ProjectX Gateway API ドキュメント

目次

ProjectX Gateway API

ProjectX Trading, LLC - その取引プラットフォーム ProjectX を通じて、プロップファームや評価プロバイダー向けの完全なエンドツーエンドソリューションを提供しています。これらの機能には、アカウントのカスタマイズ、リスクルールとモニタリング、清算、統計、堅牢な権限設定が含まれます。このAPIはREST APIアーキテクチャを利用して、プロップファームのトレーダー操作を管理します。

必要条件

- REST APIの理解
- cURLまたはPostmanでサンプルリクエストを作成する能力

参照元: https://gateway.docs.projectx.com/docs/intro/

Getting Started

当社は、企業のあらゆる側面にアクセスし管理するための非常に堅牢なAPIを設計しました。

参照元: https://gateway.docs.projectx.com/docs/category/getting-started/

Authenticate

認証関連のドキュメントです。

参照元: https://gateway.docs.projectx.com/docs/category/authenticate/

Authenticate (with API key)

当社はAPIに送信されるすべてのリクエストを認証するためにJSON Web Tokenを利用しています。このプロセスには、今後のリクエストに必要なセッショントークンの取得が含まれます。

ステップ 1

始めるには、以下のものが必要です：

- 企業から取得したAPIキー。これらの認証情報がない場合は、企業にお問い合わせください。
- こちらから取得した接続URL。

ステップ 2

API リファレンス: Login API

ユーザー名とAPIキーを使用して**POST**リクエストを作成します。

```
curl -X 'POST' \
  'https://gateway-api-demo.s2f.projectx.com/api/Auth/loginKey' \
  -H 'accept: text/plain' \
  -H 'Content-Type: application/json' \
  -d '{
  "userName": "string",
  "apiKey": "string"
}'
```

ステップ 3

APIレスポンスを処理し、結果がSuccess (0)であることを確認してから、セッショントークンを安全な場所に保存します。 このセッショントークンは、Gateway APIへの完全なアクセスを許可します。

```
{
    "token": "your_session_token_here",
    "success": true,
    "errorCode": 0,
    "errorMessage": null
}
```

注意事項

以降のすべてのリクエストでは、`Bearer`メソッドを使用して**"Authorization"** HTTPヘッダーにセッショントークンを提供する必要があります。

セッショントークンの有効期間は24時間です。同じセッションを継続して使用するには、トークンを再検証する必要があります。

次のステップでは、トークンの有効期限が切れた場合にセッションを延長/再検証する方法について説明します。

参照元: https://gateway.docs.projectx.com/docs/getting-started/authenticate/authenticate-api-key/

Authenticate (for authorized applications)

当社はAPIに送信されるすべてのリクエストを認証するためにJSON Web Tokenを利用しています。

ステップ 1

企業に提供されている管理者認証情報（ユーザー名とパスワード、appId、verifyKey）を取得します。APIで認証するにはこれらの認証情報が必要です。

これらの認証情報をお持ちでない場合は、詳細についてアカウントマネージャーにお問い合わせください。

ステップ 2

API リファレンス: Login API

ユーザー名とパスワードを使用して**POST**リクエストを作成します。

```
curl -X 'POST' \
  'https://gateway-api-demo.s2f.projectx.com/api/Auth/loginApp' \
  -H 'accept: text/plain' \
  -H 'Content-Type: application/json' \
  -d '{
  "userName": "yourUsername",
  "password": "yourPassword",
  "deviceId": "yourDeviceId",
  "appId": "B76015F2-04D3-477E-9191-C5E22CB2C957",
  "verifyKey": "yourVerifyKey"
}'
```

ステップ 3

APIレスポンスを処理し、結果がSuccess (0)であることを確認してから、セッショントークンを安全な場所に保存します。 このセッショントークンは、Gateway APIへの完全なアクセスを許可します。

```
{
    "token": "your_session_token_here",
    "success": true,
    "errorCode": 0,
    "errorMessage": null
}
```

注意事項

以降のすべてのリクエストでは、`Bearer`メソッドを使用して**"Authorization"** HTTPヘッダーにセッショントークンを提供する必要があります。

セッショントークンの有効期間は24時間です。同じセッションを継続して使用するには、トークンを再検証する必要があります。

次のステップでは、トークンの有効期限が切れた場合にセッションを延長/再検証する方法について説明します。

参照元: https://gateway.docs.projectx.com/docs/getting-started/authenticate/authenticate-as-application/

Connection URLs

接続URLページでは、異なる環境を選択できます。以下の環境が利用可能です：

- Alpha Ticks
- Blue Guardian
- Blusky
- E8
- Funding Futures
- The Futures Desk
- Futures Elite
- FX
- IFY Futures
- GoatFunded
- TickTickTrader
- TopOneFutures
- Topstep
- XTX3Funding

