

AlgoTik TSE

Tehran Stock Exchange Data Library

کتابخانه جامع دریافت اطلاعات بازار بورس تهران

 Python Package

Stocks • Options • ETFs • Bonds • Funds • Currency

سهام • اختیار معامله • صندوق ETF • اوراق بدھی • صندوق‌ها • ارز و سکه

Version 1.0.0 – Python 3.8+

algotik.com

Author: Mohsen Alipour – alipour@algotik.ir

Telegram: @algotik

AlgoTik TSE

A comprehensive Python library for fetching market data from the Tehran Stock Exchange (TSETMC) and currency/coin prices (TGJU).

Supports stocks, options, ETFs, bonds, and treasury bills.

All outputs are returned as

Pandas DataFrames

with Jalali (Shamsi) date support.

فارسی IR

این کتابخانه جهت دریافت اطلاعات بازار بورس تهران و قیمت ارز و سکه توسعه یافته است. خروجی تمامی توابع با فرمت دیتافریم پانداز و با پشتیبانی از تاریخ شمسی ارائه می‌شود.

ویژگی‌ها:

دسترسی به داده‌ها با استفاده از نماد فارسی سهم

تعدیل قیمت خودکار (افزایش سرمایه + سود نقدی)

تشخیص هوشمند جابجایی نماد بین بازارها

دسترسی به همه شاخص‌های بازار (صنایع و کل)

قابلیت دانلود دسته‌جمعی سابقه قیمت

دریافت اطلاعات حقیقی/حقوقی

دریافت لیست سهامداران عمدۀ

دریافت سابقه افزایش سرمایه

دریافت قیمت ارز و سکه (دلار، یورو، سکه امامی و ...)

دریافت اطلاعات لحظه‌ای کل بازار در یک درخواست (Market Watch)

دریافت داده‌های اینترادی (کندل و تیک دقیقه‌ای، بازه‌ها: ۱ دقیقه تا ۱۲ ساعت)

لیست اختیار معامله‌ها با تجزیه خودکار (نوع، دارایی پایه، قیمت اعمال، سرسید)

دریافت زنجیره اختیار معامله با Open Interest

لیست صندوق‌های ETF با محاسبه تخفیف/حباب NAV

لیست اوراق مرابحه و خزانه با استخراج تاریخ سرسید

نام‌گذاری استاندارد (`*_get`) در کنار نام‌های اصلی

پشتیبانی از تاریخ شمسی، میلادی و نام روز هفته

- تنظیمات قابل پیکربندی: SSL, Timeout, Rate Limiting, Retry

مدیریت خودکار خطأ و Rate Limiting برای جلوگیری از بلاک شدن

🌐 وبسایت: algotik.com | 📺 تلگرام: t.me/algotik

Table of Contents

- [Installation](#)
 - [Quick Start](#)
 - [API Reference](#)
 - [get_history\(\)](#) – Historical price data
 - [get_client_type\(\)](#) – Retail / Institutional data
 - [get_capital_increase\(\)](#) – Capital increase history
 - [get_detail\(\)](#) – Full stock detail
 - [get_info\(\)](#) – Instrument information
 - [get_stats\(\)](#) – Instrument statistics
 - [get_symbols\(\)](#) – List all market symbols
 - [get_shareholders\(\)](#) – Major shareholders
 - [get_currency\(\)](#) – Currency & coin prices
 - [get_intraday\(\)](#) – Intraday tick & candle data
 - [get_market_snapshot\(\)](#) – Live market snapshot (all instruments)
 - [get_market_client_type\(\)](#) – Bulk individual/institutional data
 - [list_options\(\)](#) – List all active options
 - [get_options_chain\(\)](#) – Options chain with Open Interest
 - [list_etfs\(\)](#) – List ETFs with NAV discount
 - [list_bonds\(\)](#) – List bonds & treasury bills with maturity
 - [list_funds\(\)](#) – List all investment funds with NAV, returns & portfolio
 - [Legacy Aliases](#)
 - [Configuration](#)
 - [Examples](#)
 - [Market Screening](#) – Top volume, gainers & losers
 - [ETF Discount/Premium](#) – NAV arbitrage
 - [Currency & Gold](#) – Dollar, Euro, Gold Coin
 - [Options Overview](#) – Active options & top traded
 - [Fund Comparison](#) – Equity vs Fixed Income funds
 - [Bond Maturity](#) – Sukuk & treasury maturity
 - [Institutional Money Flow](#) – Net buying/selling
 - [All Asset Types](#) – Market instrument breakdown
 - [Intraday Candles](#) – 5min & 1h candles
 - [Stock Detail & Shareholders](#) – Company info
 - [Data Sources](#)
 - [License](#)
-

Installation

```
pip install algotik-tse
```

Upgrade to latest version:

```
pip install algotik-tse --upgrade
```

Requirements:

Python 3.6+ | pandas | requests | persiantools | lxml | numpy

Quick Start

شروع سریع – توضیحات فارسی

کد	توضیح
<code>att.get_history('شتران')</code>	دریافت سابقه قیمت تعديل شده سهم
<code>att.get_client_type('شتران')</code>	دریافت اطلاعات حقیقی/حقوقی
<code>att.get_symbols()</code>	لیست تمام نمادهای بازار (سهام، اوراق، اختیار، صندوق و ...)
<code>att.get_currency('dollar')</code>	قیمت دلار آمریکا
<code>att.get_intraday('شتران')</code>	کندهای ۱ دقیقه‌ای امروز
<code>att.get_market_snapshot()</code>	اطلاعات لحظه‌ای کل بازار
<code>att.list_options()</code>	لیست تمام اختیار معامله‌های فعال
<code>att.get_options_chain('اهرم')</code>	زنگیره اختیار معامله با Open Interest
<code>att.list_etfs()</code>	لیست صندوق‌های ETF با تخفیف/حباب NAV
<code>att.list_bonds()</code>	لیست اوراق بدهی (مراجعه، اجاره، خزانه) با سررسید
<code>att.list_funds()</code>	لیست صندوق‌های سرمایه‌گذاری با NAV، بازدهی و ترکیب پرتفوی

```
import algotik_tse as att

# Get adjusted stock price history
df = att.get_history('شتران', start='1404-06-01', end='1404-08-01')
print(df.head())
```

J-Date	Open	High	Low	Close	Volume
1404-06-01	2008	2028	1969	2020	58693215
1404-06-03	1995	2011	1932	1932	56282643
1404-06-04	1888	1944	1888	1912	128242492
1404-06-05	1889	1965	1885	1897	80085551
1404-06-08	1875	1898	1875	1897	161293403

```
# Get retail/institutional data
df_ri = att.get_client_type('شتراں', limit=100)

# List all stocks in the market
all_stocks = att.get_symbols()

# Get US Dollar price history
usd = att.get_currency('dollar', limit=365)

# Intraday 1-minute candles (today's data)
intraday = att.get_intraday('شتراں', interval='1min')

# Historical intraday (multi-day)
hist = att.get_intraday('شتراں', interval='5min',
                        start='1404-11-01', end='1404-11-06')

# Live market data for ALL instruments in one call
data = att.get_market_snapshot()
print(data['stocks'].shape)                                # DataFrame of all instruments
print(data['market_time'])                               # '04/11/29 15:04:05'
print(data['index_value'])                             # 3806743.94

# Options chain for a specific underlying
chain = att.get_options_chain('اهرم')
print(chain['calls'].head())                            # Calls DataFrame
print(chain['underlying_price'])                      # Current underlying price

# List all ETFs with NAV discount
etfs = att.list_etfs()
print(etfs[['Symbol', 'Close', 'NAV', 'NAV_Discount']].head())

# List all bonds with maturity info
bonds = att.list_bonds()
print(bonds[['Symbol', 'Ticker', 'BondType', 'MaturityJalali', 'DaysToMaturity']].head())

# Investment funds - NAV, returns, portfolio composition
funds = att.list_funds()
equity_funds = att.list_funds(fund_type='equity')
```

API Reference

get_history()

Get historical price data for one or more symbols. Prices are

auto-adjusted

for splits & dividends by default.

توضیحات فارسی –

تابع `()get_history` برای دریافت سابقه قیمت سهام از سایت TSETMC استفاده می‌شود. قیمت‌ها به صورت پیش‌فرض تعدیل شده (برای افزایش سرمایه و سود نقدی) ارائه می‌شوند.

پارامترها:

پارامتر	نوع	پیش‌فرض	توضیح
<code>symbol</code>	<code>str</code> یا <code>list</code>	–	نماد فارسی سهم (مثلاً ' <code>شتران</code> ') یا لیست نمادها
<code>start</code>	<code>str</code>	<code>None</code>	تاریخ شروع شمسی (مثلاً ' <code>01-01-1402</code> ')
<code>end</code>	<code>str</code>	<code>None</code>	تاریخ پایان شمسی
<code>limit</code>	<code>int</code>	<code>0</code>	تعداد آخرین روزهای معاملاتی (<code>0</code> = کل تاریخچه)
<code>auto_adjust</code>	<code>bool</code>	<code>True</code>	تعديل خودکار قيمت (افزايش سرمایه + سود نقدی)
<code>output_type</code>	<code>str</code>	<code>'standard'</code>	(فقط OHLCV) یا <code>'full'</code> (همه ستون‌ها)
<code>date_format</code>	<code>str</code>	<code>'jalali'</code>	(میلادی)، یا <code>'both'</code> (هر دو) (شمسی)، <code>'gregorian'</code> (<code>'jalali'</code>)
<code>raw</code>	<code>bool</code>	<code>False</code>	فرمت TSETMC برای وارد کردن در نرم‌افزارهای معاملاتی
<code>return_type</code>	<code>str/list</code>	<code>None</code>	محاسبه بازده: <code>'simple'</code> , <code>'log'</code> , <code>'both'</code> [<code>simple</code> , <code>'Close'</code> , <code>5</code>]
<code>save_to_file</code>	<code>bool</code>	<code>False</code>	ذخیره نتیجه در فایل CSV
<code>adjust_volume</code>	<code>bool</code>	<code>False</code>	تعديل حجم معاملات برای افزایش سرمایه
<code>dropna</code>	<code>bool</code>	<code>True</code>	حذف ستون‌های اضافی در حالت چند نمادی
<code>ascending</code>	<code>bool</code>	<code>True</code>	مرتب‌سازی صعودی (<code>True</code>) یا نزولی (<code>False</code>) بر اساس تاریخ
<code>save_path</code>	<code>str</code>	<code>None</code>	مسیر فایل CSV برای ذخیره (مثلاً ' <code>'output.csv'</code> ')
<code>progress</code>	<code>bool</code>	<code>True</code>	نمایش نوار پیشرفت

خروجی‌های مختلف:

حالات عادی (`standard`): ۵ ستون – `Open` (باز)، `High`، `Low` (بیشترین)، `Close` (پایانی)، `Volume` (حجم) – همه `int64`

حالات کامل (`full`): ۱۰ ستون – علاوه بر موارد بالا `Final` (قیمت پایانی میانگین وزنی)، `No.` (تعداد معاملات)، `Value` (ارزش معاملات ریالی)، `Ticker` (نام روز هفته فارسی)، `Weekday_fa` (نماد)

بدون تعديل (`auto_adjust=False`): ستون `Adj Close` (قیمت تعديل شده) اضافه می‌شود و قیمت‌های OHLC خام (بدون تعديل) هستند

فرمت TSE: نام ستون‌ها مطابق TSETMC مثل `<TICKER>`, `<HIGH>`, `<CLOSE>` و ...

