



API Endpoints Summary - Forex Trading Platform



Endpoints Listos para Frontend



CURRENCIES MODULE - /currencies

typescript

```
// GET /currencies - Obtener todas las divisas con filtros
// GET /currencies/active - Obtener divisas activas
// GET /currencies/pairs - Obtener pares de divisas disponibles
// GET /currencies/code/:code - Obtener divisa por código
// GET /currencies/:id - Obtener divisa por ID
// POST /currencies - Crear nueva divisa
// PATCH /currencies/:id - Actualizar divisa
// PATCH /currencies/:id/activate - Activar divisa
// PATCH /currencies/:id/deactivate - Desactivar divisa
// DELETE /currencies/:id - Eliminar divisa
// POST /currencies/validate-pair - Validar par de divisas
```



CLIENTS MODULE - /clients

typescript

```
// GET /clients - Obtener todos los clientes con filtros y paginación
// GET /clients/active - Obtener clientes activos
// GET /clients/search?q=query - Buscar clientes
// GET /clients/stats - Estadísticas de clientes
// GET /clients/segment/:segmentId - Clientes por segmento
// GET /clients/identifier/:identifier - Cliente por identificador
// GET /clients/:id - Obtener cliente por ID
// POST /clients - Crear nuevo cliente
// PATCH /clients/:id - Actualizar cliente
// PATCH /clients/:id/activate - Activar cliente
// PATCH /clients/:id/deactivate - Desactivar cliente
// DELETE /clients/:id - Eliminar cliente
```



OPERATION FOLDERS MODULE - /operation-folders

typescript

```
// GET /operation-folders - Obtener todos los libros con filtros
// GET /operation-folders/active - Obtener libros activos
// GET /operation-folders/trading - Obtener libros de trading
// GET /operation-folders/sales - Obtener libros de sales
// GET /operation-folders/stats - Estadísticas de libros
// GET /operation-folders/code/:code - Libro por código
// GET /operation-folders/:id - Obtener libro por ID
// POST /operation-folders - Crear nuevo libro
// PATCH /operation-folders/:id - Actualizar libro
// PATCH /operation-folders/:id/activate - Activar libro
// PATCH /operation-folders/:id/deactivate - Desactivar libro
// DELETE /operation-folders/:id - Eliminar libro
```

FX OPERATIONS MODULE - /fx-operations

typescript

```
// GET /fx-operations - Obtener todas las operaciones con filtros y paginación
// GET /fx-operations/pending - Obtener operaciones pendientes
// GET /fx-operations/stats - Estadísticas de operaciones
// GET /fx-operations/volume-analysis?days=30 - Análisis de volumen
// GET /fx-operations/position-mx/:currency - Posición MX por divisa
// GET /fx-operations/by-currency/:currencyId - Operaciones por divisa
// GET /fx-operations/latest-by-currency/:currencyId - Últimas operaciones por divisa
// GET /fx-operations/:id - Obtener operación por ID
// POST /fx-operations - Crear nueva operación
// PATCH /fx-operations/:id - Actualizar operación
// PATCH /fx-operations/:id/confirm - Confirmar operación
// PATCH /fx-operations/:id/cancel - Cancelar operación
// POST /fx-operations/:id/calculate-pnl - Calcular PnL
// DELETE /fx-operations/:id - Eliminar operación
```

USER PROFILES MODULE - /user-profiles

typescript

```
// GET /user-profiles - Obtener todos los perfiles
// GET /user-profiles/:id - Obtener perfil por ID
// GET /user-profiles/:id/users - Usuarios por perfil
```

USERS MODULE - /users

typescript

```
// GET /users - Obtener todos los usuarios
// GET /users/active - Obtener usuarios activos
// GET /users/profile/:profileId - Usuarios por perfil
// GET /users/sales-group/:salesGroupId - Usuarios por grupo de ventas
// GET /users/:id - Obtener usuario por ID
// POST /users - Crear nuevo usuario
// PATCH /users/:id - Actualizar usuario
// PATCH /users/:id/activate - Activar usuario
// PATCH /users/:id/deactivate - Desactivar usuario
// DELETE /users/:id - Eliminar usuario
```

Mapeo Frontend ↔ Backend

Trading Module (Frontend)

typescript

```
// Posición MX
GET /fx-operations/position-mx/USD ← PosiciónMX.tsx
GET /fx-operations/latest-by-currency/1 ← Últimas operaciones

// Trading Spreads
GET /spreads/trading?currency=USD ← TradingSpreadManager.tsx
POST /spreads/trading ← Crear spread de trading
PATCH /spreads/trading/:id ← Actualizar spread

// Consulta Operaciones
GET /fx-operations?filters ← OperationsQuery.tsx
GET /fx-operations/:id ← Detalle de operación
```

