Brain Mind and Markets Lab

THE UNIVERSITY OF MELBOURNE

FACULTY OF BUSINESS & ECONOMICS

FMClient Tutorial

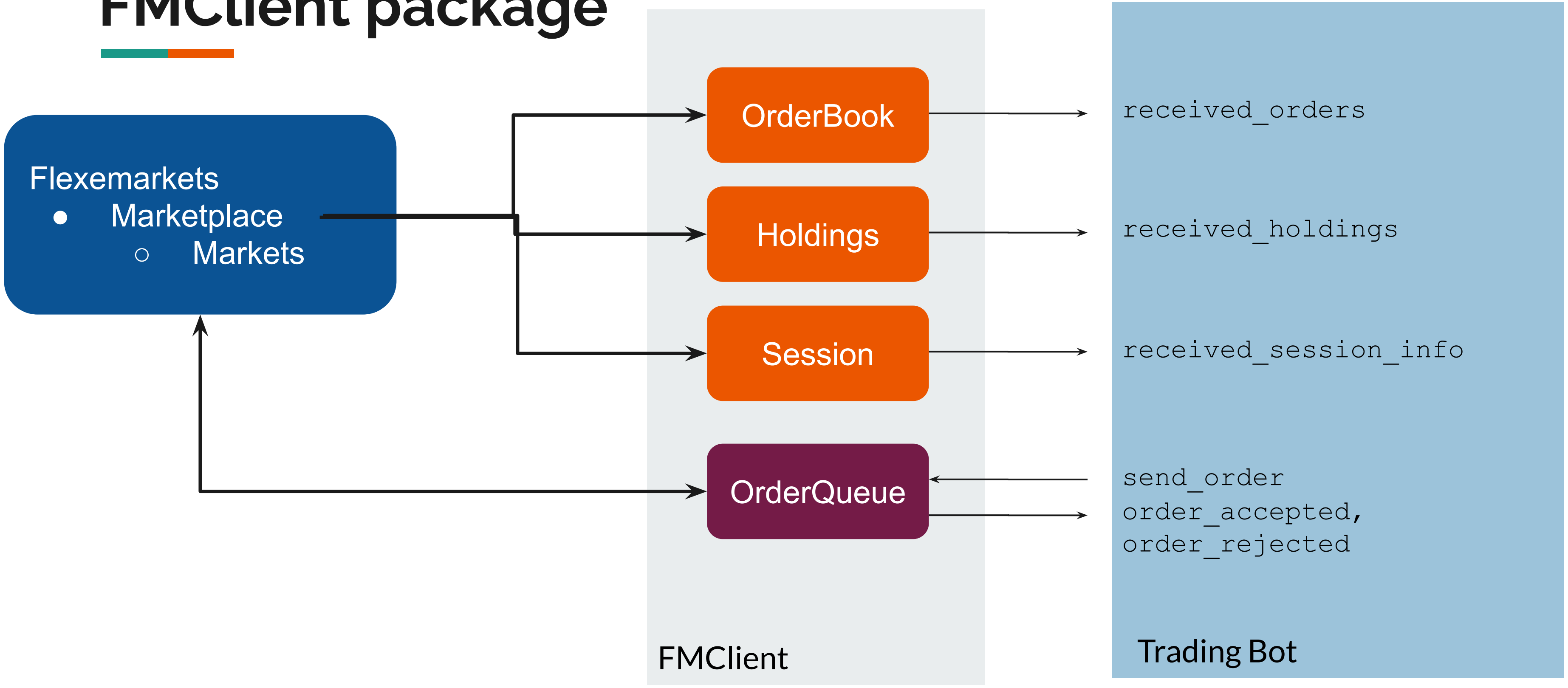
Let us build robots!

Nitin Yadav and Peter Bossaerts.

Overview

- FMClient Package
- Initialising
- Information processing
    - Session
    - Holdings
    - Order book
- Interacting
    - Sending Limit and Cancel orders
    - Order acceptance/rejection
- Algohost

FMClient package



Initialisation

- Using `initialised()`
    - Agent logs in to FM
    - Setup communication with Flexemarkets
    - Gets marketplace information
        - Markets to trade in
        - State of the marketplace (i.e., open or close)
    - Gets holdings information
- Method is automatically called by Agent class.

Markets

`self.markets`

- Dictionary with keys as market id, and values as market information.
- Example:

```
999: {'id': 999, 'minimum': 0, 'maximum': 1000, 'tick': 1, 'name':
     'Widget', 'item': 'widget', 'description': 'Widget',
     'privateMarket': False}
```

Market Id · Private or Public · Minimum price · Maximum price · Tick size

Note: There may be multiple markets in a marketplace.

Holdings

`self.holdings`

- An object with ‘cash’ and ‘assets’
- Cash
  - holdings.cash
  - holdings.cash_available
- Assets
  - A dictionary with Market as key and Asset as value
  - An ‘asset’ object has units, units_available, etc
- Updates received via `recieved_holdings`

Note: There may be multiple markets in a marketplace.

Order book

`recieved_orders(self, orders: List[order])`

- Called automatically when new orders/trades arrive
- Called *only* when marketplace is open
- `orders` is a list of orders
- Properties of order object:
  - `price`: The price each unit in cents
  - `units`: Number of units
  - `order_side`:`OrderSide.SELL or OrderSide.BUY`
  - `Order_type`:`OrderType.LIMIT or OrderType.CANCEL`
  - `date_created`: The date and time when the order was received at the exchange
  - `mine`: True if the order is from the same account as the agent, False otherwise.
  - `is_private`: True if the order is from a private market.
  - `owner_or_target`: Only valid for private orders. Tells the owner/target of the order.

Sending orders to public market

`self.send_order(order)`

```
order = Order.create_new()
new_order.market = Market(50)
new_order.price = 100
new_order.units = 1
new_order.order_side = OrderSide.SELL
new_order.order_type = OrderType.LIMIT
new_order.ref = “test order”

self.send_order(new_order)
```

Server response callback via `order_accepted` and `order_rejected`

Sending orders to private market

`self.send_order(order)`

```
order = Order.create_new()
new_order.market = Market(50)
new_order.price = 100
new_order.units = 1
new_order.order_side = OrderSide.SELL
new_order.order_type = OrderType.LIMIT
new_order.ref = “private order”
Self.owner_or_target = “M000”

self.send_order(new_order)
```

Server response callback via `order_accepted` and `order_rejected`

Cancelling orders

`self.send_order(order)`

```
cancel_order = copy.copy(order)
cancel_order.type = OrderType.CANCEL
cancel_order.ref = "o1_cancel"
self.send_order(cancel_order)
```

Check if the order is yours and hasn't traded yet!

Server response callback via `order_accepted` and `order_rejected`

Types of Bots

- Market Maker
  - Sends buy/sell orders and waits for their execution before refilling.
- Reactive
  - Reacts to profitable trade opportunities in the order book.

Tasks

1. Print your active orders in the market
2. Try different interactions with Flexemarkets UI
    a. Send an order via UI and check if your bot gets it
    b. Cancel an order via UI and check if the order is absent in the next order book update
    c. Send order via bot and cancel via the UI (and vice versa)
3. Check if your bot gets updated holdings when an order executes (i.e., it trades).
4. Check what happens if your bot sends invalid orders.
5. Introduce an error in your bot and observe what happens.