تاریخ میلادی: ایندکس `datetime64` از نوع `Date` به جای رشته شمسی

بازده: ستون `returns` اضافه می‌شود – ساده، لگاریتمی، یا هر دو

چند نمادی: ستون‌ها `Column` و `Symbol` می‌شوند: (`Column, Symbol`)

نکات مهم: - برای دریافت شاخص‌کل یا شاخص‌های صنایع، نام شاخص را به عنوان نماد وارد کنید (مثلاً '`شاخص کل`', '`شاخص صنعت فلزات`

`اساس`) - با `save_to_file=True` خروجی به صورت فایل CSV ذخیره می‌شود - در حالت چند نمادی، فقط روزهای مشترک معاملاتی بین نمادها

نمایش داده می‌شود

```

att.get_history(
    symbol='شتران',           # str or list - symbol name(s) in Persian
    start=None,                # str - start date in Jalali 'YYYY-MM-DD' (e.g. '1402-01-01')
    end=None,                  # str - end date in Jalali 'YYYY-MM-DD'
    limit=0,                   # int - number of last trading days (0 = all history)
    raw=False,                 # bool - use TSETMC column names
    auto_adjust=True,          # bool - adjust for splits & dividends
    output_type='standard',    # str - 'standard' (OHLCV) or 'full' (all columns)
    date_format='jalali',      # str - 'jalali', 'gregorian', or 'both'
    progress=True,              # bool - show download progress bar
    save_to_file=False,         # bool - save result to CSV file
    dropna=True,                # bool - drop extra columns in multi-stock mode
    adjust_volume=False,        # bool - adjust volume for capital increases
    return_type=None,           # str/list - 'simple', 'log', 'both', or ['simple','Close',5]
    ascending=True,              # bool - sort by date ascending (True) or descending (False)
    save_path=None,             # str - file path to save CSV (e.g. 'output.csv')
)

```

Standard output (default)

```
df = att.get_history('شتران', limit=10)
```

	Open	High	Low	Close	Volume
J-Date					
1404-10-14	3960	3996	3810	3996	1272346113
1404-10-15	4098	4098	4098	4098	450168956
1404-10-16	4220	4220	4220	4220	326395132
1404-10-17	4346	4346	4346	4346	892210289
1404-10-20	4216	4476	4216	4218	1862610980

- Index:** `J-Date` (Jalali string, e.g. `1404-10-14`)
- Columns:** `Open`, `High`, `Low`, `Close`, `Volume` – all `int64`

Full output

```
df = att.get_history('شتران', limit=5, output_type='full')
```

	Open	High	Low	Close	Final	Volume	No.	Value	Weekday_fa	Ticker
J-Date										
1404-11-25	4400	4475	4218	4218	4308	238550890	4846	1027754207785	شنبه	شتران
1404-11-26	4179	4179	4179	4179	4179	39453982	748	164878190778	یکشنبه	شتران
1404-11-27	4054	4109	4054	4064	4056	430020598	7394	1744313323314	دوشنبه	شتران
1404-11-28	4010	4100	3958	4066	4037	164209800	4199	662877646216	سه شنبه	شتران

Column	Description
<code>Open, High, Low, Close</code>	Adjusted OHLC prices (int)
<code>Final</code>	Weighted average closing price — قیمت پایانی
<code>Volume</code>	Trade volume
<code>No.</code>	Number of trades
<code>Value</code>	Total trade value (Rials)
<code>Weekday_fa</code>	Day of week in Persian (شنبه, یکشنبه, ...)
<code>Ticker</code>	Symbol name

Gregorian dates

```
df = att.get_history('شتران', limit=5, date_format='gregorian')
```

Date	Open	High	Low	Close	Volume
2026-02-14	4400	4475	4218	4218	238550890
2026-02-15	4179	4179	4179	4179	39453982
2026-02-16	4054	4109	4054	4064	430020598
2026-02-17	4010	4100	3958	4066	164209800

- **Index:** `Date` (`datetime64`)
- Use `date_format='both'` to get both Jalali & Gregorian columns.
- Full mode with Gregorian shows `Weekday` (Monday, Tuesday, ...) instead of `Weekday_fa` .

Auto-adjust off

```
df = att.get_history('شتران', limit=5, auto_adjust=False)
```

J-Date	Open	High	Low	Close	Adj Close	Volume
1404-11-25	4400.0	4475.0	4218.0	4218.0	4218	238550890
1404-11-26	4179.0	4179.0	4179.0	4179.0	4179	39453982
1404-11-27	4054.0	4109.0	4054.0	4064.0	4064	430020598
1404-11-28	4010.0	4100.0	3958.0	4066.0	4066	164209800

- Adds `Adj Close` column. OHLC are raw (unadjusted) and `float64` .

TSE format

```
df = att.get_history('شتران', limit=3, raw=True)
```

	<TICKER>	<FIRST>	<HIGH>	<LOW>	<CLOSE>	<VALUE>	<VOL>	<OPENINT>	<PER>
<OPEN>	<LAST>								
<DTYYYYMMDD>									
2026-02-15	Palayesh.Tehran	4179.0	4179.0	4179.0	4179.0	164878190778	39453982	748	D
4308.0	4179.0								
2026-02-16	Palayesh.Tehran	4054.0	4109.0	4054.0	4056.0	1744313323314	430020598	7394	D
4179.0	4064.0								
2026-02-17	Palayesh.Tehran	4010.0	4100.0	3958.0	4037.0	662877646216	164209800	4199	D
4056.0	4066.0								

- TSETMC-compatible column names for import into trading software.

Return calculation

```
# Simple 1-day returns
df = att.get_history('شتران', limit=10, return_type='simple')
# Adds 'returns' column: (Close[t] - Close[t-1]) / Close[t-1]

# Log returns
df = att.get_history('شتران', limit=10, return_type='log')
# Adds 'returns' column: ln(Close[t] / Close[t-1])

# Both simple & log returns
df = att.get_history('شتران', limit=10, return_type='both')
# Adds 'simple_returns' and 'log_returns' columns

# Custom: simple 5-day returns on Close
df = att.get_history('شتران', limit=15, return_type=['simple', 'Close', 5])
```

	Open	High	Low	Close	Volume	returns
J-Date						
1404-11-06	4490	4490	4490	4490	16195770	NaN
1404-11-07	4356	4356	4356	4356	276970553	-0.029844
1404-11-08	4226	4356	4226	4259	1731947316	-0.022268
1404-11-11	4240	4330	4110	4110	379107775	-0.034985
1404-11-12	4110	4278	4087	4278	700763528	0.040876

Multi-stock

```
df = att.get_history(['شتران', 'فملی'], limit=5)
```

	Open	High	Low	Close	Volume	Open	High	Low	Close	Volume
J-Date	شتران	شتران	شتران	شتران	شتران	شتران	شتران	شتران	شتران	شتران
1404-11-25	4400	4475	4218	4218	238550890	14890	15080	14310	14310	306133075
1404-11-26	4179	4179	4179	4179	39453982	14020	14100	14020	14020	185179129
1404-11-27	4054	4109	4054	4064	430020598	13600	13970	13600	13900	214659584
1404-11-28	4010	4100	3958	4066	164209800	14030	14120	13790	14030	139758819

- Returns a `MultiIndex` column structure: `(Column, Symbol)`.

Index support

```
# کل شاخص (Total Market Index)
idx = att.get_history('کل', 'شاخص', limit=10)

# Industry indices
idx = att.get_history('اساسی', 'شاخص صنعت فلزات', limit=10)
```

	Open	High	Low	Close	Volume
J-Date					
1404-11-25	4081300.0	4090060.0	3986100.0	3986106.0	2.184455e+10
1404-11-26	3898000.0	3898000.0	3881860.0	3881867.0	2.381066e+10
1404-11-27	3800290.0	3822580.0	3799820.0	3822568.0	2.270925e+10

get_client_type()

Get historical

Retail / Institutional

(حقیقی / حقوقی) trading data.

توضیحات فارسی - ()get_client_type

تابع ()get_client_type برای دریافت اطلاعات معامله‌گران حقیقی و حقوقی استفاده می‌شود.

خروجی عادی (۱۲ ستون):

	ستون	توضیح
N_buy_retail		تعداد معاملات خرید حقیقی
N_buy_institutional		تعداد معاملات خرید حقوقی
N_sell_retail		تعداد معاملات فروش حقیقی
N_sell_institutional		تعداد معاملات فروش حقوقی
Vol_buy_retail		حجم خرید حقیقی
Vol_buy_institutional		حجم خرید حقوقی
Vol_sell_retail		حجم فروش حقیقی
Vol_sell_institutional		حجم فروش حقوقی
Val_buy_retail		ارزش خرید حقیقی (ریالی)
Val_buy_institutional		ارزش خرید حقوقی (ریالی)
Val_sell_retail		ارزش فروش حقیقی (ریالی)
Val_sell_institutional		ارزش فروش حقوقی (ریالی)

خروجی کامل (۲۰ ستون):

علاوه بر ۱۲ ستون بالا:

توضیح	ستون اضافی
سرانه خرید حقیقی	<code>Per_capita_buy_retail</code>
سرانه فروش حقیقی	<code>Per_capita_sell_retail</code>
سرانه خرید حقوقی	<code>Per_capita_buy_institutional</code>
سرانه فروش حقوقی	<code>Per_capita_sell_institutional</code>
قدرت خریدار به فروشنده حقیقی	<code>Power_retail</code>
قدرت خریدار به فروشنده حقوقی	<code>Power_institutional</code>
نام روز هفته فارسی	<code>Weekday_fa</code>
نماد	<code>Ticker</code>

```
att.get_client_type(
    symbol='شتران',
    start=None,
    end=None,
    limit=0,
    raw=False,
    output_type='standard',
    date_format='jalali',
    progress=True,
    save_to_file=False,
    dropna=True,
    ascending=True,
    save_path=None,
)

```

Standard output (12 columns)

```
df = att.get_client_type('شتران', limit=5)
```

	N_buy_retail	N_buy_institutional	N_sell_retail	N_sell_institutional	Vol_buy_retail	Vol_buy_institutional	Vol_sell_retail	Vol_sell_institutional
J-Date								
1404-11-25	1499		12	883		10	95906966	
142643924	216933695		21617195	414366661677		613387546108	935110119463	
92644088322								
1404-11-26	531		3	47		4	14403982	
25050000	28630968		10823014	60194240778		104683950000	119648815272	
45229375506								
1404-11-27	2465		10	1969		27	319021634	
110998964	277392757		152627841	1294256635847		450056687467	1125177802472	
619135520842								
1404-11-28	1260		11	1171		9	112375538	
51834262	156350833		7858967	453981687814		208895958402	631059614604	
31818031612								

Column	Description
N_buy_retail	Number of individual (حقیقی) buy trades
N_buy_institutional	Number of institutional (حقوقی) buy trades
N_sell_retail	Number of individual sell trades
N_sell_institutional	Number of institutional sell trades
Vol_buy_retail	Individual buy volume
Vol_buy_institutional	Institutional buy volume
Vol_sell_retail	Individual sell volume
Vol_sell_institutional	Institutional sell volume
Val_buy_retail	Individual buy value (Rials)
Val_buy_institutional	Institutional buy value (Rials)
Val_sell_retail	Individual sell value (Rials)
Val_sell_institutional	Institutional sell value (Rials)

Full output (20 columns)

```
df = att.get_client_type('شتران', limit=5, output_type='full')
```

Adds 8 extra columns to the standard 12:

Extra Column	Description
Per_capita_buy_retail	Average buy value per individual trade
Per_capita_sell_retail	Average sell value per individual trade
Per_capita_buy_institutional	Average buy value per institutional trade
Per_capita_sell_institutional	Average sell value per institutional trade
Power_retail	Individual buyer/seller power ratio
Power_institutional	Institutional buyer/seller power ratio
Weekday_fa	Day name in Persian
Ticker	Symbol name

Date range & Gregorian

```
# Jalali date range
df = att.get_client_type('شتران', start='1404-06-01', end='1404-08-01')

# Gregorian index
df = att.get_client_type('شتران', limit=10, date_format='gregorian')
# Index: 'Date' (datetime64)
```

get_capital_increase()

Get the full history of capital increases for a stock.

توضیحات فارسی – ()get_capital_increase

تابع ()get_capital_increase ساقه کامل افزایش سرمایه یک نماد را برمی‌گرداند.

