

BankLink API – Endpoints Principales

Documentación de API

10 de noviembre de 2025

Índice

1	Healthcheck	2
2	Auth	2
2.1	Registrar usuario	2
2.2	Login	3
3	Clientes	3
4	Cuentas	4
5	Bancos Externos	4
6	Transferencias interbancarias	5
6.1	Enviar (outbound)	5
6.2	Recibir (inbound)	6
7	Modelo de error (común)	6
8	Ejemplos rápidos (cURL)	6
8.1	Crear cliente	6
8.2	Crear cuenta	7
8.3	Depósito	7
8.4	Registrar banco externo	7
8.5	Enviar transferencia	7

Información General

- **Base URL (local):** <http://localhost:5134>
- **Formato:** JSON (Content-Type: application/json)
- **Autenticación:** Authorization: Bearer <JWT> (cuando está habilitado)

1 Healthcheck

Verifica el estado de la API.

Método	URL	Descripción
GET	/health	Verifica que la API esté viva.

Ejemplo (200 OK)

```
{  
  "ok": true,  
  "at": "2025-11-03T13:03:54.7361445Z"  
}
```

2 Auth

Gestión de autenticación y registro de usuarios.

2.1 Registrar usuario

Método	URL	Body (requerido)
POST	/api/Auth/register	{ email, fullName, password }

Request

```
{  
  "email": "fede@banklink.local",  
  "fullName": "Federico Calveira",  
  "password": "Admin123!"  
}
```

Response (201 Created)

Headers: Location: /api/auth/me

```
{  
  "accessToken": "<jwt>",  
  "expiresAtUtc": "2025-11-03T15:16:28.3876671Z"  
}
```

Método	URL	Body (requerido)
POST	/api/Auth/login	{ email, password }

2.2 Login

Request

```
{
  "email": "fede@banklink.local",
  "password": "Admin123!"
}
```

Response (200 OK)

```
{
  "accessToken": "<jwt>",
  "expiresAtUtc": "2025-11-03T15:17:13.4480429Z"
}
```

3 Clientes

Operaciones CRUD para la gestión de clientes.

Método	URL	Descripción
GET	/api/Clientes	Lista de clientes.
GET	/api/Clientes/{id}	Cliente por id.
POST	/api/Clientes	Crear cliente.
PUT	/api/Clientes/{id}	Actualizar cliente.
DELETE	/api/Clientes/{id}	Eliminar cliente.

Body (POST / PUT)

```
{
  "nombre": "Ana",
  "apellido": "García",
  "dni": "39112222",
  "direccion": "Av. Siempreviva 123",
  "telefono": "1123456789",
  "email": "ana@example.com"
}
```

Response (200/201)

```
{
  "id": 1,
  "nombre": "Ana",
  "apellido": "García",
  "dni": "39112222",
  "direccion": "Av. Siempreviva 123",
  "telefono": "1123456789",
  "email": "ana@example.com"
}
```

Errores frecuentes: 400 (validación), 404 (no encontrado), 409 (DNI duplicado).

4 Cuentas

Gestión de cuentas bancarias, saldos y movimientos.

Método	URL	Descripción
GET	/api/Cuentas	Lista de cuentas.
GET	/api/Cuentas/{id}	Cuenta por id.
POST	/api/Cuentas	Crear cuenta.
PUT	/api/Cuentas/{id}/estado	Cambiar estado (activa/inactiva).
GET	/api/Cuentas/{id}/movimientos	Movimientos de la cuenta.
POST	/api/Cuentas/{id}/depositos	Depósito (acredita saldo + mov).
POST	/api/Cuentas/{id}/retiros	Retiro (debita saldo + mov).

Body – Crear cuenta (POST)

```
{  
    "accountNumber": "001-000001",  
    "type": 1,  
    "clientId": 1,  
    "initialBalance": 5000  
}
```

Response (201)

```
{  
    "id": 1,  
    "accountNumber": "001-000001",  
    "type": 1,  
    "clientId": 1,  
    "balance": 5000,  
    "status": 1  
}
```

Body – Actualizar estado (PUT) { "status": 2 } (1=Activa, 2=Inactiva)

Body – Depósito / Retiro

```
{  
    "monto": 1500,  
    "descripcion": "Sueldo"  
}
```

Response Depósito/Retiro (201)

Location: /api/Cuentas/{id}/movimientos

```
{  
    "saldo": 7100  
}
```

Errores frecuentes: 400 (validación), 404 (cuenta no existe), 422 (saldo insuficiente), 409 (número de cuenta duplicado).

5 Bancos Externos

Configuración de otras entidades bancarias.

