FGAPI 定義書

rev 0.5

本書について

本書で定義する API について

本書では FGAPI で実装すべき「**API の仕様**」を定義します。
仕様に準拠する API をすべて実装した状態を「**FGAPI 準拠**」とします。

なお本書では、具体的な API の実現方法（プログラミング言語、動作環境等）は指定しません。実現方法は API の提供者にて適切に決定します。

API の拡張について

API の拡張は、FGAPI の仕様に準拠する限り、任意に実施可能です。
逆に、FGAPI 準拠から除外されるのは、以下のいずれかに該当する場合です。

- ✓ リクエストのクエリ・POST パラメータへの、必須のフィールドの追加
- ✓ リクエストのクエリ・POST パラメータからの、必須もしくは任意のフィールドの削除
- ✓ レスポンスからの、フィールドの削除
- ✓ API のパスの変更 （※1）

※1）特例として、パスの前方にディレクトリを挿入することが可能です。ただし、挿入するディレクトリは、すべての API で共通とする必要があります。
例えば以下のように、すべての API に対して共通の「**/api/v1**」の挿入が可能です。

標準のパス	挿入後のパス
/auth/login	**/api/v1**/auth/login
/auth/token	**/api/v1**/auth/token
/accounts/list	**/api/v1**/accounts/list
/xyz	**/api/v1**/xyz

用語の定義

API 使用者の種類

API 使用者として、以下の 2 種類を定義します。

- ✓ クライアント（省略時、CLI）：Fintech アプリケーションおよびサーバ
- ✓ エンドユーザー（省略時、EUS）：クライアントの利用者
  ※ 前提として、エンドユーザーは「FGAPI を提供する銀行の口座および FGAPI の接続先となる既存システム（インターネットバンキング等）のアカウントを保有」しており、「既存システムを利用可能」な状態であるものとします。

API の実装規則

基本的な規則

RESTful API として実装します。

リクエストパラメータに関する規則

HTTP メソッドに応じて、パラメータを以下のように指定します

- ✓ GET：クエリパラメータで指定
- ✓ POST/PUT/DELETE：リクエストボディに「application/x-www-form-urlencoded」形式で指定

文字コードは UTF-8 とします。

レスポンスに関する規則

実行結果の通知

実行結果の成功・失敗は HTTP ステータスコードで以下のように表現します。

- ✓ 200：[成功]
- ✓ 400：[失敗] パラメータが不正
- ✓ 401：[失敗] アクセストークン等の認証情報が不正
- ✓ 403：[失敗] API 呼び出しに必要な権限が不足している
- ✓ 404：[失敗] API が未定義

その他のサーバエラーについては、500 番台の HTTP ステータスコードを使用します。

レスポンスデータの形式

JSON 形式のテキストデータが、HTTP レスポンスボディとして返却されます。
JSON データの内容は、各 API の仕様に従います。

文字コードは UTF-8 とします。
JSON データのキーおよび値は、UNICODE エスケープシーケンス変換することを推奨しますが、必須ではありません。

認証に関する規則

認証レベル

アクセスするデータに応じて、以下の認証レベルを提供します。

- ✓ システム認証：システムデータ（支店情報や設定情報等）へのアクセス用
- ✓ ユーザー認証：EUS が所有するユーザーデータ（口座情報等）へのアクセス用

トークン認証

認証が必要な API では、アクセストークンを利用した認証（「OAuth 2.0 Authorization Framework (RFC6749)」に準拠）を実施します。

API の呼び出し時は、以下の形式で HTTP ヘッダ中にアクセストークンを指定します。

Authorization:Bearer (アクセストークンの値)

API はアクセストークンからアクセス元の権限を特定した上で、適切な動作をします。

データ形式に関する規則

データの ID について

各種の API で取得可能なデータは、当該データを一意に特定するための ID を持ちます。
既存システム上で ID を持つ場合、その値を API 上の ID とする事が望ましいです。

「既存システム上で ID を持たない」「既存システム上で複合 ID になっている」「セキュリティ上の理由により既存システム上の ID を公開できない」等の場合は、API 用の ID を動的に生成しても構いません。
ただし、生成する ID は以下の要件を満たす必要があります。

- ✓ 一意性の担保：当該 API から取得しうる全データ間で、ユニークな値となる事
- ✓ 可逆性の担保：生成した ID から、既存システム上のデータを一意に特定できる事
- ✓ 同一性の担保：同一のデータから生成した ID は、常に同一の値となる事

例えば、API「/branches@GET（支店情報の取得 API）」にて、既存システム上では支店データが ID を持たない場合、以下のように API 用の ID を動的に生成して返却することが可能です。