توضیح	ستون
تعداد سهام قبل از افزایش سرمایه	old_shares_amount
تعداد سهام بعد از افزایش سرمایه	new_shares_amount

ایندکس: تاریخ میلادی (datetime64)

داده‌ها از قدیمی‌ترین به جدیدترین مرتب شده‌اند

```
df = att.get_capital_increase('شتران')
```

	old_shares_amount	new_shares_amount
date		
2025-03-02	3.900000e+11	5.395000e+11
2024-02-17	2.750000e+11	3.900000e+11
2022-11-02	1.700000e+11	2.750000e+11
2021-10-17	7.500000e+10	1.700000e+11
2020-10-04	4.400000e+10	7.500000e+10
2019-08-07	2.400000e+10	4.400000e+10
2018-07-24	1.600000e+10	2.400000e+10
2017-02-04	1.200000e+10	1.600000e+10

- Index: date (datetime64) – Gregorian)
- Columns: old_shares_amount , new_shares_amount

get_detail()

Get comprehensive detail for a stock (ISIN, company name, market, sector, etc.).

توضیحات فارسی – ()get_detail

تابع ()get_detail اطلاعات جامع نماد شامل کد ISIN، نام شرکت، نام لاتین، بازار، کد تابلو و سایر مشخصات را برمی‌گرداند.

خروجی: دیتاframes با ۱۵ ردیف (کلید-مقدار)

ایندکس: نام فیلد به فارسی (مثلًا `کد 12 رقمی نماد`، `نماد فارسی`، `بازار`)

ستون: — مقدار هر فیلد

```
df = att.get_detail('شتران')
```

key	value
کد 12 رقمی نماد	IR01PTEH0001
کد 5 رقمی نماد	PTEH1
نام لاتین شرکت	Palayesh Tehran
کد 4 رقمی شرکت	PTEH
پالایش نفت تهران	نام شرکت
شتران	نماد فارسی
پالایش نفت تهران	نماد 30 رقمی فارسی
کد 12 رقمی شرکت	IR01PTEH0007
بازار اول (تابلوی اصلی) بورس	بازار
1	کد تابلو

- **Shape:** (15, 1) – 15 key-value rows
- **Index:** `key` (str) – Persian field names
- **Column:** `value`

`get_info()`

Get instrument information (EPS, sector PE, PSR, sector name, threshold data, etc.).

تابع `get_info()` توضیحات فارسی —

اطلاعات ابزار مالی شامل EPS تخمینی، P/E، PSR، نام گروه صنعت، و اطلاعات آستانه قیمتی را برمی‌گرداند.

خروجی: دیتافریم با ۴۶ ردیف (کلید-مقدار)

(`eps_estimatedEPS` ، `eps_sectorPE` ، `sector_lSecVal`)

ایندکس: شناسه فیلد (مثلًا `eps_estimatedEPS` ، `sector_lSecVal`)

ستون: — مقدار هر فیلد

```
df = att.get_info('شتران')
```

key	value
<code>eps_estimatedEPS</code>	1018
<code>eps_sectorPE</code>	4.58
<code>eps_psr</code>	5933.701
<code>sector_cSecVal</code>	23
<code>sector_lSecVal</code>	فراورده های نفتی، کک و سوخت هسته ای

- **Shape:** (46, 1) – 46 key-value rows
- **Index:** `key` (str) – field identifiers (e.g. `eps_estimatedEPS` , `sector_lSecVal`)
- **Column:** `value`

get_stats()

Get trading statistics for a stock (averages, rankings over 3 and 12 months).

توضیحات فارسی – ()get_stats

تابع ()get_stats آمار معاملاتی نماد شامل میانگین و رتبه ارزش، حجم و تعداد معاملات در بازه‌های ۳ ماهه و ۱۲ ماهه را برمی‌گرداند. خروجی: دیتاframes با ۸۸ ردیف (کلید-مقدار) ایندکس: نام آماره به فارسی (مثلًا میانگین ارزش معاملات در ۳ ماه گذشته) ستون: مقدار عددی هر آماره شامل: رتبه‌بندی نماد از نظر حجم، ارزش و دفعات معاملات نسبت به کل بازار

```
df = att.get_stats('شتران')
```

key	value
2.443327	میانگین ارزش معاملات در ۳ ماه گذشته
1.461746	میانگین ارزش معاملات در ۱۲ ماه گذشته
4.500000	رتبه ارزش معاملات در ۳ ماه گذشته
5.200000	رتبه ارزش معاملات در ۱۲ ماه گذشته
6.053144	میانگین حجم معاملات در ۳ ماه گذشته
4.740798	میانگین حجم معاملات در ۱۲ ماه گذشته
1.200000	رتبه حجم معاملات در ۳ ماه گذشته
1.100000	رتبه حجم معاملات در ۱۲ ماه گذشته
8.543000	میانگین دفعات معاملات روزانه در ۳ ماه گذشته
6.474000	میانگین دفعات معاملات روزانه در ۱۲ ماه گذشته

- Shape:** (88, 1) – 88 key-value rows
- Index:** `key` (str) – Persian statistic names
- Column:** `value`

get_shareholders()

Get major shareholders of a stock (current or historical).

توضیحات فارسی – ()get_shareholders

تابع ()get_shareholders لیست سهامداران عمده یک نماد را برمی‌گرداند – هم فعلی و هم تاریخی.

پارامترها:

پارامتر	توضیح
<code>symbol</code>	نماد فارسی سهم
<code>date</code>	تاریخ شمسی به فرمت YYYYMMDD برای دریافت سهامداران در تاریخ خاص (<code>None</code> = آخرین اطلاعات)
<code>include_id</code>	اگر <code>True</code> باشد، شناسه سهامدار (<code>share_holder_id</code>) نیز اضافه می‌شود

ستون‌های خروجی:

توضیح	ستون
نام سهامدار	<code>share_holder_name</code>
تعداد سهام	<code>number_of_shares</code>
درصد مالکیت	<code>percentage_of_shares</code>
وضعیت تغییر (1 = بدون تغییر, 3 = تغییر یافته)	<code>change_state</code>
مقدار تغییر	<code>change_amount</code>
تاریخ ثبت (میلادی YYYYMMDD)	<code>date</code>
شناسه عددی سهامدار (فقط با <code>include_id=True</code> با)	<code>share_holder_id</code>

```
att.get_shareholders(
    symbol='شتران',      # str - symbol name in Persian
    date=None,             # str - Jalali date 'YYYYMMDD' for historical data (None = latest)
    include_id=False,       # bool - include shareholder IDs
)
```

Current shareholders

```
df = att.get_shareholders('شتران')
```

			share_holder_name	number_of_shares	percentage_of_shares	
change_state	change_amount	date				
0	0.0	20260218	3.234498	بانک صادرات ایران	5.995	
1	2.569312	-	شرکت سرمایه گذاری ایرانیان - سهامی خاص	e+10	4.762	
1	0.0	20260218	-	شرکت سرمایه گذاری ا.ا.تهران - سهامی عام	e+10	4.021
2	2.169540	-	شرکت .س. سهام عدالت .ا.خراسان رضوی - سع - م ک م ف ع	e+10	3.879	
1	0.0	20260218	-	شرکت .س. سهام عدالت .ا.خراسان رضوی - سع - م ک م ف ع	e+10	3.331
3	2.092901	-	PRX1.797408	موسسه شرکه 76894-سبد-	e+10	
4	0.0	20260218	-33322			
1	0.0	20260218				

Column	Description
<code>share_holder_name</code>	Shareholder name
<code>number_of_shares</code>	Number of shares held
<code>percentage_of_shares</code>	Ownership percentage
<code>change_state</code>	Change indicator (1=unchanged, 3=changed)
<code>change_amount</code>	Amount of change
<code>date</code>	Date of record (YYYYMMDD)

Historical shareholders

```
df = att.get_shareholders('داتام', date='14021006')
```

	change_amount	date	share_holder_name	number_of_shares	percentage_of_shares	change_state
0	6.732833	20231230	- شرکت توسعه تجارت داتام - سهامی خاص	-6e+09	67.32	3
1	BFM1.500000		صندوق سرمایه گذاری آ.ب . افتخار حافظ	-1e+09	15.00	0
	6.232833e+09					

With shareholder IDs

```
df = att.get_shareholders('شتران', include_id=True)
# Adds 'share_holder_id' column (7 columns total)
```

get_symbols()

Get a list of all symbols in Tehran Stock Exchange markets — including stocks, ETFs, bonds, options, and more.

توضیحات فارسی –

تابع `(get_symbols)` فهرست تمام نمادهای بازار سرمایه را برمی‌گرداند. علاوه بر سهام و حق‌تقدم و صندوق‌ها، اکنون می‌توانید اوراق بدھی، اختیار معامله، تسهیلات مسکن، گواهی‌های کالایی و گواهی‌های انرژی را هم دریافت کنید.

پارامترهای فیلتر بازار (سهام):

توضیح	مقدار پیش‌فرض	پارامتر
شامل نمادهای بورس	True	bourse / main_market
شامل نمادهای فرابورس (شامل نوآفرین)	True	farabourse / otc
شامل نمادهای بازار پایه	True	payeh / base_market
فیلتر رنگ بازار پایه: 'زرد', 'نارنجی', 'قرمز'	None	payeh_color / base_market_tier

پارامترهای نوع دارایی:

توضیح	مقدار پیش‌فرض	پارامتر
شامل نمادهای حق تقدیر	<code>False</code>	<code>haghe_taqadom / rights</code>
شامل صندوق‌های ETF و سرمایه‌گذاری	<code>False</code>	<code>sandogh / funds</code>
شامل اوراق بدهی: اخزا، اراد، صکوک، استناد شهری	<code>False</code>	<code>bonds</code>
شامل اختیار معامله: خرید و فروش سهام و صندوق	<code>False</code>	<code>options</code>
شامل تسهیلات مسکن	<code>False</code>	<code>mortgage</code>
شامل گواهی‌های کالایی: گواهی سپرده، زعفران	<code>False</code>	<code>commodity</code>
شامل گواهی‌های انرژی: گواهی ظرفیت برق	<code>False</code>	<code>energy</code>
فرمت خروجی: 'list' یا 'dataframe'	<code>'dataframe'</code>	<code>output</code>

:('output='dataframe' (هنگام ستون‌های خروجی)

توضیح	ستون
نماد (ایندهکس DataFrame)	<code>symbol</code>
نام کامل فارسی	<code>name</code>
کد ISIN	<code>instrument_isin</code>
نام انگلیسی	<code>english_name</code>
کد ۴ رقمی شرکت	<code>company_code</code>
شرکت ISIN	<code>company_isin</code>
نوع بازار	<code>market</code>
گروه صنعت	<code>industry_group</code>
نوع دارایی:	<code>asset_type</code>
شناسه عددی نماد	<code>instrument_id</code>

مثال‌های فیلتر:

```

        → فقط سهام (پیش‌فرض) → att.get_symbols()
        → سهام + اوراق بدهی → att.get_symbols(bonds=True)
        → فقط اختیار معامله → att.get_symbols(bourse=False, farabourse=False, payeh=False, options=True)
        → سهام + صندوق + اوراق + اختیار → att.get_symbols(sandogh=True, bonds=True, options=True)
        → خروجی به صورت لیست نمادها → att.get_symbols(output='list')
    
```

```

att.get_symbols(
    bourse=True,           # bool - include Bourse stocks (alias: main_market)
    farabourse=True,       # bool - include Fara Bourse stocks (alias: otc)
    payeh=True,            # bool - include Payeh market stocks (alias: base_market)
    haghe_taqadom=False,   # bool - include subscription rights (alias: rights)
    sandogh=False,          # bool - include ETFs/funds (alias: funds)
    bonds=False,            # bool - include bonds, sukuk, treasury bills
    options=False,          # bool - include stock & fund options (calls + puts)
    mortgage=False,         # bool - include housing facility certificates
    commodity=False,        # bool - include commodity certificates
    energy=False,            # bool - include energy certificates
    payeh_color=None,        # str or list - filter Payeh by tier (alias: base_market_tier)
    output='dataframe',      # str - 'dataframe' or 'list'
    progress=True,           # bool - show progress messages
)

