

API Integration Details

Overview

FiredUp integrates with Tink using the Connectivity API v1. This document details all API endpoints used, OAuth scopes requested, and implementation specifics.

Tink API Version

Component	Version
API	Connectivity v1
Tink Link	v1
Base URL	https://api.tink.com
Tink Link URL	https://link.tink.com

OAuth Scopes

Client Token Scopes

Scope	Purpose	Usage
user:create	Create Tink users	One-time per user
authorization:grant	Generate auth codes	Each bank connection

User Token Scopes

Scope	Purpose	Required
accounts:read	Read account list	Yes
transactions:read	Read transactions	Yes

Scope	Purpose	Required
credentials:read	Read credential status	Yes
credentials:write	Initial credential setup	Yes
balances:read	Read account balances	Yes

Scopes NOT Requested

Scope	Reason Not Needed
payment:write	No payment initiation
payment:read	No payment status checks
beneficiaries:*	No beneficiary management
user:delete	Users deleted via own system

API Endpoints Used

Authentication Endpoints

1. Get Client Access Token

POST https://api.tink.com/api/v1/oauth/token
 Content-Type: application/x-www-form-urlencoded

```
client_id={CLIENT_ID}
client_secret={CLIENT_SECRET}
grant_type=client_credentials
scope=user:create,authorization:grant
```

Response:

```
{
  "access_token": "CLIENT_TOKEN",
  "token_type": "bearer",
  "expires_in": 1800,
```

```
"scope": "user:create,authorization:grant"
}
```

Implementation: tink_service.py:get_client_access_token()

2. Create User

```
POST https://api.tink.com/api/v1/user/create
Authorization: Bearer {CLIENT_TOKEN}
Content-Type: application/json

{
  "external_user_id": "hb2_{sha256_hash}",
  "market": "PL",
  "locale": "pl_PL"
}
```

Response:

```
{
  "user_id": "tink_user_id_here",
  "external_user_id": "hb2_{sha256_hash}"
}
```

Note: 409 Conflict is expected if user already exists.

Implementation: tink_service.py:create_user()

3. Generate Delegated Authorization Code

```
POST https://api.tink.com/api/v1/oauth/authorization-grant/delegate
Authorization: Bearer {CLIENT_TOKEN}
Content-Type: application/x-www-form-urlencoded
```

```
response_type=code
external_user_id=hb2_{sha256_hash}
actor_client_id=df05e4b379934cd09963197cc855bfe9
scope=accounts:read,transactions:read,credentials:write,credentials:read,balance:read
```

Response:

```
{
  "code": "authorization_code_here"
}
```

Note: `actor_client_id=df05e4b379934cd09963197cc855bfe9` is Tink Link's official client ID.

Implementation: `tink_service.py:get_authorization_code()`

4. Exchange Code for User Token

```
POST https://api.tink.com/api/v1/oauth/token
Content-Type: application/x-www-form-urlencoded

client_id={CLIENT_ID}
client_secret={CLIENT_SECRET}
code={AUTHORIZATION_CODE}
grant_type=authorization_code
```

Response:

```
{
  "access_token": "USER_ACCESS_TOKEN",
  "token_type": "bearer",
  "expires_in": 7200,
  "refresh_token": "USER_REFRESH_TOKEN",
  "scope": "accounts:read,transactions:read,..."
}
```

Implementation: `tink_service.py:exchange_code_for_tokens()`

5. Refresh Access Token

```
POST https://api.tink.com/api/v1/oauth/token
Content-Type: application/x-www-form-urlencoded

client_id={CLIENT_ID}
```

```
client_secret={CLIENT_SECRET}
refresh_token={REFRESH_TOKEN}
grant_type=refresh_token
```

Implementation: tink_service.py:refresh_access_token()

Data Endpoints

6. List Accounts

```
GET https://api.tink.com/data/v2/accounts
Authorization: Bearer {USER_ACCESS_TOKEN}
```

Response:

```
{
  "accounts": [
    {
      "id": "account_id",
      "name": "Main Account",
      "type": "CHECKING",
      "balances": {
        "available": {
          "amount": { "value": { "unscaledValue": "150000",
            "scale": "2" }, "currencyCode": "PLN" }
        }
      },
      "identifiers": {
        "iban": { "iban": "PL12345678901234567890123456" }
      }
    }
  ]
}
```

Implementation: tink_service.py:get_accounts()

7. List Transactions

```
GET https://api.tink.com/data/v2/transactions
Authorization: Bearer {USER_ACCESS_TOKEN}
```

Query Parameters: | Parameter | Value | Description | |-----|-----|-----| | `pageSize` | 100 | Max transactions per page | | `pageToken` | (from response) | Pagination cursor |

Response:

```
{
  "transactions": [
    {
      "id": "txn_id",
      "accountId": "account_id",
      "amount": {
        "value": { "unscaledValue": "-5000", "scale": "2" },
        "currencyCode": "PLN"
      },
      "descriptions": {
        "display": "ZABKA WARSZAWA",
        "original": "ZABKA SP Z O O WARSZAWA PL"
      },
      "dates": {
        "booked": "2026-01-15"
      },
      "categories": {
        "pfm": {
          "id": "expenses:food.groceries",
          "name": "Groceries"
        }
      },
      "status": "BOOKED"
    }
  ],
  "nextPageToken": "token_for_next_page"
}
```