それぞれの環境には、その環境で使用するための特定の接続URLがあります。

参照元: https://gateway.docs.projectx.com/docs/getting-started/connection-urls/

Placing Your First Order

このドキュメントでは、APIを使用して最初の注文を行うプロセスについて説明します。注文を正常に実行するには、ユーザーに関連付けられたアクティブな取引アカウントが必要です。アカウント詳細の取得、利用可能な契約の閲覧、注文の実行に関する手順に従ってください。

ステップ 1

注文プロセスを開始するには、まずユーザーにリンクされたアクティブなアカウントのリストを取得する必要があります。このステップは、注文を行う前にアカウントのステータスを確認するために不可欠です。

API URL: POST https://gateway-api-demo.s2f.projectx.com/api/account/search

API リファレンス: /api/account/search

リクエスト

```
{
  "onlyActiveAccounts": true
}
```

レスポンス

```
{
  "accounts": [
    {
      "id": 1,
      "name": "TEST_ACCOUNT_1",
      "canTrade": true,
      "isVisible": true
    }
  ],
  "success": true,
  "errorCode": 0,
  "errorMessage": null
}
```

cURLリクエスト

```
curl -X 'POST' \
  'https://gateway-api-demo.s2f.projectx.com/api/Account/search' \
  -H 'accept: text/plain' \
  -H 'Content-Type: application/json' \
```

```
  -d '{
  "onlyActiveAccounts": true
}'
```

ステップ 2

アクティブなアカウントを確認したら、次のステップは取引可能な契約のリストを取得することです。この情報は、注文に適切な契約を選択するのに役立ちます。

API URL: POST https://gateway-api-demo.s2f.projectx.com/api/contract/search

API リファレンス: /api/contract/search

リクエスト

```
{
  "live": false,
  "searchText": "NQ"
}
```

レスポンス

```
{
  "contracts": [
    {
      "id": "CON.F.US.ENQ.H25",
      "name": "ENQH25",
      "description": "E-mini NASDAQ-100: March 2025",
      "tickSize": 0.25,
      "tickValue": 5,
      "activeContract": true
    },
    {
      "id": "CON.F.US.MNQ.H25",
      "name": "MNQH25",
      "description": "Micro E-mini Nasdaq-100: March 2025",
      "tickSize": 0.25,
      "tickValue": 0.5,
      "activeContract": true
    },
    {
      "id": "CON.F.US.NQG.G25",
      "name": "NQGG25",
      "description": "E-Mini Natural Gas: February 2025",
      "tickSize": 0.005,
      "tickValue": 12.5,
      "activeContract": true
    },
```

```
    {
      "id": "CON.F.US.NQM.G25",
      "name": "NQMG25",
      "description": "E-Mini Crude Oil: February 2025",
      "tickSize": 0.025,
      "tickValue": 12.5,
      "activeContract": true
    }
  ],
  "success": true,
  "errorCode": 0,
  "errorMessage": null
}
```


cURLリクエスト

```
curl -X 'POST' \
  'https://gateway-api-demo.s2f.projectx.com/api/Contract/search' \
  -H 'accept: text/plain' \
  -H 'Content-Type: application/json' \
  -d '{
  "live": false,
  "searchText": "NQ"
}'
```

最終ステップ

アカウントIDと選択した契約IDをメモしたら、注文を行う準備が整いました。正常な取引を行うために正確な詳細を提供するようにしてください。

API URL: POST https://gateway-api-demo.s2f.projectx.com/api/order/place

API リファレンス: /api/order/place

パラメータ

名前	型	説明	必須	null 許容
accountId	integer	アカウントID。	必須	false
contractId	string	契約ID。	必須	false
type	integer	注文タイプ: 1 = Limit 2 = Market 4 = Stop 5 = TrailingStop 6 = JoinBid 7 = JoinAsk	必須	false
side	integer	注文のサイド: 0 = Bid (買い) 1 = Ask (売り)	必須	false
size	integer	注文のサイズ。	必須	false

名前	型	説明	必須	null 許容
limitPrice	decimal	該当する場合、注文の指値価格。	オプション	true
stopPrice	decimal	該当する場合、注文の逆指値価格。	オプション	true
trailPrice	decimal	該当する場合、注文のトレイル価格。	オプション	true
customTag	string	注文のオプションカスタムタグ。	オプション	true
linkedOrderId	integer	リンクされた注文ID。	オプション	true

リクエスト

```
{
  "accountId": 1,
  "contractId": "CON.F.US.DA6.M25",
  "type": 2,
  "side": 1,
  "size": 1,
  "limitPrice": null,
  "stopPrice": null,
  "trailPrice": null,
  "customTag": null,
  "linkedOrderId": null
}
```