```

Default: all regular stocks

```
df = att.get_symbols()
```

market	industry_group	asset_type	name	instrument_isin	english_name	company_code	company_isin
symbol				instrument_id			
بورس ایران	آباد	توریستی و رفاهی آبادگران ایران	آباد	IRO1ABAD0001	Abadgaran	ABAD	IRO1ABAD0002
بورس ایران	هتل و رستوران	بازار دوم بورس	بازار دوم	stock 59612098290740355			
دکتر عبیدی	لبراتوار داروسازی	دکتر عبیدی	دعبید	IRO1ABDI0001	Dr. Abidi Lab.	ABDI	IRO1ABDI0004
سیمان آبیک	دارویی	بازار دوم بورس مواد و محصولات دارویی سیمان آبیک	سایپاک	stock 49054891736433700	Abiak Cement	ABIK	IRO1ABIK0005
سیمان، آهک و گچ	اول (تابلوی اصلی) بورس	اول (تابلوی اصلی) بورس		stock 70883594945615893			بازار

- Shape:** DataFrame of all symbols in selected mode
- Index:** `symbol` (str) – Persian ticker symbol

Column	Description
<code>name</code>	Full company/instrument name
<code>instrument_isin</code>	ISIN code (e.g. <code>IRO1ABAD0001</code>)
<code>english_name</code>	English name
<code>company_code</code>	4-character company code
<code>company_isin</code>	Company-level ISIN
<code>market</code>	(بورس / فرابورس / پایه) Market name
<code>industry_group</code>	Industry sector name
<code>asset_type</code>	Asset class: <code>stock</code> , <code>right</code> , <code>fund</code> , <code>bond</code> , <code>option</code> , <code>mortgage</code> , <code>commodity</code> , <code>energy</code>
<code>instrument_id</code>	Numeric instrument identifier

Filter by market

```
# Bourse only
att.get_symbols(farabourse=False, payeh=False)

# Fara Bourse only
att.get_symbols(bourse=False, payeh=False)

# Payeh market only
att.get_symbols(bourse=False, farabourse=False)
```

Filter Payeh by color (تبلو)

```
# زرد (yellow) only
att.get_symbols(bourse=False, farabourse=False, payeh_color='زرد')

# نارنجی (orange) only
att.get_symbols(bourse=False, farabourse=False, payeh_color='نارنجی')

# قرمز (red) only
att.get_symbols(bourse=False, farabourse=False, payeh_color='قرمز')
```

Include additional asset types

```
# With subscription rights
att.get_symbols(haghe_taqadom=True)

# With ETFs/funds
att.get_symbols(sandogh=True)

# Bonds only (no stocks)
att.get_symbols(bourse=False, farabourse=False, payeh=False, bonds=True)

# Options only (no stocks)
att.get_symbols(bourse=False, farabourse=False, payeh=False, options=True)

# All asset types at once
att.get_symbols(
    haghe_taqadom=True, sandogh=True, bonds=True,
    options=True, mortgage=True, commodity=True, energy=True,
)

# Filter by asset_type column
df = att.get_symbols(sandogh=True, bonds=True)
only_bonds = df[df['asset_type'] == 'bond']
only_funds = df[df['asset_type'] == 'fund']
```

Output as list

```
symbols = att.get_symbols(output='list')
# Returns: ['سازپیک', 'دعبید', 'آباد', ...].
```

get_currency()

Get historical price data for currencies and coins from TGJU.

توضیحات فارسی – 

تابع `(get_currency)` داده‌های تاریخی ارز و سکه را از سایت TGJU دریافت می‌کند.

پارامترها:

توضیح	مقدار پیش‌فرض	پارامتر
نام ارز/سکه به فارسی یا انگلیسی (رشته یا لیست)	–	<code>name</code>
تاریخ شروع شمسی به فرمت YYYY-MM-DD	<code>None</code>	<code>start</code>
تاریخ پایان شمسی به فرمت YYYY-MM-DD	<code>None</code>	<code>end</code>
تعداد آخرین روزهای معاملاتی (۰ = همه)	<code>0</code>	<code>limit</code>
نوع خروجی: 'full' یا 'standard'	<code>'standard'</code>	<code>output_type</code>
فرمت تاریخ: 'both' یا 'jalali', 'gregorian'	<code>'jalali'</code>	<code>date_format</code>
نمایش نوار پیشرفت	<code>True</code>	<code>progress</code>
ذخیره خروجی در فایل CSV	<code>False</code>	<code>save_to_file</code>
حذف ستون‌های اضافی در حالت چند ارزه	<code>True</code>	<code>dropna</code>
محاسبه بازدهی: 'simple', 'log', 'both'	<code>None</code>	<code>return_type</code>
مرتب‌سازی صعودی (True) یا نزولی (False) بر اساس تاریخ	<code>True</code>	<code>ascending</code>
مسیر فایل CSV برای ذخیره (مثال: 'output.csv')	<code>None</code>	<code>save_path</code>

ارزها و سکه‌های پشتیبانی شده (۱۴ مورد):

توضیح	نام فارسی	نام انگلیسی
دلار آمریکا	دلار	dollar
یورو اروپا	یورو	euro
درهم امارات	درهم	derham
لیر ترکیه	لیر	lira
پوند انگلیس	پوند	pond
یوان چین	یوان	yuan
سکه طلا تمام بهار آزادی (امامی)	سکه امامی	sekeh
نیم سکه بهار آزادی	نیم سکه	nim_sekeh
ربع سکه بهار آزادی	ربع سکه	rob_sekeh
سکه گرمی	سکه گرمی	sekeh_gerami
انس جهانی طلا	انس	ons
مثقال طلا	مثقال	mesghal
طلا ۱۸ عیار	طلا ۱۸ عیار	gold_18
طلا ۲۴ عیار	طلا ۲۴ عیار	gold_24

ستون‌های خروجی (حالت استاندارد):

توضیح	ستون
قیمت باز شدن	Open
بالاترین قیمت	High
پایین‌ترین قیمت	Low
قیمت بسته شدن	Close

نکات مهم: - می‌توان از نام فارسی یا انگلیسی استفاده کرد (مثلاً 'dollar' = 'دلار') - برای دریافت چند ارز همزمان، لیست ارسال کنید: `['$date_format='gregorian']` → ستون‌ها `MultiIndex` خواهند بود - با `datetime64` ایندکس به `return_type='log'` تغییر می‌کند - بازدهی لگاریتمی اضافه می‌شود

```

att.get_currency(
    name='dollar',                                # str or list - name in Persian or English
    start=None,                                    # str - start date in Jalali
    end=None,                                     # str - end date in Jalali
    limit=0,                                       # int - number of last trading days
    output_type='standard',                         # str - 'standard' or 'full'
    date_format='jalali',                           # str - 'jalali', 'gregorian', or 'both'
    progress=True,                                 # bool - show progress bar
    save_to_file=False,                            # bool - save to CSV
    dropna=True,                                  # bool - drop extra cols in multi-currency
    return_type=None,                             # str/list - 'simple', 'log', 'both', or ['simple','Close',5]
    ascending=True,                               # bool - sort ascending (True) or descending (False)
    save_path=None,                             # str - file path to save CSV
)

```

Supported names

English	فارسی	Description
dollar	دلار	US Dollar
euro	یورو	Euro
derham	درهم	UAE Dirham
lira	لیر	Turkish Lira
pond	پوند	British Pound
yuan	یوان	Chinese Yuan
sekeh	سکه امامی	Gold Coin (Emami)
nim_sekeh	نیم سکه	Half Coin
rob_sekeh	ربع سکه	Quarter Coin
sekeh_gerami	سکه گرمی	Gram Coin
ons	انس	Gold Ounce
mesghal	منقال	Mesghal
gold_18	طلای ۱۸ عیار	18K Gold
gold_24	طلای ۲۴ عیار	24K Gold

Standard output

```
df = att.get_currency('dollar', limit=10)
```

	Open	High	Low	Close
J-Date				
1404-11-16	1609350.0	1624700.0	1572300.0	1622400.0
1404-11-18	1619350.0	1619700.0	1564300.0	1564700.0
1404-11-19	1554450.0	1591700.0	1554300.0	1589500.0
1404-11-20	1592600.0	1617700.0	1592300.0	1612300.0
1404-11-21	1613600.0	1637700.0	1613300.0	1632400.0
1404-11-23	1624550.0	1627700.0	1617300.0	1625500.0
1404-11-25	1621350.0	1621700.0	1583800.0	1583900.0
1404-11-26	1586600.0	1603700.0	1586300.0	1597300.0
1404-11-27	1598550.0	1603700.0	1591300.0	1599600.0
1404-11-28	1599900.0	1629700.0	1599800.0	1608600.0

- Columns: `Open`, `High`, `Low`, `Close` (all `float64`)
- Index: `J-Date` (str – Jalali date)

Persian names work too

```
df = att.get_currency('دلار', limit=10)      # Same result as 'dollar'
df = att.get_currency('سکه', limit=10)        # Emami gold coin
df = att.get_currency('ربع سکه', limit=10)    # Quarter coin
```

With Gregorian dates

```
df = att.get_currency('dollar', limit=10, date_format='gregorian')
# Index: 'Date' (datetime64)
```

With return calculation

```
df = att.get_currency('dollar', limit=15, return_type='log')
```

	Open	High	Low	Close	returns
J-Date					
1404-11-09	1609450.0	1649700.0	1564300.0	1584400.0	NaN
1404-11-11	1599650.0	1649700.0	1582300.0	1629600.0	0.028129
1404-11-12	1624300.0	1624700.0	1574300.0	1589550.0	-0.024884
1404-11-13	1586600.0	1586700.0	1532300.0	1544400.0	-0.028815
1404-11-14	1544650.0	1571700.0	1534300.0	1568500.0	0.015484

Multiple currencies

```
df = att.get_currency(['سکه', 'ربيع سکه', 'euro'], limit=10)
```

	Open rob-seke	High rob-seke	Low rob-seke	Close rob-seke	Open euro	High euro	Low euro	Close euro
J-Date								
1404-11-16	5500000000.0	5655000000.0	5500000000.0	5600000000.0	1856100.0	1917900.0	1856100.0	1915600.0
1404-11-18	5400000000.0	5400000000.0	5200000000.0	5200000000.0	1914100.0	1914400.0	1849000.0	1849000.0
1404-11-19	5249000000.0	5507000000.0	5249000000.0	5495000000.0	1837300.0	1881300.0	1837200.0	1878600.0

- Returns a `MultiIndex` column structure: `(Column, Currency)`.

Date range

```
df = att.get_currency('dollar', start='1404-06-01', end='1404-08-01')
# Returns 51 trading days of dollar price history
```

get_intraday()

Get intraday (tick-level or candle) trade data for a symbol – today or historical.

توضیحات فارسی – get_intraday()

تابع  داده‌های معاملات درون‌روزی (تیک یا کندل) یک نماد را برمی‌گرداند – امروز یا تاریخی.

