01-20T14:30:00"

}

## **⚠️ Codes d'Erreur**

### **Codes HTTP Standards**

* **200**: Succès
* **201**: Créé avec succès
* **204**: Succès sans contenu
* **400**: Requête invalide
* **401**: Non authentifié
* **403**: Accès refusé
* **404**: Ressource non trouvée
* **409**: Conflit (ressource déjà existante)
* **422**: Entité non traitable (validation échouée)
* **500**: Erreur serveur interne

### **Format des Erreurs**

{

"status": 400,

"message": "Description de l'erreur",

"timestamp": "2024-01-20T14:30:00"

}

### **Erreurs de Validation**

{

"username": "Le nom d'utilisateur est obligatoire",

"email": "L'email doit être valide",

"password": "Le mot de passe doit contenir au moins 8 caractères"

}

## **🚀 Conseils d'Utilisation**

### **1. Sécurité**

* Toujours utiliser HTTPS en production
* Stocker les tokens de manière sécurisée
* Implémenter un système de refresh automatique
* Valider les données côté client ET serveur

### **2. Performance**

* Mettre en cache les données statiques (providers, types d'actifs)
* Implémenter la pagination pour les listes
* Utiliser des requêtes conditionnelles (If-Modified-Since)

### **3. UX/UI**

* Afficher des indicateurs de chargement
* Gérer les états d'erreur avec des messages clairs
* Implémenter un système de retry automatique
* Prévoir des fallbacks pour les connexions bancaires

### **4. Monitoring**

* Logger les erreurs côté client
* Monitorer les performances des API
* Tracker l'usage des fonctionnalités bancaires

Ce guide couvre l'intégralité des endpoints disponibles dans l'API Mosaïque Capital. Pour toute question spécifique ou besoin d'aide sur l'implémentation, n'hésitez pas à consulter le code source ou à demander des précisions.