レスポンス

```
{
  "orderId": 9056,
  "success": true,
  "errorCode": 0,
  "errorMessage": null
}
```

cURLリクエスト

```
curl -X 'POST' \
  'https://gateway-api-demo.s2f.projectx.com/api/Order/place' \
```

```
    -H 'accept: text/plain' \
    -H 'Content-Type: application/json' \
    -d '{
    "accountId": 1,
    "contractId": "CON.F.US.DA6.M25",
    "type": 2,
    "side": 1,
    "size": 1,
    "limitPrice": null,
    "stopPrice": null,
    "trailPrice": null,
    "customTag": null,
    "linkedOrderId": null
  }'
```

参照元: https://gateway.docs.projectx.com/docs/getting-started/placing-your-first-order/

API Reference

API リファレンスのセクションです。

参照元: https://gateway.docs.projectx.com/docs/category/api-reference/

Account

アカウント関連のAPIです。

参照元: https://gateway.docs.projectx.com/docs/category/account/

Search Accounts

API URL: POST https://gateway-api-demo.s2f.projectx.com/api/Account/search

API リファレンス: /api/account/search

説明

アカウントを検索します。

パラメータ

名前	型	説明	必須	null許容
onlyActiveAccounts	boolean	アクティブなアカウントのみをフィルタリングするかどうか。	必須	false

使用例

リクエスト例

```
curl -X 'POST' \
  'https://gateway-api-demo.s2f.projectx.com/api/Account/search' \
  -H 'accept: text/plain' \
  -H 'Content-Type: application/json' \
  -d '{
  "onlyActiveAccounts": true
}'
```

レスポンス例

```
{
  "accounts": [
    {
      "id": 1,
      "name": "TEST_ACCOUNT_1",
      "balance": 50000,
      "canTrade": true,
      "isVisible": true
    }
  ],
  "success": true,
  "errorCode": 0,
  "errorMessage": null
}
```

エラーの場合: response status is 401

参照元: https://gateway.docs.projectx.com/docs/api-reference/account/search-accounts/

Market Data

市場データに関するAPIです。

参照元: https://gateway.docs.projectx.com/docs/category/market-data/

Search Contracts

API URL: POST https://gateway-api-demo.s2f.projectx.com/api/Contract/search

API リファレンス: /api/contract/search

説明

契約を検索します。

パラメータ

名前	型	説明	必須	null許容
searchText	string	検索する契約の名前。	必須	false
live	boolean	sim/liveデータサブスクリプションを使用して契約を検索するかどうか。	必須	false

使用例

リクエスト例

```
curl -X 'POST' \
  'https://gateway-api-demo.s2f.projectx.com/api/Contract/search' \
  -H 'accept: text/plain' \
  -H 'Content-Type: application/json' \
  -d '{
  "live": false,
  "searchText": "NQ"
}'
```

レスポンス例

```
{
  "contracts": [
    {
      "id": "CON.F.US.ENQ.H25",
      "name": "ENQH25",
      "description": "E-mini NASDAQ-100: March 2025",
      "tickSize": 0.25,
      "tickValue": 5,
      "activeContract": true
    },
    {
      "id": "CON.F.US.MNQ.H25",
      "name": "MNQH25",
      "description": "Micro E-mini Nasdaq-100: March 2025",
      "tickSize": 0.25,
      "tickValue": 0.5,
      "activeContract": true
    },
    {
      "id": "CON.F.US.NQG.G25",
      "name": "NQGG25",
      "description": "E-Mini Natural Gas: February 2025",
```

```
      "tickSize": 0.005,
      "tickValue": 12.5,
      "activeContract": true
    },
    {
      "id": "CON.F.US.NQM.G25",
      "name": "NQMG25",
      "description": "E-Mini Crude Oil: February 2025",
      "tickSize": 0.025,
      "tickValue": 12.5,
      "activeContract": true
    }
  ],
  "success": true,
  "errorCode": 0,
  "errorMessage": null
}
```


エラーの場合: response status is 401

参照元: https://gateway.docs.projectx.com/docs/api-reference/market-data/search-contracts/

Search Contracts By ID

API URL: POST https://gateway-api-demo.s2f.projectx.com/api/Contract/searchById

API リファレンス: /api/contract/searchbyid

説明

契約をIDで検索します。

パラメータ

名前	型	説明	必須	null許容
contractId	string	検索する契約のID。	必須	false

使用例

リクエスト例

```
curl -X 'POST' \
  'https://gateway-api-demo.s2f.projectx.com/api/Contract/searchById' \
  -H 'accept: text/plain' \
  -H 'Content-Type: application/json' \
  -d '{
  "contractId": "CON.F.US.ENQ.H25"
}'
```