پارامترها:

پارامتر	مقدار پیش‌فرض	توضیح
symbol	–	نماد فارسی سهم
interval	'1min'	بازه زمانی کندل (جدول زیر)
start	'None'	تاریخ شروع (شمسی یا میلادی) برای داده تاریخی
end	'None'	تاریخ پایان (شمسی یا میلادی) برای داده تاریخی
progress	True	نمایش گزارش پیشرفت

بازه‌های زمانی پشتیبانی شده:

مقدار	توضیح
'tick'	داده خام تیک/اسنپ‌شات (بدون تجمعی)
'1min'	کندل ۱ دقیقه‌ای
'5min'	کندل ۵ دقیقه‌ای
'15min'	کندل ۱۵ دقیقه‌ای
'30min'	کندل ۳۰ دقیقه‌ای
'1h'	کندل ۱ ساعته
'4h'	کندل ۴ ساعته
'12h'	کندل ۱۲ ساعته

ستون‌های خروجی (حالت کندل):

ستون	توضیح
Open	قیمت باز شدن (<code>int</code>)
High	بالاترین قیمت (<code>int</code>)
Low	پایین‌ترین قیمت (<code>int</code>)
Close	قیمت بسته شدن (<code>int</code>)
Volume	حجم معاملات (<code>int</code>)
TradeCount	تعداد معاملات (<code>int</code>)

ستون‌های خروجی (حالت تیک – امروز):

ستون	توضیح
TradeNo	شماره معامله
Price	قیمت معامله (<code>float</code>)
Volume	حجم معامله
J-Date	تاریخ شمسی

ستون‌های خروجی (حالت تیک – تاریخی):

ستون	توضیح
Price	قیمت (<code>float</code>)
Volume	حجم
TradeCount	تعداد معاملات

نکات مهم: - بدون `start / end` : داده‌های امروز برگردانده می‌شود - با `start / end` : داده‌های تاریخی از آرشیو TSE دریافت می‌شود - ایندکس است - برای تیک امروز: معاملات تکی - برای تیک تاریخی: اسنپ‌شات‌های هر روز - این تابع **فقط برای سهام همیشه `datetime64` از نوع `DateTime`** کار می‌کند، نه شاخص‌ها

```
att.get_intraday(
    symbol='شتران',
    interval='1min',
    start=None,
    end=None,
    progress=True,
)
```

Supported intervals:

Value	Description
'tick'	Raw tick/snapshot data (no aggregation)
'1min'	1-minute candles
'5min'	5-minute candles
'15min'	15-minute candles
'30min'	30-minute candles
'1h'	1-hour candles
'4h'	4-hour candles
'12h'	12-hour candles

Today's candles (no start/end)

```
df = att.get_intraday('شتران', interval='1min')
```

DateTime	Open	High	Low	Close	Volume	TradeCount
2026-02-18 09:00:00	4079	4089	4067	4070	3687191	53
2026-02-18 09:01:00	4071	4089	4070	4089	952191	18
2026-02-18 09:02:00	4089	4089	4080	4089	475747	17
2026-02-18 09:03:00	4089	4117	4089	4117	1025581	30
2026-02-18 09:04:00	4116	4118	4090	4100	1307914	32

- Columns: `Open`, `High`, `Low`, `Close` (int), `Volume` (int), `TradeCount` (int)
- Index: `DateTime` (`datetime64`)

```
# 5-minute candles
df = att.get_intraday('شتران', interval='5min')
```

DateTime	Open	High	Low	Close	Volume	TradeCount
2026-02-18 09:00:00	4079	4118	4067	4100	7448624	150
2026-02-18 09:05:00	4100	4100	4037	4050	5321877	147
2026-02-18 09:10:00	4043	4049	4010	4032	9777574	220

```
# 1-hour candles
df = att.get_intraday('شتران', interval='1h')
```

DateTime	Open	High	Low	Close	Volume	TradeCount
2026-02-18 09:00:00	4079	4118	3966	3966	70752848	1993
2026-02-18 10:00:00	3965	3988	3916	3916	163093506	3051
2026-02-18 11:00:00	3916	3916	3916	3916	8004606	330
2026-02-18 12:00:00	3916	3917	3916	3916	85675975	1018

4-hour & 12-hour candles (new intervals)

```
# 4-hour candles
df = att.get_intraday('شتران', interval='4h')
```

	Open	High	Low	Close	Volume	TradeCount
DateTime						
2026-02-18 08:00:00	4079	4118	3916	3916	241850960	5374
2026-02-18 12:00:00	3916	3917	3916	3916	85675975	1018

```
# 12-hour candles
df = att.get_intraday('شتران', interval='12h')
```

	Open	High	Low	Close	Volume	TradeCount
DateTime						
2026-02-18 00:00:00	4079	4118	3916	3916	241850960	5374
2026-02-18 12:00:00	3916	3917	3916	3916	85675975	1018

Today's raw ticks

```
df = att.get_intraday('شتران', interval='tick')
```

	TradeNo	Price	Volume	J-Date
DateTime				
2026-02-18 09:00:17	1	4079.0	400000	1404-11-29
2026-02-18 09:00:17	2	4079.0	240000	1404-11-29
2026-02-18 09:00:17	3	4079.0	200000	1404-11-29
2026-02-18 09:00:17	4	4079.0	177119	1404-11-29
2026-02-18 09:00:17	5	4079.0	119805	1404-11-29

- **Columns:** TradeNo, Price, Volume, J-Date
- **Shape:** ~6000+ rows per day (individual trades)

Historical candles (with start/end)

```
# Single day
df = att.get_intraday('شتران', interval='5min', start='1404-11-06')
```

	Open	High	Low	Close	Volume	TradeCount
DateTime						
2026-01-26 09:00:00	4490	4490	4490	4490	4510224	17
2026-01-26 09:05:00	4490	4490	4490	4490	3264852	32
2026-01-26 09:10:00	4490	4490	4490	4490	655961	27

```
# Multi-day range
df = att.get_intraday('شتران', interval='1min',
                      start='1404-11-01', end='1404-11-06')
# Shape: (838, 6) - 838 one-minute candles across 5 trading days
```

Historical raw snapshots

```
df = att.get_intraday('شتران', interval='tick', start='1404-11-06')
```

	Price	Volume	TradeCount
DateTime			
2026-01-26 09:00:40	4490.0	1237786	88
2026-01-26 09:01:07	4490.0	75473	2
2026-01-26 09:01:16	4490.0	13313	1

- Columns:** `Price`, `Volume`, `TradeCount` (no TradeNo or J-Date for historical)
- Shape:** ~400 snapshots per day (~725 price points from ClosingPriceHistory)

Note:

This function works for individual stocks only – not for indices.

get_market_snapshot()

Get comprehensive real-time market data for

`all`

instruments in one API call – stocks, ETFs, options, bonds, and more.

توضیحات فارسی –

تابع `get_market_snapshot()` اطلاعات لحظه‌ای کل بازار را در یک فراخوانی برمی‌گرداند – سهام، صندوق‌های ETF، اختیار معامله، اوراق وغیره.

خروجی:

دیکشنری با سه کلید:

توضیح	نوع	کلید
دیتا فریم تمام نمادها با ۲۵ ستون	DataFrame	<code>stocks</code>
ساعت و تاریخ بازار (شمسی، مثلاً <code>"15:04:05 04/11/29"</code>)	str	<code>market_time</code>
مقدار شاخص کل بازار	float	<code>index_value</code>

ستون‌های دیتا فریم (۲۵ ستون):

توضیح	نوع	ستون
کد ابزار (شناسه یکتا – کلید اتصال با سایر داده‌ها)	<code>str</code>	<code>InsCode</code>
شناسه بین‌المللی اوراق بهادر	<code>str</code>	<code>ISIN</code>
نماد (مثالاً <code>نوری</code> ، <code>شتراں</code>)	<code>str</code>	<code>Symbol</code>
نام کامل شرکت	<code>str</code>	<code>Name</code>
زمان آخرين معامله (<code>HH:MM:SS</code> مرجع)	<code>str</code>	<code>Time</code>
قیمت دیروز (مرجع)	<code>int64</code>	<code>Yesterday</code>
قیمت پایانی (میانگین وزنی)	<code>int64</code>	<code>Close</code>
آخرین قیمت معامله شده	<code>int64</code>	<code>Last</code>
تعداد معاملات	<code>int64</code>	<code>TradeCount</code>
حجم معاملات	<code>int64</code>	<code>Volume</code>
ارزش معاملات (ریال)	<code>int64</code>	<code>Value</code>
پایین‌ترین قیمت روز	<code>int64</code>	<code>Low</code>
بالاترین قیمت روز	<code>int64</code>	<code>High</code>
سود هر سهم	<code>int64</code>	<code>EPS</code>
قیمت دیروز (آخرین معامله)	<code>int64</code>	<code>PriceYesterday</code>
کد جریان بازار	<code>int64</code>	<code>Flow</code>
کد گروه صنعت	<code>str</code>	<code>SectorCode</code>
سقف مجاز قیمت	<code>int64</code>	<code>MaxAllowed</code>
کف مجاز قیمت	<code>int64</code>	<code>MinAllowed</code>
حجم مبدأ	<code>int64</code>	<code>BaseVolume</code>
کد نوع ابزار (جدول زیر)	<code>int64</code>	<code>InstrumentType</code>
ارزش خالص دارایی (برای ETF‌ها، بقیه صفر)	<code>int64</code>	<code>NAV</code>
شناسه بازار: ... ، <code>N1</code> ، <code>N2</code> ، <code>Z1</code> ، <code>P1</code> ، <code>B1</code>	<code>str</code>	<code>MarketCode</code>
تغییر = پایانی - دیروز	<code>int64</code>	<code>Change</code>
درصد تغییر	<code>float64</code>	<code>ChangePct</code>

کدهای نوع ابزار (`InstrumentType`) :

کد	نوع
200	شاخص
208	صکوک
300	سهام بورس
301	حق تقدم بورس
303	سهام بورس (تابلوی فرعی)
305	صندوق
306	اوراق بدھی (مراجھہ/اجارہ/خزانہ)
309	فراپورس
311 / 312	اختیارمعاملہ (خرید/فروش)
400 / 403	حق تقدم فراپورس
706	اوراق دولتی

مثال فیلتر: - ETF - `InstrumentType == 305` → فقط صندوقهای - سهام عادی - `InstrumentType.isin([300, 303, 309])` - → فقط سهام عادی - `InstrumentType.isin([300, 303, 309])` - → فقط اختیارمعاملہ `InstrumentType.isin([311, 312])`

```
data = att.get_market_snapshot()

stocks_df      = data['stocks']      # DataFrame - all instruments with full data
market_time    = data['market_time']  # str - Jalali date/time (e.g. "04/11/29 15:04:05")
index_value    = data['index_value'] # float - شاخص کل کل (total market index)
```

Full output

```
data = att.get_market_snapshot()
print(data['stocks'].shape)          # DataFrame of all instruments
print(data['market_time'])          # '04/11/29 15:04:05'
print(data['index_value'])          # 3806743.94
```

	InsCode	ISIN	Symbol	Name	Time	Yesterday	Close				
Last	TradeCount	Volume	Value	Low	High	EPS	Price	Yesterday	Flow	SectorCode	MaxAllowed
MinAllowed	BaseVolume	InstrumentType	NAV	MarketCode	Change						
0	9538218081776543	IRO9AHRM0281	15407	15803		12:25:07	1404/11/29-14000-	1404/11/29-14000-	-	فهرم 1116	اختبارخ اهرم
1000	0	0		68	0	15803	14641	1016851000	66	14	15210
3	311A	-396	-2.51								
1	63185775846688586	IRO9AHRM0331	7454	8000		12:28:48	1404/11/29-22000-	1404/11/29-22000-	-	فهرم 1121	اختبارخ اهرم
1000	0	0		68	0	8200	6910	17800942000	2388	48	7220
3	311A	-546	-6.82								
2	358972276573533	IRO9AHRM0341	5354	6051		12:29:12	1404/11/29-24000-	1404/11/29-24000-	-	فهرم 1122	اختبارخ اهرم
1000	0	0		68	0	6200	4755	36508070000	6819	161	5230
3	311A	-697	-11.52								

Stocks DataFrame columns (25 columns)