API 用 ID =(金融機関コード：4 桁)-(支店番号：3 桁) 例）1234-576

タイムゾーンに関する規則

リクエスト・レスポンス共に、日付を扱う場合はタイムゾーンを「JST（UTC+0900）」とします。

API の使用について

API 使用の準備①：FTGAPI が API 使用者に対して提供する情報

FGAPI は、事前に API 使用者に対して以下の情報を通知します。

CLI に通知する情報

- ✓ API キー
- ✓ パスワード

EUS に通知する情報

- ✓ なし

※ API のコール時は、上記の情報を用いて認証等を実施します。

API 使用の準備②：API 使用者が FGAPI に対して提供する情報

API 使用者は、事前に FGAPI に対して以下の情報を提供します。

CLI が提供する情報

✓ 認証サーバのドメイン名（認証系 API でリダイレクト先とするサーバ）

✓ API 呼び出し元サーバの IP アドレス（必要な場合のみ）

EUS が提供する情報

✓ なし

※ API のコール時は、以上の情報を用いてアクセス制限等を実施します。

API 使用の流れ

取引明細を取得して仕訳入力を行うクライアントを例に、基本的な API 使用の流れを説明します。

A) クライアントと FGAPI の連携登録

1. クライアントで実装した連携登録画面を開きます。
2. 連携登録画面から「/auth/login@GET」の画面に遷移し、エンドユーザーの持つ銀行システムのアカウント情報にて認証・認可します。
3. A-2 のコールバック先で「/auth/token@POST」をコールし、アクセストークン・リフレッシュトークンを発行します。
4. 発行したアクセストークン・リフレッシュトークンを、クライアント内に保存します。

B) クライアント上で取引明細を取得して仕訳入力

1. A-4 で保存したリフレッシュトークンを指定して「/auth/token@PUT」をコールし、アクセストークンを発行します。
   ※ アクセストークンが期限内である場合は、アクセストークンを再利用します。
2. B-1 で発行したアクセストークンを指定して「/accounts@GET」をコールし、エンドユーザーの保有する口座情報の一覧を取得します。
3. B-1 で発行したアクセストークン、B-2 で取得した口座情報を指定して「/transactions@GET」をコールし、取引明細の一覧を取得します。

C) クライアントと FGAPI の連携解除（退会等）

1. A-3 で発行したリフレッシュトークンを指定して「/auth/token@DELETE」をコールし、アクセストークン・リフレッシュトークンを削除します。

※各 API の詳細については、次節以降を参照。

API 定義

FGAPI では以下の API を定義します。

分類	パス	HTTP メソッド	トークン認証	説明
システム	/auth/system_token	POST	不要	システム認証用のアクセストークンを

認証				発行します。
ユーザー認証	/auth/login	GET	不要	エンドユーザーの認証・認可をします。 Web 画面として提供します。
ユーザー認証	/auth/login	POST	不要	エンドユーザーの認証・認可をします。 /auth/login@GET の WebAPI 版です。 ※ この API の実装は必須ではありません。
ユーザー認証	/auth/token	POST	システム認証	アクセストークンとリフレッシュトークンを発行します。
ユーザー認証	/auth/token	PUT	システム認証	発行済みのリフレッシュトークンから、アクセストークンとリフレッシュトークンを再発行します。
ユーザー認証	/auth/token	DELETE	システム認証	アクセストークン・リフレッシュトークンを無効化します。
銀行情報	/branches	GET	システム認証	支店情報の一覧を取得します。
口座情報	/accounts	GET	ユーザー認証	エンドユーザーが保有する、口座情報の一覧を取得します。
取引明細	/transactions	GET	ユーザー認証	口座情報に紐づく、取引明細の一覧を取得します。
取引明細	/balances	GET	ユーザー認証	口座情報の日次の残高情報を取得します。

システム認証系 API

/auth/system_token@POST

API 定義

パス	/auth/system_token
HTTP メソッド	POST
トークン認証	不要
説明	システム認証用のアクセストークンを発行します。 アクセストークンは、API の呼び出しごとに、システム全体でユニークかつランダムな文字列として生成します。 また、アクセストークンにはリクエスト元の「AP」と「認可した権限情報」を紐づけて保存します。

	本 API の成功時も、既存のアクセストークンは継続して使用可能です。

POST パラメータ

フィールド	必須	型	値	デフォルト値
api_key	○	string	API キーを指定します。	なし
password	○	string	パスワードを指定します。	なし
grants		string	本認証において認可する権限を指定します。 複数の権限を指定する場合は「,」区切りとします。 ワイルドカードとして「*」を指定可能とします。 ※権限の種別は、FGAPI では定義しません。	なし