レスポンス例


```
{
  "contracts": [
    {
      "id": "CON.F.US.ENQ.H25",
      "name": "ENQH25",
      "description": "E-mini NASDAQ-100: March 2025",
      "tickSize": 0.25,
      "tickValue": 5,
      "activeContract": true
    }
  ],
  "success": true,
  "errorCode": 0,
  "errorMessage": null
}
```


エラーの場合: response status is 401

参照元: https://gateway.docs.projectx.com/docs/api-reference/market-data/search-contracts-by-id/

Retrieve Bars

API URL: POST https://gateway-api-demo.s2f.projectx.com/api/History/retrieveBars

API リファレンス: /api/history/retrieveBars

説明

価格バーを取得します。

パラメータ

名前	型	説明	必須	null許容
contractId	integer	契約ID。	必須	false
live	boolean	simまたはliveデータサブスクリプションを使用してバーを取得するかどうか。	必須	false
startTime	datetime	履歴データの開始時間。	必須	false

名前	型	説明	必須	null許容
endTime	datetime	履歴データの終了時間。	必須	false
unit	integer	履歴データの集約単位: 1 = 秒 2 = 分 3 = 時間 4 = 日 5 = 週 6 = 月	必須	false
unitNumber	integer	集約する単位の数。	必須	false
limit	integer	取得するバーの最大数。	必須	false
includePartialBar	boolean	現在の時間単位を表す部分的なバーを含めるかどうか。	必須	false

使用例

リクエスト例

```
curl -X 'POST' \
  'https://gateway-api-demo.s2f.projectx.com/api/History/retrieveBars' \
  -H 'accept: text/plain' \
  -H 'Content-Type: application/json' \
  -d '{
  "contractId": "CON.F.US.RTY.Z24",
  "live": false,
  "startTime": "2024-12-01T00:00:00Z",
  "endTime": "2024-12-31T21:00:00Z",
  "unit": 3,
  "unitNumber": 1,
  "limit": 7,
  "includePartialBar": false
}'
```

レスポンス例

```
{
  "bars": [
    {
      "t": "2024-12-20T14:00:00+00:00",
      "o": 2208.100000000,
      "h": 2217.000000000,
      "l": 2206.700000000,
      "c": 2210.100000000,
      "v": 87
```

```
      },
      {
        "t": "2024-12-20T13:00:00+00:00",
        "o": 2195.800000000,
        "h": 2215.000000000,
        "l": 2192.900000000,
        "c": 2209.800000000,
        "v": 536
      },
      {
        "t": "2024-12-20T12:00:00+00:00",
        "o": 2193.600000000,
        "h": 2200.300000000,
        "l": 2192.000000000,
        "c": 2198.000000000,
        "v": 180
      },
      {
        "t": "2024-12-20T11:00:00+00:00",
        "o": 2192.200000000,
        "h": 2194.800000000,
        "l": 2189.900000000,
        "c": 2194.800000000,
        "v": 174
      },
      {
        "t": "2024-12-20T10:00:00+00:00",
        "o": 2200.400000000,
        "h": 2200.400000000,
        "l": 2191.000000000,
        "c": 2193.100000000,
        "v": 150
      },
      {
        "t": "2024-12-20T09:00:00+00:00",
        "o": 2205.000000000,
        "h": 2205.800000000,
        "l": 2198.900000000,
        "c": 2200.500000000,
        "v": 56
      },
      {
        "t": "2024-12-20T08:00:00+00:00",
        "o": 2207.700000000,
        "h": 2210.100000000,
        "l": 2198.100000000,
        "c": 2204.900000000,
        "v": 144
      }
    ],
    "success": true,
    "errorCode": 0,
    "errorMessage": null
  }
```

エラーの場合: response status is 401

参照元: https://gateway.docs.projectx.com/docs/api-reference/market-data/retrieve-bars/

Orders

注文関連のAPIです。

参照元: https://gateway.docs.projectx.com/docs/category/orders/

Order Place

API URL: POST https://gateway-api-demo.s2f.projectx.com/api/Order/place

API リファレンス: /api/order/place

説明

注文を行います。

パラメータ

名前	型	説明	必須	null 許容
accountId	integer	アカウントID。	必須	false
contractId	string	契約ID。	必須	false
type	integer	注文タイプ: 1 = Limit 2 = Market 4 = Stop 5 = TrailingStop 6 = JoinBid 7 = JoinAsk	必須	false
side	integer	注文のサイド: 0 = Bid (買い) 1 = Ask (売り)	必須	false
size	integer	注文のサイズ。	必須	false
limitPrice	decimal	該当する場合、注文の指値価格。	オプション	true
stopPrice	decimal	該当する場合、注文の逆指値価格。	オプション	true
trailPrice	decimal	該当する場合、注文のトレイル価格。	オプション	true
customTag	string	注文のオプションカスタムタグ。	オプション	true
linkedOrderId	integer	リンクされた注文ID。	オプション	true