Column	Type	Description
<code>InsCode</code>	str	Instrument code (unique ID, used for joining with other data)
<code>ISIN</code>	str	International Securities ID (e.g. <code>IRO1NORI0001</code>)
<code>Symbol</code>	str	نماد (e.g. نوری , شتران)
<code>Name</code>	str	Full company name (e.g. پتروشیمی نوری)
<code>Time</code>	str	Last trade time (HH:MM:SS)
<code>Yesterday</code>	int64	Yesterday's reference/closing price (دیروز)
<code>Close</code>	int64	Today's closing / weighted avg price (قیمت پایانی)
<code>Last</code>	int64	Last traded price (آخرین معامله)
<code>TradeCount</code>	int64	Number of trades
<code>Volume</code>	int64	Total volume
<code>Value</code>	int64	Total value (Rials)
<code>Low</code>	int64	Day's lowest price
<code>High</code>	int64	Day's highest price
<code>EPS</code>	int64	Earnings per share
<code>PriceYesterday</code>	int64	Yesterday's last traded price (قیمت دیروز - آخرین معامله)
<code>Flow</code>	int64	Market flow code (جريان بازار)
<code>SectorCode</code>	str	کد گروه صنعت (Industry/sector group code)
<code>MaxAllowed</code>	int64	Upper price limit (سقف مجاز)
<code>MinAllowed</code>	int64	Lower price limit (کف مجاز)
<code>BaseVolume</code>	int64	Base volume / total shares (حجم مبنا)
<code>InstrumentType</code>	int64	Type code: <code>300</code> =stock, <code>305</code> =ETF, <code>306</code> =bond, <code>309</code> =OTC, <code>311</code> / <code>312</code> =option, <code>706</code> =govt bond, ...
<code>NAV</code>	int64	Net Asset Value (for ETFs, 0 for stocks)
<code>MarketCode</code>	str	Market identifier: <code>N1</code> , <code>N2</code> , <code>Z1</code> , <code>P1</code> , <code>B1</code> , ...
<code>Change</code>	int64	Close - Yesterday
<code>ChangePct</code>	float64	Change as percentage

Filter by instrument type

```
data = att.get_market_snapshot()

# Regular stocks only (300=Bourse, 303=Bourse secondary, 309=FaraBourse)
stocks = data['stocks'][data['stocks']['InstrumentType'].isin([300, 303, 309])]
print(stocks.shape)    # (941, 25)

# ETFs only (305)
etfs = data['stocks'][data['stocks']['InstrumentType'] == 305]
print(etfs.shape)      # ETF instruments

# Options only (311/312)
options = data['stocks'][data['stocks']['InstrumentType'].isin([311, 312])]
print(options.shape)   # option contracts
```

Practical examples

```
# Top volume stocks
stocks = data['stocks'][data['stocks']['InstrumentType'].isin([300, 303, 309])]
top = stocks.nlargest(10, 'Volume')[['Symbol', 'Last', 'Volume', 'ChangePct']]

# Stocks up > 3%
gainers = stocks[stocks['ChangePct'] > 3][['Symbol', 'Last', 'ChangePct']]

# Filter by sector
sector = stocks[stocks['SectorCode'] == '27'][['Symbol', 'Name', 'Last', 'EPS']]
```

get_market_client_type()

Get individual (حقیقی) vs institutional (حقوقی) trade data for

all

instruments in one call.

 **توضیحات فارسی** — [\(\)get_market_client_type](#)

اطلاعات حقیقی/حقوقی تمام نمادهای بازار را در یک فراخوانی برمی‌گرداند.

تابع [\(\)get_market_client_type](#): **ستون‌های خروجی (۱۱ ستون):**

توضیح	ستون
کد ابزار (کلید اتصال با <code>(get_market_snapshot</code>)	<code>InsCode</code>
تعداد معاملات خرید حقیقی	<code>Buy_I_Count</code>
تعداد معاملات خرید حقوقی	<code>Buy_N_Count</code>
حجم خرید حقیقی	<code>Buy_I_Volume</code>
حجم خرید حقوقی	<code>Buy_N_Volume</code>
تعداد معاملات فروش حقیقی	<code>Sell_I_Count</code>
تعداد معاملات فروش حقوقی	<code>Sell_N_Count</code>
حجم فروش حقیقی	<code>Sell_I_Volume</code>
حجم فروش حقوقی	<code>Sell_N_Volume</code>
خالص حجم حقیقی (خرید - فروش)	<code>Net_I_Volume</code>
خالص حجم حقوقی (خرید - فروش)	<code>Net_N_Volume</code>

ابعاد خروجی:

یک سطر به ازای هر نماد $\times 11$ ستون

: `(get_market_snapshot` ترکیب با

برای ساخت فیلتر بازار کامل، این دو تابع را با `merge` ترکیب کنید:

```
merged = data['stocks'].merge(ct, on='InsCode', how='left')
```

اکنون هم داده قیمتی و هم اطلاعات حقیقی/حقوقی در یک دیتافریم دارید!

کاربردها: - شناسایی نمادهایی با خرید سنگین حقوقی: `Net_N_Volume > 1_000_000` - محاسبه نسبت حقیقی به حقوقی در خرید و فروش

```
df = att.get_market_client_type()
```

	InsCode	Buy_I_Count	Buy_N_Count	Buy_I_Volume	Buy_N_Volume	Sell_I_Count	Sell_N_Count
	Sell_I_Volume	Sell_N_Volume	Net_I_Volume	Net_N_Volume			
0	39453972158399542		502	10	66409668	119990996	176
	27770768	158629896	38638900	-38638900			8
1	65249046611427924		168	13	12411142	58529003	127
	7625574	63314571	4785568	-4785568			11
2	34718633636164421		441	4	127933422	25586667	255
	54682505	98837584	73250917	-73250917			4

- **Shape:** ($\sim 1880, 11$)

Column	Description
<code>InsCode</code>	Instrument code (join key with <code>get_market_snapshot</code>)
<code>Buy_I_Count</code>	Individual buy trade count
<code>Buy_N_Count</code>	Institutional buy trade count
<code>Buy_I_Volume</code>	Individual buy volume
<code>Buy_N_Volume</code>	Institutional buy volume
<code>Sell_I_Count</code>	Individual sell trade count
<code>Sell_N_Count</code>	Institutional sell trade count
<code>Sell_I_Volume</code>	Individual sell volume
<code>Sell_N_Volume</code>	Institutional sell volume
<code>Net_I_Volume</code>	Net individual volume (buy - sell)
<code>Net_N_Volume</code>	Net institutional volume (buy - sell)

Join with `get_market_snapshot`

```
data = att.get_market_snapshot()
ct = att.get_market_client_type()

merged = data['stocks'].merge(ct, on='InsCode', how='left')
# Now you have price data + client type data in one DataFrame!

# Find stocks with strong institutional buying
inst_buy = merged[merged['Net_N_Volume'] > 1_000_000]
print(inst_buy[['Symbol', 'Last', 'Volume', 'Net_N_Volume', 'Net_I_Volume']])
```

`list_options()`

List all active option contracts from the live market. Automatically parses option names to extract structured metadata (type, underlying, strike, expiry).

 `list_options` – توضیحات فارسی

تابع `()list_options` لیست تمام اختیار معامله‌های فعال بازار را برمی‌گرداند و نام اختیار معامله را به صورت خودکار تجزیه می‌کند.

خروجی:

دیتا فریم شامل تمام قراردادهای اختیار خرید و فروش فعال بازار

پارامترها:

توضیح	پیشفرض	پارامتر
فیلتر بر اساس نام دارایی پایه (مثلاً 'اهرم')	None	underlying
نمایش گزارش پیشرفت	True	progress

ستون‌های خروجی:

توضیح	ستون
نوع: 'call' (اختیار خرید) یا 'put' (اختیار فروش)	OptionType
نام دارایی پایه	Underlying
قیمت اعمال	Strike
تاریخ سرسید شمسی	ExpiryJalali
روزهای باقیمانده تا سرسید	DaysToExpiry

```
att.list_options(
    underlying=None,      # str or None - filter by underlying name (e.g. 'اهرم')
    progress=True,        # bool - show progress messages
)
```

```
# All active options
options = att.list_options()
print(options.shape)
```

Calls: 757, Puts: 716
Unique underlyings (20):
['اکسپریس', 'اکسپریس', 'اکسپریس']

Sample:

Symbol	OptionType	Underlying	Strike	ExpiryJalali	DaysToExpiry	Close	Volume
1114فهرم	call	229	17024	0	1404/11/29	12000	اهرم
1115فهرم	call	536	16103	0	1404/11/29	13000	اهرم
1116فهرم	call	66	15407	0	1404/11/29	14000	اهرم
1117فهرم	call	359	13988	0	1404/11/29	15000	اهرم
1118فهرم	call	73	13421	0	1404/11/29	16000	اهرم

Column	Type	Description
Column	Type	Description
---	---	---
OptionType	str	'call' or 'put'
Underlying	str	Underlying asset name (e.g. 'اهرم')
Strike	int	Strike price
ExpiryJalali	str	Expiry date in Jalali (e.g. '1404/11/29')
ExpiryGregorian	date	Expiry date in Gregorian
DaysToExpiry	int	Trading days until expiry
Yesterday	int	Yesterday's reference price
Change, ChangePct	int/float	Price change and percentage
MaxAllowed, MinAllowed	int	Price limits
Close, Last, Volume, Value, TradeCount	int	Market data from market_watch

```
# Filter by underlying
ahrm = att.list_options(underlying='اهرم')
print(ahrm[['Symbol', 'OptionType', 'Strike', 'ExpiryJalali', 'Close', 'Volume']])
```

get_options_chain()

Get a structured options chain for a specific underlying asset, optionally with Open Interest data from the TSETMC API.

توضیحات فارسی – `get_options_chain`

تابع `get_options_chain` زنجیره اختیار معامله یک دارایی پایه خاص را بر می‌گرداند.

پارامترها:

توضیح	پیش‌فرض	پارامتر
نام دارایی پایه (مثلاً 'اهرم')	–	underlying
دریافت Open Interest API به ازای هر قرارداد	False	fetch_oi
نمایش گزارش پیشرفت	True	progress

خروجی: دیکشنری با کلیدهای: - `calls` – دیتا فریم اختیارهای خرید - `puts` – دیتا فریم اختیارهای فروش - `underlying_price` - قیمت فعلی دارایی پایه - `expiry_dates` – لیست تاریخهای سرسید

: `fetch_oi=True` با ستونهای اضافی

توضیح	ستون
OpenInterest	تعداد موقعیت باز
ContractSize	اندازه قرارداد

```
att.get_options_chain(
    underlying='اهرم',      # str - underlying asset name
    fetch_oi=False,          # bool - fetch Open Interest (slower, per-contract API call)
    progress=True,           # bool - show progress messages
)
```

Returns a

dict

with the following keys:

Key	Type	Description
calls	DataFrame	All call option contracts
puts	DataFrame	All put option contracts
underlying_name	str	Underlying asset name
underlying_price	int	Current price of the underlying
expiry_dates	list	List of unique expiry dates (Jalali)
market_time	str	Market snapshot timestamp

```
chain = att.get_options_chain('اهرم')
print(chain['underlying_price']) # 29300
print(chain['expiry_dates'])    # ['1404/11/29', '1404/12/26', '1405/01/26', '1405/02/30', '1405/03/27']
print(f"Calls: {len(chain['calls'])}, Puts: {len(chain['puts'])}") # Calls: 69, Puts: 69
print(chain['calls'][['Symbol', 'Strike', 'Close', 'Volume', 'DaysToExpiry']].head())
```

Symbol	Strike	Close	Volume	DaysToExpiry
0	229	17024	12000	1114مهر
0	536	16103	13000	1115مهر
0	66	15407	14000	1116مهر
0	359	13988	15000	1117مهر
0	73	13421	16000	1118مهر

With Open Interest

(requires additional API calls per contract):

```
chain = att.get_options_chain('اهرم', fetch_oi=True)
print(chain['calls'][['Symbol', 'Strike', 'Close', 'OpenInterest', 'ContractSize']].head())
```

Symbol	Strike	Close	OpenInterest	ContractSize
1000	3892	17024	12000	1114مهرم
1000	3575	16103	13000	1115مهرم
1000	2272	15407	14000	1116مهرم
1000	2180	13988	15000	1117مهرم
1000	1618	13421	16000	1118مهرم

Extra Column (with `fetch_oi=True`)

Description

<code>OpenInterest</code>	(تعداد موقعیت باز) Open Interest
<code>ContractSize</code>	(اندازه قرارداد) Contract size
<code>BeginDate</code>	Contract begin date
<code>EndDate</code>	Contract end date

`list_etfs()`

List all active ETFs (Exchange-Traded Funds) with NAV and discount/premium calculation.