レスポンス

フィールド	必須	型	値	デフォルト値
access_token	○	string	発行されたアクセストークン。 有効期限は任意で、60 分程度を推奨。	なし
created_at	○	integer	アクセストークンの生成時間。 UNIX 時間で表記。	なし
expired_at	○	integer	アクセストークンの期限切れ時間。 UNIX 時間で表記。	なし

ユーザー認証系 API

/auth/login@GET

API 定義

パス	/auth/login
HTTP メソッド	GET
トークン認証	不要
説明	ブラウザに認証用の Web 画面を表示し、エンドユーザーの認証・認可を行います。 認証方法に指定はありません。ID・パスワード認証に加え、2 要素認証やパズル認証等、任意の認証方法を使用可能です。 認証に成功した場合、認証トークンを生成し、クエリパラメータで指定されたコールバック先にリダイレクトします。 コールバック先には、GET パラメータで認証トークンを通知します。 認証トークンは、API の呼び出しごとに、システム全体でユニークかつランダムな文字列として生成します。

	また、認証トークンには「認証したエンドユーザー情報」と「認可した権限情報」を紐づけて保存します。 ※1）ユーザー認証の認証・認可の仕組みは、基本的に「OAuth 2.0 Authorization Framework (RFC6749)」に従います。 ※2）本 API は HTML で実装した Web 画面が必要です。

クエリパラメータ

フィールド	必須	型	値	デフォルト値
api_key	○	string	API キーを指定します。	なし
redirect_uri	○	string	リダイレクト先の URI を指定します。 認証サーバのドメイン名と異なる場合、HTTP レスポンスコード 401 が返却されます。	なし
grants		string	本認証において認可する権限を指定します。 複数の権限を指定する場合は「,」区切りとします。 ワイルドカードとして「*」を指定可能とします。 ※権限の種別は、FGAPI では定義しません。	なし

レスポンス

認証に成功すると、callback_uri に以下の GET パラメータを追加した URI にリダイレクトします。

フィールド	必須	型	値	デフォルト値
auth_token	○	string	認証トークン。 有効期限は任意で、3 分以内を推奨。	なし

例えば、calback_uri が「https://fintech-garden.com/fgapi_cb/」、認証トークンが「XYZ1234」である場合、「https://fintech-garden.com/fgapi_cb/?auth_token= XYZ1234」にリダイレクトします。

/auth/login@POST

API 定義

パス	/auth/login
HTTP メソッド	POST
トークン認証	不要
説明	/auth/login@GET の機能を RestAPI として提供し、エンドユーザーの認証・認可を行います。 ※ GET 版と比較してセキュリティレベルが劣るため、この API の実装は必須ではありません。実装しない場合は、API の呼び出し時に HTTP レスポンスコード 404 を返却してください。

クエリパラメータ

フィールド	必須	型	値	デフォルト値
api_key	○	string	API キーを指定します。	なし
grants		string	本認証において認可する権限を指定します。 複数の権限を指定する場合は「,」区切りとします。 ワイルドカードとして「*」を指定可能とします。 ※権限の種別は、FGAPI では定義しません。	なし
id	○	string	ログイン ID を指定します。 複数の ID を組み合わせる必要がある場合、任意の文字列で連結して送付します。（店番＋口座番号など）	
password	○	string	ログインパスワードを指定します。 複数のパスワードを組み合わせる必要がある場合、任意の文字列で連結して送付します。	

レスポンス

フィールド	必須	型	値	デフォルト値
auth_token	○	string	認証トークン。 有効期限は任意で、3 分以内を推奨。	なし

/auth/token@POST

API 定義

パス	/auth/token
HTTP メソッド	POST
トークン認証	システム認証
説明	/auth/login@GET で認証したエンドユーザーに対応する、アクセストークンとリフレッシュトークンを発行します。 アクセストークン・リフレッシュトークンは、API の呼び出しごとに、システム全体でユニークかつランダムな文字列として生成します。 また、アクセストークン・リフレッシュトークンには「認証したエンドユーザー情報」と「認可した権限情報」を紐づけて保存します。 本 API の成功時、既存のアクセストークンとリフレッシュトークンは即時に無効化されます。 ※以降の各種 API では、ここで発行したアクセストークンを認証キーとして使用します。

POST パラメータ

フィールド	必須	型	値	デフォルト値

api_key	○	string	API キーを指定します。	なし
password	○	string	パスワードを指定します。	なし
auth_token	○	string	/auth/login@GET のコールバック先が受け取った、認証トークンを指定します。	なし