使用例

リクエスト例

```
curl -X 'POST' \
  'https://gateway-api-demo.s2f.projectx.com/api/Order/place' \
  -H 'accept: text/plain' \
  -H 'Content-Type: application/json' \
  -d '{
  "accountId": 465,
  "contractId": "CON.F.US.DA6.M25",
  "type": 2,
  "side": 1,
  "size": 1,
  "limitPrice": null,
  "stopPrice": null,
  "trailPrice": null,
  "customTag": null,
  "linkedOrderId": null
}'
```

レスポンス例

```
{
  "orderId": 9056,
  "success": true,
  "errorCode": 0,
  "errorMessage": null
}
```

エラーの場合: response status is 401

参照元: https://gateway.docs.projectx.com/docs/api-reference/order/order-place/

Order Modify

API URL: POST https://gateway-api-demo.s2f.projectx.com/api/Order/modify

API リファレンス: /api/order/modify

説明

オープン注文を変更します。

パラメータ

名前	型	説明	必須	null許容
accountId	integer	アカウントID。	必須	false
orderId	integer	注文ID。	必須	false
size	integer	注文のサイズ。	オプション	true
limitPrice	decimal	該当する場合、注文の指値価格。	オプション	true
stopPrice	decimal	該当する場合、注文の逆指値価格。	オプション	true
trailPrice	decimal	該当する場合、注文のトレイル価格。	オプション	true

使用例

リクエスト例

```
curl -X 'POST' \
  'https://gateway-api-demo.s2f.projectx.com/api/Order/modify' \
  -H 'accept: text/plain' \
  -H 'Content-Type: application/json' \
  -d '{
  "accountId": 465,
  "orderId": 26974,
  "size": 1,
  "limitPrice": null,
  "stopPrice": 1604,
  "trailPrice": null
}
'
```

レスポンス例

```
{
  "success": true,
  "errorCode": 0,
  "errorMessage": null
}
```

エラーの場合: response status is 401

参照元: https://gateway.docs.projectx.com/docs/api-reference/order/order-modify/

Order Cancel

API URL: POST https://gateway-api-demo.s2f.projectx.com/api/Order/cancel

API リファレンス: /api/order/cancel

説明

注文をキャンセルします。

パラメータ

名前	型	説明	必須	null許容
accountId	integer	アカウントID。	必須	false
orderId	integer	注文ID。	必須	false

使用例

リクエスト例

```
curl -X 'POST' \
  'https://gateway-api-demo.s2f.projectx.com/api/Order/cancel' \
  -H 'accept: text/plain' \
  -H 'Content-Type: application/json' \
  -d '{
  "accountId": 465,
  "orderId": 26974
}'
```

レスポンス例

```
{
  "success": true,
  "errorCode": 0,
  "errorMessage": null
}
```

エラーの場合: response status is 401

参照元: https://gateway.docs.projectx.com/docs/api-reference/order/order-cancel/

Order Search

API URL: POST https://gateway-api-demo.s2f.projectx.com/api/Order/search

API リファレンス: /api/order/search

説明

注文を検索します。

パラメータ

名前	型	説明	必須	null許容
accountId	integer	アカウントID。	必須	false
startTimestamp	datetime	タイムスタンプフィルターの開始。	必須	false
endTimestamp	datetime	タイムスタンプフィルターの終了。	オプション	true

使用例

リクエスト例

```
curl -X 'POST' \
  'https://gateway-api-demo.s2f.projectx.com/api/Order/search' \
  -H 'accept: text/plain' \
  -H 'Content-Type: application/json' \
  -d '{
  "accountId": 202,
  "startTimestamp": "2024-12-30T16:48:16.003Z",
  "endTimestamp": "2025-12-30T16:48:16.003Z"
}'
```

レスポンス例


```
{
  "orders": [
    {
      "id": 26060,
      "accountId": 545,
      "contractId": "CON.F.US.EP.M25",
      "creationTimestamp": "2025-04-14T17:49:10.142532+00:00",
      "updateTimestamp": null,
      "status": 2,
      "type": 2,
      "side": 0,
      "size": 1,
      "limitPrice": null,
      "stopPrice": null
    },
    {
      "id": 26062,
      "accountId": 545,
      "contractId": "CON.F.US.EP.M25",
      "creationTimestamp": "2025-04-14T17:49:53.043234+00:00",
      "updateTimestamp": null,
```

```
        "status": 2,
        "type": 2,
        "side": 1,
        "size": 1,
        "limitPrice": null,
        "stopPrice": null
      }
    ],
    "success": true,
    "errorCode": 0,
    "errorMessage": null
  }
```