توضیحات فارسی – `()list_etfs`

تابع `()list_etfs` لیست تمام صندوق‌های ETF فعال بازار را با محاسبه تخفیف/حساب NAV برمی‌گرداند.

ستون‌های اضافی:

ستون

توضیح

<code>NAV</code>	ارزش خالص دارایی هر واحد
<code>NAV_Discount</code>	درصد تخفیف/حساب: $(پایانی - NAV \times 100) / (NAV)$ – منفی = تخفیف، مثبت = حباب

کاربردها: - شناسایی صندوق‌های با تخفیف بالا (فرصت خرید) - شناسایی صندوق‌های با حباب (ریسک بالا) - مقایسه NAV و قیمت بازار

```
att.list_etfs(
    progress=True,    # bool – show progress messages
)
```

```
etfs = att.list_etfs()
print(etfs.shape)
print(etfs[['Symbol', 'Close', 'NAV', 'NAV_Discount', 'Volume']].head(10))
```

Symbol	Close	NAV	NAV_Discount	Volume
733611397	0.01-	14292	14290	پاسارگاد
718737652	0.17-	25190	25146	آوند
511193722	0.15	30495	30540	کارا
339753042	0.4	16058	16122	ارکیده
329479400	0.17	38861	38927	باقوت
316921179	0.07-	20095	20080	توسکا
277853594	0.08	44355	44391	افران
276491840	0.08	25425	25446	لیخند
205012592	0.11	10248	10259	همای
197467929	14.2-	34792	29850	اهرم

Column	Type	Description
NAV	int64	(ارزش خالص دارایی) Net Asset Value per share
NAV_Discount	float64	$(\text{Close} - \text{NAV}) / \text{NAV} \times 100$ – negative = discount, positive = premium
All <code>market_watch</code> columns	–	Full market data (Close, Last, Volume, etc.)

Note: NAV is parsed from `get_market_snapshot()` data and is broadcast by TSETMC for ETFs. For non-ETF instruments, NAV is 0. Not all ETFs have a non-zero NAV value.

`list_bonds()`

List all active bonds (اوراق مرابحه) and treasury bills (اسناد خزانه) with maturity date parsing.

توضیحات فارسی – `list_bonds()`

تابع `list_bonds()` لیست تمام اوراق مرابحه و اسناد خزانه فعال بازار را با تجزیه تاریخ سرسید برمی‌گرداند.

ستون‌های اضافی:

ستون	توضیح
<code>BondType</code>	نوع اوراق: 'treasury' (خزانه) یا 'ijara' (اجاره) یا 'murabaha' (مرباحه)
<code>Ticker</code>	نماد کوتاه (مثلًا ۱۷۵۴۱، ۴۰۲۴۱) از آناد
<code>MaturityJalali</code>	تاریخ سرسید شمسی
<code>DaysToMaturity</code>	روزهای باقیمانده تا سرسید

نحوه تجزیه سرسید: - مرابحه: از ش. خ. ۰۶۰۳۲۷ تاریخ ۱۴۰۶/۰۳/۲۷ استخراج می‌شود - خزانه: ۶ رقم آخر قبل از پرانتز (مثلًا ۰۷۰۶۱۴) - اجاره: بر اساس کلمه `اجاره` در نام ابزار شناسایی می‌شود (۱۴۰۷/۰۶/۱۴)

انواع اوراق:

مرباحه، اجاره، خزانه
کاربردها: - مشاهده سرسید نزدیک اوراق - مقایسه بازدهی اوراق بر اساس روزهای باقیمانده - شناسایی اوراق نزدیک به سرسید (فرصت آربیتریاز) - فیلتر بر اساس نوع اوراق (مرباحه، اجاره، خزانه)

```
att.list_bonds(
    progress=True,    # bool - show progress messages
)
```

```
bonds = att.list_bonds()
print(bonds.shape)
print(bonds[['Symbol', 'Ticker', 'BondType', 'MaturityJalali', 'DaysToMaturity', 'Close']])
```

Symbol	Ticker	BondType	MaturityJalali	DaysToMaturity	Close
235اد	235اد	murabaha	1406/09/15	656	837740
072ارون	072ارون	murabaha	1407/05/20	904	1000000
210اخز	210اخز	treasury	1405/11/12	348	726120
402اخز	402اخز	treasury	1407/06/14	929	425000
07ايرتور	07ايرتور	ijara	1407/10/15	1051	1000000
15تابان	15تابان	ijara	1406/12/22	753	1000000

(خزانه) treasury (اجاره) ijara (مراجعه) murabaha Bond types:

Column	Type	Description
BondType	str	(خزانه) 'or 'treasury' (اجاره) 'ijara' (مراجعه) 'murabaha'
Ticker	str	Short ticker from parenthetical (e.g. 40241 اخز , 1754)
MaturityJalali	str	Maturity date in Jalali (e.g. 1406/03/27)
MaturityGregorian	date	Maturity date in Gregorian
DaysToMaturity	int	Days until maturity
All market_watch columns	—	Full market data (Close, Last, Volume, etc.)

Bonds expiring soon

```
# Find bonds/treasury bills expiring within 90 days
soon = bonds[bonds['DaysToMaturity'] < 90]
print(soon[['Symbol', 'Ticker', 'BondType', 'MaturityJalali', 'DaysToMaturity', 'Close']])
```

Symbol	Ticker	BondType	MaturityJalali	DaysToMaturity	Close
208اخز	208اخز	treasury	1404/12/11	12	997710
20841اخز	20841اخز	treasury	1404/12/11	12	989017
184اد	184اد	murabaha	1404/12/24	25	991510
05قدم	05قدم	murabaha	1405/02/01	62	1000000
904اد	904اد	murabaha	1405/02/17	78	971110

Maturity parsing: Bond names contain خ.ش {YYMMDD} where YYMMDD maps to 14YY/MM/DD . Treasury names have the maturity date as the last 6 digits before the parenthetical ticker. Ijara bonds (اجاره) are identified by the keyword اجاره in the instrument name.

list_funds()

List all investment funds with detailed NAV, returns, portfolio composition and manager info.

توضیحات فارسی – 

تابع `(list_funds)` اطلاعات کامل تمام صندوق‌های سرمایه‌گذاری را از API رسمی TSETMC دریافت می‌کند – شامل NAV، بازدهی، ترکیب پرتفوی، مدیر صندوق و متولی.

دسته‌بندی صندوق‌ها:

نوع	پارامتر	توضیح
صندوق سهامی	'equity'	سرمایه‌گذاری در سهام
صندوق درآمد ثابت	'fixed_income'	سپرده و اوراق با سود تصمیمی
صندوق مختلط	'mixed'	ترکیب سهام و اوراق
صندوق بازارگردانی	'market_maker'	بازارگردانی نمادها
صندوق جسورانه	'venture'	سرمایه‌گذاری خطروپزیر
صندوق پروژه	'project'	مبتنی بر پروژه
صندوق زمین و ساختمان	'real_estate'	املاک و مستغلات
صندوق کالایی	'commodity'	طلاء، نقره، زعفران
صندوق خصوصی	'private'	سرمایه‌گذاری خصوصی
ابر صندوق	'fund_of_funds'	صندوق در صندوق

ستون‌های NAV و بازدهی:

ستون	توضیح
<code>nav_redemption</code>	NAV ابطال (هر واحد)
<code>nav_subscription</code>	NAV صدور (هر واحد)
<code>return_1d / return_7d / return_30d</code>	بازدهی کوتاه‌مدت (درصد)
<code>return_90d / return_180d / return_365d</code>	بازدهی بلند‌مدت (درصد)
<code>return_inception</code>	بازدهی از ابتدا (درصد)

ستون‌های ترکیب پرتفوی:

توضیح	ستون
pct_stock	درصد سهام
pct_bond	درصد اوراق
pct_deposit	درصد سپرده بانکی
pct_cash	درصد نقد
pct_other	درصد سایر
pct_top5	غلظت ۵ دارایی برتر

مثال‌ها:

→ همه صندوق‌ها → `att.list_funds()`
 → فقط صندوق‌های سهامی → `att.list_funds(fund_type='equity')`
 → صندوق‌های کالایی (طلاء، نقره) → `att.list_funds(fund_type='commodity')`
 → سهامی + مختلط → `att.list_funds(fund_type=['equity', 'mixed'])`

```
att.list_funds(
    fund_type=None,      # str or list - fund category filter (default: all)
    progress=True,        # bool - show progress messages
)
```

All funds

```
df = att.list_funds()
```

			fund_name	fund_type	reg_no	nav_redemption	nav_subscription	return_365d
				manager				
0	pct_stock	pct_bond	آرمان آتیه درخشنان مس	equity	11378	569066	573967	50.47
0.00			تامین سرمایه تمدن					75.10
1	آرمان رایا یکم		آرمان رایا سهم	equity	12061	2020046	2040493	33.33
1.75		2.69	سبدگردان رایا سهم					96.36
			...					

Column	Type	Description
<code>fund_name</code>	str	Fund name in Persian
<code>fund_type</code>	str	Category: <code>equity</code> , <code>fixed_income</code> , <code>mixed</code> , <code>market_maker</code> , <code>venture</code> , <code>project</code> , <code>commodity</code> , <code>private</code> , <code>fund_of_funds</code>
<code>reg_no</code>	int	Registration number
<code>nav_redemption</code>	float	NAV for redemption (per unit)
<code>nav_subscription</code>	float	NAV for subscription (per unit)
<code>nav_statistical</code>	float	Statistical NAV
<code>net_asset</code>	float	Total net asset value (Rials)
<code>units</code>	float	Outstanding units
<code>inception_date</code>	str	Fund start date (ISO)
<code>return_1d</code> ... <code>return_365d</code>	float	Returns at various horizons (%)
<code>return_inception</code>	float	Return since inception (%)
<code>pct_stock</code>	float	Equity allocation (%)
<code>pct_bond</code>	float	Bond allocation (%)
<code>pct_deposit</code>	float	Bank deposit allocation (%)
<code>pct_cash</code>	float	Cash allocation (%)
<code>pct_other</code>	float	Other assets (%)
<code>pct_top5</code>	float	Top-5 holding concentration (%)
<code>manager</code>	str	Fund manager name
<code>investment_manager</code>	str	Investment manager
<code>custodian</code>	str	Custodian / auditor
<code>guarantor</code>	str	Guarantor (or None)
<code>market_maker</code>	str	Market maker (or None)

Filter by fund type

```
# Equity funds only
equity = att.list_funds(fund_type='equity')

# Fixed income funds
fixed = att.list_funds(fund_type='fixed_income')

# Commodity (gold, silver, saffron)
gold = att.list_funds(fund_type='commodity')

# Multiple types
mix = att.list_funds(fund_type=['equity', 'mixed', 'venture'])
```

Top performers

```
# Best annual return across all funds
df = att.list_funds()
top10 = df.nlargest(10, 'return_365d')[['fund_name', 'fund_type', 'return_365d', 'pct_stock']]
print(top10)
```

Portfolio analysis

```
# Equity funds with highest stock concentration
eq = att.list_funds(fund_type='equity')
high_stock = eq[eq['pct_stock'] > 80].sort_values('return_365d', ascending=False)
print(high_stock[['fund_name', 'pct_stock', 'pct_top5', 'return_365d']])
```

Compare NAVs

```
# All fixed income funds sorted by redemption NAV
fixed = att.list_funds(fund_type='fixed_income')
print(fixed.sort_values('nav_redemption', ascending=False)[['fund_name', 'nav_redemption', 'return_365d', 'net_asset']])
```

Legacy Aliases

 توضیحات فارسی – نامهای قدیمی

برای

 سازگاری با گذشته

(backward compatibility)، نامهای قدیمی توابع همچنان کار می‌کنند. هم نامهای قدیمی و هم نامهای جدید به طور یکسان عمل می‌کنند و نامهای قدیمی هرگز حذف نخواهند شد.