Método	URL	Descripción
GET	/api/BancosExternos	Lista de bancos externos.
GET	/api/BancosExternos/{id}	Banco por id.
POST	/api/BancosExternos	Registrar banco externo.
PUT	/api/BancosExternos/{id}	Actualizar banco externo.
DELETE	/api/BancosExternos/{id}	Eliminar banco externo.

Body (POST / PUT)

```
{
  "name": "GrupoB",
  "code": "GRPB",
  "baseUrl": "http://localhost:7071",
  "transferEndpoint": "/api/Transferencias/recibir",
  "apiKey": "SECRETO_DEL_OTRO_BANCO"
}
```

Response (200/201)

```
{
  "id": 1,
  "name": "GrupoB",
  "code": "GRPB",
  "baseUrl": "http://localhost:7071",
  "transferEndpoint": "/api/Transferencias/recibir",
  "apiKey": ""
}
```

Errores frecuentes: 400 (validación), 404, 409 (código ya registrado).

6 Transferencias interbancarias

Endpoints para enviar y recibir dinero de otros bancos.

6.1 Enviar (outbound)

Debita en BankLink e invoca /recibir del banco externo.

Idempotencia: idempotencyKey debe ser único por transferencia.

Método	URL	Descripción
POST	/api/Transferencias/enviar	Inicia una transferencia saliente.

Body (requerido)

```
{
  "originAccountId": 1,
  "destinationBankCode": "GRPB",
  "destinationAccountNumber": "002-999999",
  "amount": 1500.00,
  "concept": "Pago alquiler",
  "idempotencyKey": "b9a50cba-7a99-4fda-9af4-2f7a2b4a2f12"
}
```

Response (201)

Location: /api/Cuentas/{originAccountId}/movimientos

```
{  
  "balance": 5600.00,  
  "movementId": 123  
}
```

Errores frecuentes: 404 (cuenta origen o banco destino no existe), 422 (saldo insuficiente), 409 (idempotencyKey ya usado), 502 (error llamando al banco externo).

6.2 Recibir (inbound)

Acredita en cuenta destino local.

Método	URL	Header requerido
POST	/api/Transferencias/recibir	X-Api-Key: <API_KEY_DEL_ORIGEN>

Body

```
{  
  "originBankCode": "GRPB",  
  "originAccountNumber": "002-999999",  
  "destinationAccountNumber": "001-000001",  
  "amount": 1500.00,  
  "concept": "Pago alquiler",  
  "idempotencyKey": "b9a50cba-7a99-4fda-9af4-2f7a2b4a2f12"  
}
```

Response (201)

Location: /api/Cuentas/{destAccountId}/movimientos

```
{  
  "balance": 7100.00,  
  "movementId": 456  
}
```

Errores frecuentes: 401 (API key inválida), 404 (cuenta destino inexistente/inactiva o banco origen no registrado), 409 (idempotencyKey ya acreditado).

7 Modelo de error (común)

```
{  
  "message": "Detalle del error"  
}
```

8 Ejemplos rápidos (cURL)

8.1 Crear cliente

```
curl -X POST http://localhost:5134/api/Clientes \
-H "Content-Type: application/json" \
-d '{"nombre":"Ana","apellido":"García","dni":"39112222",
"direccion":"Av. SiempreViva 123","telefono":"1123456789",
"email":"ana@example.com"}'
```

8.2 Crear cuenta

```
curl -X POST http://localhost:5134/api/Cuentas \
-H "Content-Type: application/json" \
-d '{"accountNumber":"001-000001","type":1,"clientId":1,
"initialBalance":5000}'
```

8.3 Depósito

```
curl -X POST http://localhost:5134/api/Cuentas/1/depositos \
-H "Content-Type: application/json" \
-d '{"monto":1500,"descripcion":"Sueldo"}'
```

8.4 Registrar banco externo

```
curl -X POST http://localhost:5134/api/BancosExternos \
-H "Content-Type: application/json" \
-d '{"name":"GrupoB","code":"GRPB","baseUrl":"http://localhost:7071",
"transferEndpoint":"/api/Transferencias/recibir","apiKey":"SECRETO"}'
```

8.5 Enviar transferencia

```
curl -X POST http://localhost:5134/api/Transferencias/enviar \
-H "Content-Type: application/json" \
-d '{"originAccountId":1,"destinationBankCode":"GRPB",
"destinationAccountNumber":"002-999999","amount":1500,
"concept":"Pago alquiler",
"idempotencyKey":"c9b3b1a2-6d7a-4e2b-9c2f-11a222b3cd44"}'
```