エラーの場合: response status is 401

参照元: https://gateway.docs.projectx.com/docs/api-reference/order/order-search/

Order Search Open

API URL: POST https://gateway-api-demo.s2f.projectx.com/api/Order/searchOpen

API リファレンス: /api/order/searchopen

説明

オープン注文を検索します。

パラメータ

名前	型	説明	必須	null許容
accountId	integer	アカウントID。	必須	false

使用例

リクエスト例

```
curl -X 'POST' \
  'https://gateway-api-demo.s2f.projectx.com/api/Order/search' \
  -H 'accept: text/plain' \
  -H 'Content-Type: application/json' \
  -d '{
  "accountId": 212
}'
```

レスポンス例

```
{
  "orders": [
    {
      "id": 26970,
      "accountId": 212,
      "contractId": "CON.F.US.EP.M25",
      "creationTimestamp": "2025-04-21T19:45:52.105808+00:00",
      "updateTimestamp": "2025-04-21T19:45:52.105808+00:00",
      "status": 1,
      "type": 4,
      "side": 1,
      "size": 1,
      "limitPrice": null,
      "stopPrice": 5138.000000000
    }
  ],
  "success": true,
  "errorCode": 0,
  "errorMessage": null
}
```


エラーの場合: response status is 401

参照元: https://gateway.docs.projectx.com/docs/api-reference/order/order-search-open/

Positions

ポジション関連のAPIです。

参照元: https://gateway.docs.projectx.com/docs/category/positions/

Search Open Positions

API URL: POST https://gateway-api-demo.s2f.projectx.com/api/Position/searchOpen

API リファレンス: /api/position/searchOpen

説明

オープンポジションを検索します。

パラメータ

名前	型	説明	必須	null許容
accountId	integer	アカウントID。	必須	false

使用例

リクエスト例

```
curl -X 'POST' \
  'https://gateway-api-demo.s2f.projectx.com/api/Position/searchOpen' \
  -H 'accept: text/plain' \
  -H 'Content-Type: application/json' \
  -d '{
  "accountId": 536
}'
```

レスポンス例

```
{
  "positions": [
    {
      "id": 6124,
      "accountId": 536,
      "contractId": "CON.F.US.GMET.J25",
      "creationTimestamp": "2025-04-21T19:52:32.175721+00:00",
      "type": 1,
      "size": 2,
      "averagePrice": 1575.750000000
    }
  ],
  "success": true,
  "errorCode": 0,
  "errorMessage": null
}
```

エラーの場合: response status is 401

参照元: https://gateway.docs.projectx.com/docs/api-reference/positions/search-open-positions/

Close Positions

API URL: POST https://gateway-api-demo.s2f.projectx.com/api/Position/closeContract

API リファレンス: /api/position/closeContract

説明

ポジションをクローズします。

パラメータ

名前	型	説明	必須	null許容

名前	型	説明	必須	null許容
accountId	integer	アカウントID。	必須	false
contractId	string	契約ID。	必須	false

使用例

リクエスト例

```
curl -X 'POST' \
  'https://gateway-api-demo.s2f.projectx.com/api/Position/closeContract' \
  -H 'accept: text/plain' \
  -H 'Content-Type: application/json' \
  -d '{
  "accountId": 536,
  "contractId": "CON.F.US.GMET.J25"
}'
```

レスポンス例

```
{
  "success": true,
  "errorCode": 0,
  "errorMessage": null
}
```

エラーの場合: response status is 401

参照元: https://gateway.docs.projectx.com/docs/api-reference/positions/close-positions/

Close Positions Partial

API URL: POST https://gateway-api-demo.s2f.projectx.com/api/Position/partialCloseContract

API リファレンス: /api/position/partialclosecontract

説明

ポジションを部分的にクローズします。

パラメータ

名前	型	説明	必須	null許容

名前	型	説明	必須	null許容
accountId	integer	アカウントID。	必須	false
contractId	string	契約ID。	必須	false
size	integer	クローズするサイズ。	必須	false

使用例

リクエスト例

```
curl -X 'POST' \
  'https://gateway-api-demo.s2f.projectx.com/api/Position/partialCloseContract' \
  -H 'accept: text/plain' \
  -H 'Content-Type: application/json' \
  -d '{
  "accountId": 536,
  "contractId": "CON.F.US.GMET.J25",
  "size": 1
}'
```

