

Finans Public API

Documentatia publica pentru dezvoltatorii care vor sa construiasca integrari cu Finans. API-ul este orientat pe integrari generale pentru ecommerce, ERP, CRM, aplicatii interne si automatizari, cu Bearer API keys, OAuth access tokens, idempotency si endpoint-uri pentru documente fiscale.

Introduction

Prefixul de baza pentru toate endpoint-urile publice este `/api/public/v1`. Starea curenta a API-ului este suficienta pentru integrari generale care creeaza clienti, emit facturi, inregistreaza plati si descarca PDF sau XML.

Resursele disponibile acum sunt:

- Health
 - Company
 - Document series
 - Bank accounts
 - Clients
 - Products
 - Invoices
 - Payments
 - Webhooks
-

Base URL

API base URL:

```
https://app.finans.ro/api/public/v1
```

Toate exemplele din aceasta pagina folosesc acest prefix. Pentru medii interne sau de test, prefixul poate fi acelasi API public expus sub alt domeniu, dar structura de endpoint-uri ramane identica.

Authentication

API-ul foloseste chei generate din produs si trimise prin header-ul standard:

```
Authorization: Bearer fk_live_xxxxxxxxx
```

Pentru aplicatiile conectate poti folosi si token-uri OAuth:

```
Authorization: Bearer foat_xxxxxxxxx
```

Identificatorii publici folosesc prefixele:

- fk_live_...
- foat_...

Flow-ul OAuth disponibil in produs foloseste endpoint-urile:

- GET /api/public/v1/oauth/authorize
- POST /api/public/v1/oauth/authorize/decision
- POST /api/public/v1/oauth/token

Pentru verificare rapida foloseste GET /health/verify. Acesta intoarce date despre firma, conexiunea de integrare si scope-urile disponibile pe cheia curenta.

Recommended scopes

Pentru o integrare completa recomandam o cheie API dedicata, cu atat scope-uri de citire cat si de scriere.

company.read

document_series.read

bank_accounts.read

clients.read

clients.write

products.read

products.write

invoices.read

invoices.write

payments.read

payments.write

documents.read

spv.read

webhooks.read

webhooks.write

Daca primesti raspuns de tip `missing_scopes`, cheia este valida, dar nu are permisiunea ceruta de endpoint-ul apelat.

Idempotency

Pentru a evita duplicatele la retry-uri HTTP, API-ul implementeaza idempotency persistenta pentru:

- `POST /invoices`
- `POST /payments`

Trimite un header stabil pe actiunea business, nu pe fiecare request individual:

```
Idempotency-Key: external-system:order_100045:create_invoice
```

Daca primesti `409 idempotency_in_progress`, nu genera un key nou si nu schimba payload-ul. Refa acelasi request cu acelasi `Idempotency-Key` dupa un backoff scurt pana cand primesti replay-ul raspunsului final.

In varianta curenta nu exista endpoint separat de status pentru idempotency. Recovery-ul oficial este polling prin retrimiteria aceluiasi request idempotent.

Health

GET

```
/api/public/v1/health/verify
```

fara scope suplimentar

Foloseste acest endpoint ca prim pas in onboarding. Daca el raspunde corect, cheia este valida si poti vedea exact ce scope-uri ai disponibile.

Company

GET

```
/api/public/v1/company/profile
```

company.read

Returneaza datele firmei emitente folosite in documente: identificare fiscala, adresa, contact, TVA, moneda implicita si setarile generale ale firmei.

Document series

GET

```
/api/public/v1/document-series
```

document_series.read

Poti descoperi seriile active pentru documente si le poti filtra cu `?type=INVOICE`, `?type=PROFORMA` sau `?type=RECEIPT`.

La emitere poti trimite `series` sau `series_id`. Daca exista o singura serie activa sau una marcata implicit, API-ul o poate selecta automat.

Bank accounts

GET

/api/public/v1/bank-accounts

bank_accounts.read

Returneaza conturile bancare active ale firmei. Poti filtra cu ?currency=EUR sau cu alta moneda suportata pentru a alege contul corect la documentele in valuta.

Clients

GET

/api/public/v1/clients

clients.read

POST

/api/public/v1/clients

clients.write

GET

/api/public/v1/clients/{clientId}

clients.read

PATCH

/api/public/v1/clients/{clientId}

clients.write

Resource-ul este util pentru sincronizare clienti din ecommerce, ERP sau CRM.

Filtrarea pe lista se face cu parametrul search, care cauta in nume, CUI si email.

POST /api/public/v1/clients

Authorization: Bearer fk_live_xxxxxxxxx

Content-Type: application/json

```
{
  "type": "COMPANY",
  "external_id": "customer_123",
  "name": "Client Test SRL",
  "cui": "R012345678",
  "reg_com": "J40/123/2024",
  "address": "Str. Exemplu 1",
  "city": "Bucuresti",
  "county": "Bucuresti",
  "country": "RO",
  "postal_code": "010101",
  "email": "client@example.com",
  "phone": "+40722111222",
  "iban": "R049AAAA1B31007593840000",
  "bank_name": "Banca Demo",
  "vat_payer": true,
  "vat_number": "R012345678"
}
```

GET /api/public/v1/clients?search=Client%20Test&page=1&page_size=20
Authorization: Bearer fk_live_xxxxxxxxx

Products

GET

/api/public/v1/products

products.read

POST

/api/public/v1/products

products.write

GET

/api/public/v1/products/{productId}

products.read

PATCH

```
/api/public/v1/products/{productId}
```

```
products.write
```

Poate fi folosit pentru mapare SKU, sincronizare catalog si creare de servicii standardizate.