レスポンス

フィールド	必須	型	値	デフォルト値
access_token	○	string	発行されたアクセストークン。 有効期限は任意で、60 分程度を推奨。	なし
created_at	○	integer	アクセストークンの生成時間。 UNIX 時間で表記。	なし
expired_at	○	integer	アクセストークンの期限切れ時間。 UNIX 時間で表記。	なし
refresh_token	○	string	発行されたリフレッシュトークン。 有効期間は任意で、180 日程度を推奨。	なし

/auth/token@PUT

API 定義

パス	/auth/token
HTTP メソッド	PUT
トークン認証	システム認証
説明	/auth/token@GET で発行したリフレッシュトークンから、アクセストークンとリフレッシュトークンを再発行します。 本 API の成功時、既存のアクセストークンとリフレッシュトークンは即時に無効化されます。

POST パラメータ

フィールド	必須	型	値	デフォルト値
api_key	○	string	API キーを指定します。	なし
refresh_token	○	string	再発行対象のリフレッシュトークンを指定します。	なし

レスポンス

フィールド	必須	型	値	デフォルト値
access_token	○	string	アクセストークン。 有効期限は任意で、60 分程度を推奨。	なし
created_at	○	integer	アクセストークンの生成時間。 UNIX 時間で表記。	なし
expired_at	○	integer	アクセストークンの期限切れ時間。 UNIX 時間で表記。	なし

refresh_token	○	string	リフレッシュトークン。 有効期間は任意で、180 日程度を推奨。	なし

/auth/token@DELETE

API 定義

パス	/auth/token
HTTP メソッド	DELETE
トークン認証	システム認証
説明	アクセストークン・リフレッシュトークンを無効化します。 当該エンドユーザーからのアクセスを停止する際に使用します。 本 API の成功時、既存のアクセストークンとリフレッシュトークンは即時に無効化されます。

POST パラメータ

フィールド	必須	型	値	デフォルト値
api_key	○	string	API キーを指定します。	なし
refresh_token	○	string	削除対象のリフレッシュトークンを指定します。	なし

レスポンス

なし。

HTTP ステータスコード 200 にて無効化の成功を表します。

銀行情報系 API

/branches@GET

API 定義

パス	/branches
HTTP メソッド	GET
トークン認証	システム認証
説明	支店情報の一覧を取得します。

クエリパラメータ

フィールド	必須	型	値	デフォルト値
page		integer	取得対象のページ番号。	1

レスポンス

フィールド	必須	型	値
bank_code	○	string	銀行コード
bank_name	○	string	銀行の名称。
branches	○	object[]	支店情報のデータ一覧。

			支店情報が 0 件の場合、空の配列を返却します。
id	○	string	支店情報の ID。 すべての支店情報間でユニークであり、ID から支店情報を一意に特定できる必要があります。 ※既存システム上にて複合キーで管理している場合、複合キーを連結した値を ID として使用しても構いません。
branch_code	○	string	支店コード。
branch_name	○	string	支店名。
branch_name_kana	○	string	支店名のフリガナ
address_code		string	支店の所在地コード。 総務省の規定する「都道府県コード及び市区町村コード」より、該当する「団体コード」を選択します。
tel		string	支店の電話番号。
params	○	object	
page	○	integer	取得対象のページ番号。
next_page	○	integer	次ページのページ番号。 次ページが存在しない場合は「0」とします。

口座情報系 API

/accounts@GET

API 定義

パス	/accounts
HTTP メソッド	GET
トークン認証	ユーザー認証
説明	エンドユーザーが保有する、口座情報の一覧を取得します。 クレジットカード情報が含まれる場合、クレジットカードの情報も独立した口座として扱うことが可能です。 口座情報の件数が多い場合、レスポンスは 200 件/ページでページングされます。

