

Mini Prediction Exchange

User Manual (Trader View)

1 Introduction

This guide explains how to interact with the mini prediction-market exchange API:

- create a user,
- create a market with YES/NO outcomes,
- place a BUY or SELL order in that market.

We assume the backend is already running locally:

```
uvicorn app.main:app --reload
```

The base URL is:

```
http://127.0.0.1:8000
```

You can use either the interactive docs at `/docs` or the `curl` examples below.

2 Creating a User

Endpoint

- **Method:** POST
- **Path:** `/users`

Request body

At minimum, provide an e-mail:

```
{
  "email": "alice@example.com"
}
```

Response

On success, you will get a JSON object like:

```
{
  "id": 1,
  "email": "alice@example.com",
  "created_at": "2025-11-24T20:00:00"
}
```

Note: Remember the `id`. This is your *user id* and will be used later in the `X-User-Id` header when placing orders.

3 Creating a Market

Endpoint

- **Method:** POST
- **Path:** `/markets/markets`

Request body

Example: a YES/NO market on rain in Mumbai.

```
{
  "slug": "rain-mumbai-2025-12-01",
  "title": "Will it rain in Mumbai on 1 Dec 2025?",
  "description": "Resolves to YES if IMD station at Mumbai records measurable rainfall on 2025-12-01",
  "status": "OPEN",
  "trading_close_at": "2025-11-30T23:59:00",
  "settle_at": "2025-12-02T12:00:00",
  "outcomes": [
    { "name": "Yes", "code": "YES", "sort_index": 0 },
    { "name": "No", "code": "NO", "sort_index": 1 }
  ]
}
```

Response

On success you will see a market object like:

```
{
  "id": 1,
  "slug": "rain-mumbai-2025-12-01",
  "title": "Will it rain in Mumbai on 1 Dec 2025?",
  "status": "OPEN",
  "trading_close_at": "2025-11-30T23:59:00",
  "settle_at": "2025-12-02T12:00:00",
  "outcomes": [
    { "id": 1, "name": "Yes", "code": "YES", "sort_index": 0 },
    { "id": 2, "name": "No", "code": "NO", "sort_index": 1 }
  ]
}
```

The important fields for placing orders are:

- slug (here: "rain-mumbai-2025-12-01"),
- outcome codes (here: "YES" and "NO"),
- status should be "OPEN".

4 Placing an Order

Once you have:

- a user id (e.g. 1),
 - an OPEN market (e.g. slug "rain-mumbai-2025-12-01"),
 - an outcome code (e.g. "YES"),
- you can place BUY or SELL orders.

Endpoint

- **Method:** POST
- **Path:** /orders/orders

Headers

- X-User-Id: your numeric user id (e.g. 1)
- Content-Type: application/json

Order inputs

market_id

Market slug, e.g. "rain-mumbai-2025-12-01".

outcome_id

Outcome code, e.g. "YES" or "NO".

side

"BUY" or "SELL".

type

"LIMIT" or "MARKET".

price

For LIMIT orders: price between 0 and 1 (contracts are priced in probability points). For MARKET orders this can be omitted or set to null.

quantity

Number of contracts.

currency

Settlement currency (currently only "INR" is supported).

Example: placing a LIMIT BUY order

This submits a LIMIT BUY of 10 YES contracts at price 0.55 INR per contract:

POST /orders/orders

Header: X-User-Id: 1

```
{
  "market_id": "rain-mumbai-2025-12-01",
  "outcome_id": "YES",
  "side": "BUY",
  "type": "LIMIT",
  "price": 0.55,
  "quantity": 10,
  "currency": "INR"
}
```

Successful response

On success you will get an OrderResponse:

```
{
  "order_id": "1",
  "trades": [
    // list of trades (may be empty if the order just rests in the book)
  ]
}
```

Notes:

- If there are matching opposing orders in the book, the engine will immediately generate trades and partially or fully fill your order.
- At the moment, user account balances are not yet debited or credited; the order book and trade history are the main focus.

5 Quick Reference

Endpoints

- POST /users – create user.
- POST /markets/markets – create market.
- GET /markets/markets – list markets.
- GET /markets/markets/{id} – get one market.
- POST /orders/orders – place order (requires X-User-Id).

Enums

- Order side: "BUY", "SELL".
- Order type: "LIMIT", "MARKET".
- Outcome code: typically "YES", "NO".
- Market status: "DRAFT", "OPEN", "SETTLED", "CANCELLED".

Summary

For traders, the basic workflow is:

1. Open `http://127.0.0.1:8000/docs#/docs` in your browser.
2. Create a user via `POST /users`.
3. Fund your account via `POST /accounts/{user_id}/fund`.
4. Ask the admin for available market IDs and outcome IDs, or view them via `GET /markets`.
5. Place orders via `POST /orders`.
6. Track your balances and orders via the `accounts` and `orders` endpoints.

As more features are added (matching engine, order cancellation, automatic market resolution, and external data feeds), this interface can continue to serve as a convenient way to interact with the exchange in real time.