レスポンス例

```
{
  "success": true,
  "errorCode": 0,
  "errorMessage": null
}
```

エラーの場合: response status is 401

参照元: https://gateway.docs.projectx.com/docs/api-reference/positions/close-positions-partial/

Trades

取引関連のAPIです。

参照元: https://gateway.docs.projectx.com/docs/category/trades/

Trade Search

API URL: POST https://gateway-api-demo.s2f.projectx.com/api/Trade/search

API リファレンス: /api/Trade/search

説明

リクエストパラメータから取引を検索します。

パラメータ

名前	型	説明	必須	null許容
accountId	integer	アカウントID。	必須	false
startTimestamp	datetime	タイムスタンプフィルターの開始。	必須	false
endTimestamp	datetime	タイムスタンプフィルターの終了。	オプション	true

使用例

リクエスト例

```
curl -X 'POST' \
  'https://gateway-api-demo.s2f.projectx.com/api/Trade/search' \
  -H 'accept: text/plain' \
  -H 'Content-Type: application/json' \
  -d '{
  "accountId": 203,
  "startTimestamp": "2025-01-20T15:47:39.882Z",
  "endTimestamp": "2025-01-30T15:47:39.882Z"
}'
```

レスポンス例

```
{
  "trades": [
    {
      "id": 8604,
      "accountId": 203,
      "contractId": "CON.F.US.EP.H25",
      "creationTimestamp": "2025-01-21T16:13:52.523293+00:00",
      "price": 6065.250000000,
      "profitAndLoss": 50.000000000,
      "fees": 1.4000,
      "side": 1,
      "size": 1,
      "voided": false,
      "orderId": 14328
    },
    {
      "id": 8603,
      "accountId": 203,
```

```
        "contractId": "CON.F.US.EP.H25",
        "creationTimestamp": "2025-01-21T16:13:04.142302+00:00",
        "price": 6064.250000000,
        "profitAndLoss": null,
        "fees": 1.4000,
        "side": 0,
        "size": 1,
        "voided": false,
        "orderId": 14326
      }
    ],
    "success": true,
    "errorCode": 0,
    "errorMessage": null
  }
```


注：null値のprofitAndLossはハーフターン取引を示します。

エラーの場合: response status is 401

参照元: https://gateway.docs.projectx.com/docs/api-reference/trade/trade-search/

Realtime Updates

様々なイベントのリアルタイム更新に関する情報です。

参照元: https://gateway.docs.projectx.com/docs/category/realtime-updates/

Real Time Data Overview

ProjectX Real Time APIは、SignalRライブラリ（WebSocket経由）を使用して、アカウント、注文、ポジション、残高、価格に関するデータ更新にリアルタイムでアクセスすることができます。

2つのハブがあります：userハブとmarketハブです。

- userハブはユーザーのアカウント、注文、ポジションのリアルタイム更新を提供します。
- marketハブは市場取引イベント、DOMイベントなどの市場データを提供します。

SignalRとは？

SignalRは、Microsoftによって開発されたリアルタイムWebアプリケーションフレームワークで、Webアプリケーションにリアルタイム機能を追加するプロセスを簡素化します。クライアント（Webブラウザなど）とサーバー間の双方向通信を可能にし、クライアント側の継続的なポーリングや接続の手動処理を必要とせずに、ライブチャット、通知、リアルタイム更新などの機能を実現します。

SignalRは、高レベルのAPIを提供することで、リアルタイム通信の複雑さを抽象化します。WebSocket、Server-Sent Events (SSE)、Long Pollingなどの様々なトランスポートプロトコルをサポートし、クライアントとサーバーの機能に基づいて最適なトランスポートメカニズムを自動的に選択します。

このフレームワークは、接続管理、メッセージルーティング、複数のサーバー間でのスケーリングを処理し、開発者がスケーラブルで応答性の高いWebアプリケーションを構築しやすくします。SignalRは.NETやJavaScriptなど複数のプラットフォームで利用可能であり、開発者は好みのプログラミング言語やフレームワークを使用してリアルタイムアプリケーションを構築することができます。