Implementation: `tink_service.py:get_transactions()`

Tink Link

Build Tink Link URL

```
https://link.tink.com/1.0/transactions/connect-accounts
?client_id={CLIENT_ID}
&redirect_uri={REDIRECT_URI}
&state={STATE_TOKEN}
&authorization_code={AUTH_CODE}
&market=PL
&locale=pl_PL
```

Parameters:

Parameter	Description
<code>client_id</code>	FiredUp's Tink client ID
<code>redirect_uri</code>	<code>https://firedup.app/banking/tink/callback</code>
<code>state</code>	HMAC-signed state token for CSRF protection
<code>authorization_code</code>	Code from step 3
<code>market</code>	<code>PL</code> (Poland)
<code>locale</code>	<code>pl_PL</code> (Polish language)

Implementation: `tink_service.py:build_tink_link_url()`

FiredUp API Endpoints

Backend Routes for Tink

Endpoint	Method	Description
<code>/banking/tink/connect</code>	POST	Initiate bank connection
<code>/banking/tink/callback</code>	POST	Handle OAuth callback
<code>/banking/tink/connections</code>	GET	List user's connections

Endpoint	Method	Description
/banking/tink/connections/{id}	DELETE	Disconnect bank
/banking/tink/test	GET	Test/debug endpoint

Backend Routes for Transactions

Endpoint	Method	Description
/banking/transactions/sync	POST	Sync transactions from Tink
/banking/transactions	GET	List imported transactions
/banking/transactions/pending	GET	List unreviewed transactions
/banking/transactions/stats	GET	Transaction statistics
/banking/transactions/{id}/convert	POST	Convert to expense/income
/banking/transactions/{id}/accept	POST	Mark as reviewed
/banking/transactions/{id}/reject	POST	Reject transaction
/banking/transactions/bulk/accept	POST	Bulk accept
/banking/transactions/bulk/reject	POST	Bulk reject
/banking/transactions/bulk/convert	POST	Bulk convert

Error Handling

Tink API Errors

Status	Meaning	Action
400	Bad Request	Log, fail immediately
401	Unauthorized	Refresh token, retry once
403	Forbidden	Log, fail immediately
404	Not Found	Log, fail immediately
409	Conflict (user exists)	Proceed (expected)
422	Validation Error	Log, fail immediately
429	Rate Limited	Retry with Retry-After header
500-504	Server Error	Retry with exponential backoff

Custom Exceptions

```
class TinkAPIError(Exception):
    """Base exception for Tink API errors."""
    status_code: int
    endpoint: str
    response_body: str

class TinkAPIRetryExhausted(TinkAPIError):
    """All retry attempts exhausted."""
    attempts: int
```

Rate Limiting

Tink Rate Limits (Sandbox)

Limit	Value
Requests/minute	100
Requests/day	1,000

FiredUp Implementation

- Respects `Retry-After` header
 - Exponential backoff (1s base, 30s max)
 - Max 3 retry attempts
 - Jitter $\pm 25\%$ to prevent thundering herd
-

Request Examples

Complete Connection Flow

```
# Step 1: Get client token
client_token = await tink_service.get_client_access_token()

# Step 2: Create Tink user
external_id = sanitize_external_user_id(user.email)
tink_user = await tink_service.create_user(client_token,
                                           external_id)

# Step 3: Get authorization code
auth_code = await tink_service.get_authorization_code(
    client_token, external_id, scopes
)

# Step 4: Store pending auth
state = generate_state_token()
store_pending_auth(state, user.id, tink_user.id, auth_code)

# Step 5: Build Tink Link URL
```

```
url = build_tink_link_url(auth_code, state, redirect_uri)

# --- User completes flow in Tink Link ---

# Step 6: Handle callback
pending = verify_and_get_pending_auth(state)
tokens = await
    tink_service.exchange_code_for_tokens(pending.auth_code)

# Step 7: Store connection
connection = TinkConnection(
    user_id=user.id,
    tink_user_id=pending.tink_user_id,
    access_token=tokens.access_token,
    refresh_token=tokens.refresh_token,
    is_active=True
)
db.add(connection)
```

Implementation Files

File	Purpose
backend/app/services/ tink_service.py	Main Tink service (~1,200 lines)
backend/app/services/ tink_metrics_service.py	Request metrics
backend/app/services/ audit_service.py	Audit logging
backend/app/routers/tink.py	Tink API routes
backend/app/routers/ bank_transactions.py	Transaction routes
backend/app/models.py	Database models

Environment Configuration

Required Variables

Variable	Description
TINK_CLIENT_ID	Tink application client ID
TINK_CLIENT_SECRET	Tink application secret
TINK_REDIRECT_URI	OAuth callback URL

Production Values

Variable	Production Value
TINK_REDIRECT_URI	https://firedup.app/banking/tink/callback

Document Revision

Version	Date	Changes
1.0	February 2026	Initial version