Filtrarea pe lista se face cu parametrul q pentru nume si cu currency pentru moneda.

```
POST /api/public/v1/products
```

```
Authorization: Bearer fk_live_xxxxxxxxx
```

```
Content-Type: application/json
```

```
{
  "name": "Produs demo",
  "unit": "buc",
  "currency": "RON",
  "unit_price": 100,
  "vat_rate": 19,
  "price_includes_vat": false,
  "product_code": "SKU-ABC"
}
```

```
GET /api/public/v1/products?q=Produs%20demo&currency=RON&limit=20
```

```
Authorization: Bearer fk_live_xxxxxxxxx
```

Invoices

GET

```
/api/public/v1/invoices
```

```
invoices.read
```

POST

```
/api/public/v1/invoices
```

```
invoices.write
```

GET

/api/public/v1/invoices/{invoiceId}

invoices.read

PATCH

/api/public/v1/invoices/{invoiceId}

invoices.write

GET

/api/public/v1/invoices/{invoiceId}/pdf

documents.read

GET

/api/public/v1/invoices/{invoiceId}/xml

documents.read

GET

/api/public/v1/invoices/{invoiceId}/spv-status

spv.read

O factura poate fi emisa direct dintr-un sistem extern folosind un payload orientat pe integrari, cu client, source si lines.

Daca trimiti series_id, API-ul foloseste exact seria indicata. Daca omiti seria, API-ul incearca sa aleaga seria activa implicita pentru tipul documentului.

In fiecare linie, external_product_id este optional si este tratat ca alias pentru product_code. Valoarea se mapeaza la codul de produs folosit in liniile facturii si in catalogul intern, nu la un external_id separat.

Produsul nu trebuie sa existe deja in /products. Daca exista un produs activ cu acelasi cod, este reutilizat; daca nu exista, factura poate fi emisa oricum, iar catalogul este sincronizat automat din liniile documentului.

Payments

GET

/api/public/v1/payments

payments.read

POST

/api/public/v1/payments

payments.write

GET

/api/public/v1/payments/{paymentId}

payments.read

PATCH

/api/public/v1/payments/{paymentId}

payments.write

GET

/api/public/v1/payments/{paymentId}/pdf

documents.read

Dupa emiterea facturii poti inregistra plata si, pentru platile cash, poti descarca PDF-ul aferent chitantei.

Pentru platile cash poti trimite `receipt_series` sau `receipt_series_id`. Daca nu sunt trimise si exista o singura serie activa sau una implicita, API-ul o poate selecta automat.

Webhooks

GET

/api/public/v1/webhooks

webhooks.read

POST

```
/api/public/v1/webhooks
```

```
webhooks.write
```

PATCH

```
/api/public/v1/webhooks/{webhookId}
```

```
webhooks.write
```

POST

```
/api/public/v1/webhooks/{webhookId}/test
```

```
webhooks.write
```

Endpoint-urile actuale acopera configurarea, update-ul si testarea webhook-urilor. Livrarea initiala se face imediat la emiterea evenimentului, iar retry-urile automate sunt orchestrate de workerul dedicat care proceseaza coada protejata din backend.

URL-urile de webhook trebuie sa fie publice, pe `https://`, fara host-uri locale, private sau rezervate.

Daca serverul este configurat cu `PUBLIC_API_WEBHOOK_ALLOWED_HOSTS`, host-ul endpoint-ului trebuie sa intre si in allowlist-ul explicit permis.

La creare primesti o singura data `plain_secret`. Pastreaza-l securizat si verifica fiecare livrare folosind `x-finans-signature`: semnatura este HMAC-SHA256, in format hex, calculata peste sirul `<x-finans-timestamp>.<raw_body>`. Compara semnaturile in timp constant si respinge timestamp-urile mai vechi de 5 minute pentru protectie anti-replay.

Headerele trimise sunt `x-finans-signature`, `x-finans-timestamp`, `x-finans-event`, `x-finans-event-id` si `x-finans-delivery-id`. Verificarea trebuie facuta pe corpul brut, inainte de parsarea JSON.

Create invoice example

```
POST /api/public/v1/invoices
```

```
Authorization: Bearer fk_live_xxxxxxxxx
```

Idempotency-Key: external-system:order_100045:create_invoice
Content-Type: application/json

```
{
  "series": "FCT",
  "type": "INVOICE",
  "issue_date": "2026-04-19",
  "currency": "RON",
  "client": {
    "external_id": "customer_123",
    "email": "client@example.com",
    "name": "Client Test SRL",
    "cui": "R012345678",
    "address": "Str. Exemplu 1",
    "city": "Bucuresti",
    "country": "RO"
  },
  "source": {
    "system": "external_app",
    "connection_id": "ic_123",
    "external_order_id": "100045",
    "external_order_number": "#100045"
  },
  "lines": [
    {
      "external_product_id": "sku_abc",
      "description": "Produs demo",
      "quantity": 1,
      "unit_price": 100,
      "vat_rate": 19
    }
  ]
}
```

Create payment example

```
{
  "invoice_id": "inv_123",
  "amount": 119,
  "currency": "RON",
  "payment_date": "2026-04-19",
  "payment_method": "CARD",
```

```
"reference": "order-100045",  
"source": {  
  "connection_id": "ic_123",  
  "external_order_id": "100045",  
  "external_payment_id": "pay_100045"  
}  
}
```

Pentru plati cash poti trimite si `receipt_series`, apoi folosi endpoint-ul PDF pentru chitanta.