Sales Module (Frontend)

typescript

// Precios (Sales)

GET /currencies/pairs ← PricesModule.tsx

POST /fx-operations ← Crear operación de cotización

GET /fx-operations/pending ← Operaciones pendientes

// Spreads Sales

GET /spreads/sales ← SalesSpreadManager.tsx

POST /spreads/sales ← Crear spread de ventas

// Spreads Excepción

GET /spreads/exceptions ← ExceptionSpreadManager.tsx

POST /spreads/exceptions ← Crear spread de excepción

Middle Office Module (Frontend)

typescript

// Clientes

GET /clients ← ClientsManager.tsx

POST /clients ← Crear cliente

PATCH /clients/:id ← Actualizar cliente

// Libros

GET /operation-folders ← BooksManager.tsx

POST /operation-folders ← Crear libro

// Divisas

GET /currencies ← CurrenciesManager.tsx

POST /currencies ← Crear divisa

// Usuarios

GET /users ← UsersManager.tsx

POST /users ← Crear usuario

Ejemplos de Uso desde Frontend

1. Hook para Currencies

typescript

```
// hooks/api/useCurrencies.ts
export function useCurrencies() {
  const { data: currencies, isLoading, error } = useQuery(
    ['currencies'],
    () => api.get('/currencies/active')
  );

  const createCurrency = useMutation(
    (data: CreateCurrencyDto) => api.post('/currencies', data)
  );

  return { currencies, isLoading, error, createCurrency };
}
```

2. Hook para Operations

typescript

```
// hooks/api/useOperations.ts
export function useOperations(filters?: QueryFxOperationSpotDto) {
  const { data, isLoading, error } = useQuery(
    ['operations', filters],
    () => api.get('/fx-operations', { params: filters })
  );

  const createOperation = useMutation(
    (data: CreateFxOperationSpotDto) => api.post('/fx-operations', data)
  );

  const confirmOperation = useMutation(
    (id: number) => api.patch(`/fx-operations/${id}/confirm`)
  );

  return {
    operations: data?.data,
    total: data?.total,
    isLoading,
    error,
    createOperation,
    confirmOperation
  };
}
```

3. Hook para Position MX

typescript

// hooks/api/usePositionMX.ts

```
export function usePositionMX(currency: string) {  
  const { data: position, isLoading, error } = useQuery(  
    ['position-mx', currency],  
    () => api.get(`/fx-operations/position-mx/${currency}`),  
    { enabled: !!currency }  
  );  
  
  return { position, isLoading, error };  
}
```

4. Cliente API Centralizado

typescript

```
// lib/api/client.ts
```

```
class ApiClient {  
  private baseUrl = process.env.NEXT_PUBLIC_API_URL || 'http://localhost:3001';  
  
  async request<T>(endpoint: string, options?: RequestInit): Promise<T> {  
    const token = localStorage.getItem('access_token');  
  
    const response = await fetch(`${this.baseUrl}${endpoint}`, {  
      headers: {  
        'Content-Type': 'application/json',  
        'Authorization': token ? `Bearer ${token}` : '',  
        ...options?.headers,  
      },  
      ...options,  
    });  
  
    if (!response.ok) {  
      throw new Error(`API Error: ${response.statusText}`);  
    }  
  
    return response.json();  
  }  
}
```

```
// Métodos específicos
```

```
currencies = {  
  getAll: (params?: any) => this.request('/currencies', { method: 'GET' }),  
  getActive: () => this.request('/currencies/active'),  
  getPairs: () => this.request('/currencies/pairs'),  
  create: (data: any) => this.request('/currencies', {  
    method: 'POST',  
    body: JSON.stringify(data)  
  }),  
};  
  
operations = {  
  getAll: (params?: any) => this.request(`/fx-operations?${new URLSearchParams(params)}`),  
  getPending: () => this.request('/fx-operations/pending'),  
  getPositionMX: (currency: string) => this.request(`/fx-operations/position-mx/${currency}`),  
  create: (data: any) => this.request('/fx-operations', {  
    method: 'POST',  
    body: JSON.stringify(data)  
  }),  
};
```

```
confirm: (id: number) => this.request(`/fx-operations/${id}/confirm`, { method: 'PATCH' }),  
cancel: (id: number,
```