SignalRの詳細についてはこちらをご覧ください。

使用例

User Hub

```
// Import the necessary modules from @microsoft/signalr
const { HubConnectionBuilder, HttpTransportType } = require('@microsoft/signalr');

// Function to set up and start the SignalR connection
function setupSignalRConnection() {
  const JWT_TOKEN = 'your_bearer_token';
  const SELECTED_ACCOUNT_ID = 123; //your currently selected/visible account ID
  const userHubUrl = 'https://gateway-rtc-demo.s2f.projectx.com/hubs/user?
access_token=' + JWT_TOKEN;

  // Create the connection
  const rtcConnection = new HubConnectionBuilder()
    .withUrl(userHubUrl, {
      skipNegotiation: true,
      transport: HttpTransportType.WebSockets,
      accessTokenFactory: () => JWT_TOKEN, // Replace with your current JWT token
      timeout: 10000 // Optional timeout
    })
    .withAutomaticReconnect()
    .build();

  // Start the connection
  rtcConnection.start()
    .then(() => {
      // Function to subscribe to the necessary events
      const subscribe = () => {
        rtcConnection.invoke('SubscribeAccounts');
        rtcConnection.invoke('SubscribeOrders', SELECTED_ACCOUNT_ID); //you can
call this function multiple times with different account IDs
        rtcConnection.invoke('SubscribePositions', SELECTED_ACCOUNT_ID); //you can
call this function multiple times with different account IDs
        rtcConnection.invoke('SubscribeTrades', SELECTED_ACCOUNT_ID); //you can
call this function multiple times with different account IDs
      };

      // Functions to unsubscribe, if needed
      const unsubscribe = () => {
        rtcConnection.invoke('UnsubscribeAccounts');
        rtcConnection.invoke('UnsubscribeOrders', SELECTED_ACCOUNT_ID); //you can
call this function multiple times with different account IDs
        rtcConnection.invoke('UnsubscribePositions', SELECTED_ACCOUNT_ID); //you
```

```
can call this function multiple times with different account IDs
        rtcConnection.invoke('UnsubscribeTrades', SELECTED_ACCOUNT_ID); //you can
call this function multiple times with different account IDs
      };

      // Set up the event listeners
      rtcConnection.on('GatewayUserAccount', (data) => {
        console.log('Received account update', data);
      });
      rtcConnection.on('GatewayUserOrder', (data) => {
        console.log('Received order update', data);
      });
      rtcConnection.on('GatewayUserPosition', (data) => {
        console.log('Received position update', data);
      });
      rtcConnection.on('GatewayUserTrade', (data) => {
        console.log('Received trade update', data);
      });

      // Subscribe to the events
      subscribe();

      // Handle reconnection
      rtcConnection.onreconnected((connectionId) => {
        console.log('RTC Connection Reconnected');
        subscribe();
      });
    })
    .catch((err) => {
      console.error('Error starting connection:', err);
    });
}

// Call the function to set up and start the connection
setupSignalRConnection();
```

Market Hub

```
// Import the necessary modules from @microsoft/signalr
const { HubConnectionBuilder, HttpTransportType } = require('@microsoft/signalr');

// Function to set up and start the SignalR connection
function setupSignalRConnection() {
  const JWT_TOKEN = 'your_bearer_token';
  const marketHubUrl = 'https://gateway-rtc-demo.s2f.projectx.com/hubs/market?
access_token=' + JWT_TOKEN;
  const CONTRACT_ID = 'CON.F.US.RTY.H25'; // Example contract ID

  // Create the connection
  const rtcConnection = new HubConnectionBuilder()
    .withUrl(marketHubUrl, {
```

```
        skipNegotiation: true,
        transport: HttpTransportType.WebSockets,
        accessTokenFactory: () => JWT_TOKEN, // Replace with your current JWT token
        timeout: 10000 // Optional timeout
      })
      .withAutomaticReconnect()
      .build();

    // Start the connection
    rtcConnection.start()
      .then(() => {
        // Function to subscribe to the necessary events
        const subscribe = () => {
          rtcConnection.invoke('SubscribeContractQuotes', CONTRACT_ID);
          rtcConnection.invoke('SubscribeContractTrades', CONTRACT_ID);
          rtcConnection.invoke('SubscribeContractMarketDepth', CONTRACT_ID);
        };

        // Functions to unsubscribe, if needed
        const unsubscribe = () => {
          rtcConnection.invoke('UnsubscribeContractQuotes', CONTRACT_ID);
          rtcConnection.invoke('UnsubscribeContractTrades', CONTRACT_ID);
          rtcConnection.invoke('UnsubscribeContractMarketDepth', CONTRACT_ID);
        };

        // Set up the event listeners
        rtcConnection.on('GatewayQuote', (contractId, data) => {
          console.log('Received market quote data', data);
        });
        rtcConnection.on('GatewayTrade', (contractId, data) => {
          console.log('Received market trade data', data);
        });
        rtcConnection.on('GatewayDepth', (contractId, data) => {
          console.log('Received market depth data', data);
        });

        // Subscribe to the events
        subscribe();

        // Handle reconnection
        rtcConnection.onreconnected((connectionId) => {
          console.log('RTC Connection Reconnected');
          subscribe();
        });
      })
      .catch((err) => {
        console.error('Error starting connection:', err);
      });
}

// Call the function to set up and start the connection
setupSignalRConnection();
```

参照元: https://gateway.docs.projectx.com/docs/realtime/