クエリパラメータ

フィールド	必須	型	値	デフォルト値
page		integer	取得対象のページ番号。	1

レスポンス

フィールド	必須	型	値
bank_code	○	string	銀行コード

bank_name	○	string	銀行の名称。
accounts	○	object[]	口座情報のデータ一覧。 口座情報が 0 件の場合、空の配列を返却します。
id	○	string	口座情報の ID。 すべての口座情報間でユニークであり、ID から口座番号を一意に特定できる必要があります。 ※既存システム上にて複合キーで管理している場合、複合キーを連結した値を ID として使用しても構いません。
account_number	○	口座番号	口座番号。 口座番号を非公開にする場合、全体もしくは一部を* 等でマスクしても良い。
owner_type	○	string	口座の保有者の種別を表す。 ✓ c：法人 ✓ p：個人
account_type1	○	string	口座の大分類を表す。 ✓ bank：銀行口座 ✓ cc：クレジットカード口座
account_type2	○	string	普通預金、当座預金などの、口座の中分類を表す。 任意の値を返却可能ですが、預金口座である場合は、以下の値を推奨します。 ✓ 普通 ※普通預金の場合 ✓ 当座 ※当座預金の場合 ✓ 総合 ※総合預金の場合 ✓ 定期 ※定期預金の場合
account_type3		string	システム予約。 常に空欄となります。
branch_code	○	string	口座の所属する支店コード。
branch_name	○	string	口座の所属する支店名。
params	○	object	
page	○	integer	取得対象のページ番号。
next_page	○	integer	次ページのページ番号。 次ページが存在しない場合は「0」とします。

取引情報系 API

/transactions

API 定義

パス	/transactions
HTTP メソッド	GET
トークン認証	ユーザー認証
説明	指定した期間内の取引明細の一覧を取得します。 取引明細の件数が多い場合、レスポンスは 200 件/ページでページングされます。

クエリパラメータ

フィールド	必須	型	値	デフォルト値
account_id	○	string	取得対象の口座 ID。	なし
start_date	○	date	取得対象とする取引日の開始日。 書式は YYYY-MM-DD です。 指定した日付も取得対象に含みます。	なし
end_date	○	date	取得対象とする取引日の終了日。 書式は YYYY-MM-DD です。 指定した日付も取得対象に含みます。	なし
page		integer	取得対象のページ番号。	1

クエリパラメータの指定例

```
/transactions/list?account_id=12345-abc&start_date=2020-01-20&end_date=2020-02-19
```

レスポンス

フィールド	必須	型	値
transactions	○	object[]	取引明細のデータ一覧。 取得対象の取引明細が 0 件の場合は、空の配列を返却します。 ※取り消された取引明細は返却対象から除外してください。
id	○	string	取引明細の ID。 すべての取引明細間でユニークであり、ID から取引明細を一意に特定できる必要があります。 ※既存システム上の ID と同一の値である必要はありません。
date	○	date	取引日。 書式は ISO8601 の拡張形式に従います。 例）2020-01-10T13:15:35+09:00
amount	○	integer	金額。 入金の場合は正の値、出金の場合は負の値とします。

description	○	string	取引内容。 振込人名義や摘要など、エンドユーザーが取引内容を理解できる値を返却します。
balance	○	integer	取引終了時点における口座残高。
params	○	object	
account_id	○	string	取得対象の口座 ID。
start_date	○	string	取得対象の開始日。 書式は YYYY-MM-DD です。
end_date	○	string	取得対象の終了日。 書式は YYYY-MM-DD です。
page	○	integer	取得対象のページ番号。
next_page	○	integer	次ページのページ番号。 次ページが存在しない場合は「0」とします。

/balances

API 定義

パス	/balances
HTTP メソッド	GET
トークン認証	ユーザー認証
説明	指定した期間内の日次の残高情報を取得します。 日次の残高とは、当該の日付中で、最後の取引後の残高を指します。 残高情報の件数が多い場合、レスポンスは 200 件/ページでページングされます。

クエリパラメータ

フィールド	必須	型	値	デフォルト値
account_id	○	string	取得対象の口座 ID。	なし
start_date	○	date	取得対象とする開始日。 書式は YYYY-MM-DD です。 指定した日付も取得対象に含みます。	なし
end_date	○	date	取得対象とする終了日。 書式は YYYY-MM-DD です。 指定した日付も取得対象に含みます。	なし
page		integer	取得対象のページ番号。	1

クエリパラメータの指定例

```
/balances/list?account_id=12345-abc&start_date=2020-01-20&end_date=2020-02-19
```

レスポンス

フィールド	必須	型	値
balances	○	object[]	日次の残高データの一覧。 取得対象の残高データが 0 件の場合は、空の配列

			を返却します。
date	○	date	残高の取得対象の日付。 書式は ISO8601 の拡張形式に従います。 例）2020-01-10T13:15:35+09:00
balance	○	integer	date 中、最後の取引後の残高。
params	○	object	
account_id	○	string	取得対象の口座 ID。
start_date	○	string	取得対象の開始日。 書式は YYYY-MM-DD です。
end_date	○	string	取得対象の終了日。 書式は YYYY-MM-DD です。
page	○	integer	取得対象のページ番号。
next_page	○	integer	次ページのページ番号。 次ページが存在しない場合は「0」とします。