For

backward compatibility

, the original function names are still available as aliases. Both old and new names work identically – the old names will never be removed.

Standard Name	Legacy Alias	Description
<code>get_history()</code>	<code>stock()</code>	Historical price data
<code>get_client_type()</code>	<code>stock_RI()</code> , <code>stock_RL()</code>	Retail / institutional data
<code>get_capital_increase()</code>	<code>stock_capital_increase()</code>	Capital increase history
<code>get_intraday()</code>	<code>stock_intraday()</code>	Intraday tick & candle data
<code>get_detail()</code>	<code>stockdetail()</code>	Full stock detail
<code>get_info()</code>	<code>stock_information()</code>	Instrument information
<code>get_stats()</code>	<code>stock_statistics()</code>	Instrument statistics
<code>get_symbols()</code>	<code>stocklist()</code>	List all symbols
<code>get_shareholders()</code>	<code>shareholders()</code>	Major shareholders
<code>get_currency()</code>	<code>currency_coin()</code>	Currency & coin prices
<code>get_market_snapshot()</code>	<code>market_watch()</code>	Live market snapshot
<code>get_market_client_type()</code>	<code>market_client_type()</code>	Bulk individual/institutional

```
import algotik_tse as att

# These are identical:
df = att.get_history('شتران', limit=100)
df = att.stock('شتران', limit=100)           # legacy - still works

# Legacy names for all functions:
att.stock_intraday('شتران', interval='4h')    # same as att.get_intraday( ... )
att.market_watch()                                # same as att.get_market_snapshot()
att.currency_coin('dollar', limit=30)             # same as att.get_currency( ... )
```

Legacy Parameter Names

نامهای قدیمی پارامترها نیز همچنان پشتیبانی می‌شوند:

Old parameter names are also still supported for backward compatibility:

New Name	Old Name	Functions
symbol	stock	get_history(), get_client_type(), get_capital_increase(), get_shareholders()
name	currency_coin_name	get_currency()
limit	values	get_history(), get_client_type(), get_currency()
raw	tse_format	get_history(), get_client_type()
dropna	multi_stock_drop	get_history(), get_client_type()
dropna	multi_currencies_drop	get_currency()
include_id	shh_id	get_shareholders()
output_type='full'	output_type='complete'	get_history(), get_client_type(), get_currency()

```
# Old parameter names still work:
df = att.get_history(stock='شتران', values=100, tse_format=True)
# is the same as:
df = att.get_history(symbol='شتران', limit=100, raw=True)
```

Configuration

All settings are accessible via the global `settings` object:

```
import algotik_tse as att

# SSL verification (default: False)
att.settings.ssl_verify = True

# Request timeout in seconds (default: 10)
att.settings.timeout = 15

# Maximum retry attempts on failure (default: 3)
att.settings.max_retries = 5

# Delay between requests in seconds - prevents TSETMC rate-limiting (default: 0.3)
att.settings.rate_limit_delay = 0.5
```

Setting	Default	Description
ssl_verify	False	Enable/disable SSL certificate verification
timeout	10	Request timeout in seconds
max_retries	3	Maximum retry attempts on HTTP failure
rate_limit_delay	0.3	Delay between consecutive requests (seconds)

Note: TSETMC may temporarily block your IP if you send too many requests. The `rate_limit_delay` setting adds a pause between requests to avoid this. If you are downloading data for many symbols, keep this value at `0.3` or higher.

توضیحات فارسی – تنظیمات

تنظیمات از طریق شیء سراسری `settings` قابل دسترسی هستند:

توضیح	پیشفرض	تنظیم
فعال/غیرفعال کردن تأیید گواهی SSL	<code>False</code>	<code>ssl_verify</code>
زمان انتظار درخواست (ثانیه)	<code>10</code>	<code>timeout</code>
حداکثر تعداد تلاش مجدد در صورت خطا	<code>3</code>	<code>max_retries</code>
تأخير بین درخواست‌های متوالی (ثانیه)	<code>0.3</code>	<code>rate_limit_delay</code>

هشدار: سایت TSETMC ممکن است در صورت ارسال درخواست‌های زیاد، IP شما را مسدود کند. تنظیم `rate_limit_delay` یک مکث بین درخواست‌ها اضافه می‌کند. اگر برای نمادهای زیادی داده دانلود می‌کنید، این مقدار را `0.3` یا بیشتر نگه دارید.

Examples

Market Screening

شناسایی نمادهای پرحجم، بیشترین رشد و بیشترین افت با `get_market_snapshot`

```
import algotik_tse as att

data = att.get_market_snapshot()
stocks = data['stocks']

# Filter real stocks only
real = stocks[stocks['InstrumentType'].isin([300, 303, 309])].copy()

# Top 10 by volume
top_vol = real.nlargest(10, 'Volume')[['Symbol', 'Last', 'ChangePct', 'Volume']]
print(top_vol)
```

(بیشترین حجم) Output

Symbol	Last	ChangePct	Volume	
4753792093	3-	502	خودرو	
3700000000	0	6988	ودی	
3631847572	0-	515	خسپا	
2230874402	3-	1257	وبملت	
1025392677	4-	407	وتجارت	
5956144448	4-	342	ذوب	
435798815	0-	404	اخبار	
397605094	0-	6430	وسایا	
363823141	3-	504	وبصادر	
343240547	4-	7540	شبندر	

```
# Top 10 gainers / losers
top_gain = real.nlargest(10, 'ChangePct')[['Symbol', 'Last', 'ChangePct', 'Volume']]
top_loss = real.nsmallest(10, 'ChangePct')[['Symbol', 'Last', 'ChangePct', 'Volume']]
print(top_gain)
print(top_loss)
```

(بیشترین رشد و افت) Output

```
# Top 10 gainers:
Symbol Last ChangePct Volume
319963504 62 199797 آلومینا 4
9850557 6 1444 وپسا
2755408 6 920 فبیرا
11698640 5 1062 خفنر
5083029 5 2595 سفاسی
3673013 5 4645 وگستر
7461884 5 3307 ولشرق
368174 4 7150 آبادا
12709430 4 3965 آریان
308651 4 2060 آوا
```

```
# Top 10 losers:
Symbol Last ChangePct Volume
65799 6- 140400 شکبیر
1229688 6- 3629 پلاسک
3996087 5- 60340 شیراز
450580 5- 14810 دشیمی
2287467 5- 10900 پارتا
96995 5- 79120 غبهنوش
8029618 5- 4499 پاکشو
3257908 5- 2596 خنصیر
9626376 5- 2566 شسینا
105105 5- 4400 شدوص
```

ETF Discount/Premium Analysis

شناختی صندوق‌های ETF با بیشترین تخفیف یا حباب نسبت به NAV – فرصت‌های آربیتراژ

```

import algotik_tse as att

etfs = att.list_etfs()
active = etfs[etfs['NAV'] > 0].copy()
print(f"Total ETFs: {len(etfs)}, with NAV data: {len(active)}")

# Most discounted (buying opportunity)
discounted = active.nsmallest(5, 'NAV_Discount')
print(discounted[['Symbol', 'Close', 'NAV', 'NAV_Discount', 'Volume']])

# Most premium (overvalued)
premium = active.nlargest(5, 'NAV_Discount')
print(premium[['Symbol', 'Close', 'NAV', 'NAV_Discount', 'Volume']])

```

Output

Total ETFs: 328, with NAV data: 247

# Top 5 most discounted ETFs (vs NAV):				
Symbol	Close	NAV	NAV_Discount	Volume
6236158	33-	453363	303890	دارا یکم
5522069	27-	385211	280500	پالایش
85301838	18-	25176	20730	بیدار
43573177	16-	22627	18990	شتاب
35709297	16-	17864	15080	جهش

# Top 5 most premium ETFs (vs NAV):				
Symbol	Close	NAV	NAV_Discount	Volume
496071	23	19240	23642	گارانتی
200159	12	14293	16039	عرش
78547	11	39792	44280	آسام
5111640	10	11034	12193	مالک آتیه
62216	10	18508	20435	رخش

Currency & Gold Prices

دریافت قیمت لحظه‌ای دلار و یورو

```

import algotik_tse as att

usd = att.get_currency('dollar', limit=5)
print(usd)

eur = att.get_currency('euro', limit=5)
print(eur)

```

Output

```
# Dollar (last 5 days):
      Open      High       Low      Close
J-Date
1404-11-25 1,621,350 1,621,700 1,583,800 1,583,900
1404-11-26 1,586,600 1,603,700 1,586,300 1,597,300
1404-11-27 1,598,550 1,603,700 1,591,300 1,599,600
1404-11-28 1,599,900 1,629,700 1,599,800 1,608,600
1404-11-29 1,610,300 1,629,700 1,610,300 1,623,350

# Euro (last 5 days):
      Open      High       Low      Close
J-Date
1404-11-25 1,685,200 1,685,200 1,646,000 1,646,100
1404-11-26 1,643,700 1,669,300 1,643,700 1,669,100
1404-11-27 1,670,600 1,692,200 1,670,600 1,682,600
1404-11-28 1,682,700 1,714,000 1,682,700 1,689,700
1404-11-29 1,692,400 1,714,600 1,692,400 1,706,500
```

Options Overview

آمار کلی بازار اختیار معامله – تعداد قراردادها، دارایی‌های پایه و پرمعامله‌ترین آپشن‌ها

```
import algotik_tse as att

options = att.list_options()
print(f"Total active options: {len(options)}")
print(f"Unique underlyings: {options['Underlying'].nunique()}")

# Top underlyings by option count
top = options.groupby('Underlying').size().nlargest(10).reset_index(name='Count')
print(top)

# Most traded options today
top_vol = options.nlargest(10, 'Volume')
print(top_vol[['Symbol', 'Underlying', 'OptionType', 'Strike', 'ExpiryJalali', 'Volume', 'Close']])
```

Output

Total active options: 1473

Unique underlyings: 20

Top 10 underlyings by number of options:

Underlying Count

138	اهرم
133	شستا
128	خودرو
128	وبملت
126	ذوب
119	خسپا
98	فولاد
74	جهش
56	تامیکو
56	فملی

Top 10 most traded options:

Symbol	Underlying	OptionType	Strike	ExpiryJalali	Volume	Close
خسپا	1138	ضسپا	call	500	1404/11/29	12385699
خودرو	1250	فخدود	call	550	1404/12/06	10291618
اهرم	1125	پهرم	call	30000	1404/11/29	3331288
خودرو	1249	ضخود	call	500	1404/12/06	3270106
خسپا	1247	ضسپا	call	500	1404/12/26	3085993
وتجارت	1235	ضجارت	call	550	1404/12/19	2591303
خودرو	1249	طخدود	put	500	1404/12/06	2475166
اهرم	1125	طهرم	put	30000	1404/11/29	2348106
وبملت	1205	ضملت	call	1300	1404/12/19	1668000
شستا	1242	طستا	put	1610	1404/12/13	1457739

Fund Comparison

مقایسه صندوق‌های سهامی و درآمد ثابت – تعداد صندوق‌ها و ستون‌های خروجی

```
import algotik_tse as att

# Equity vs Fixed Income funds
equity_funds = att.list_funds(fund_type='equity')
fi_funds = att.list_funds(fund_type='fixed_income')

print(f"Equity funds: {len(equity_funds)}")
print(f"Fixed income funds: {len(fi_funds)}")
print(f"Columns: {list(equity_funds.columns)}")

# Get ALL fund types at once
all_funds = att.list_funds()
print(f"All funds: {len(all_funds)}")
print(all_funds.groupby('fund_type').size())
```

Output

Equity funds: 122
Fixed income funds: 165

Columns: ['fund_name', 'fund_type', 'reg_no', 'nav_redemption',
'nav_subscription', 'nav_statistical', 'net_asset', 'units',
'inception_date', 'return_1d', 'return_7d', 'return_30d',
'return_90d', 'return_180d', 'return_365d', 'return_inception',
'pct_stock', 'pct_bond', 'pct_deposit', 'pct_cash', 'pct_other',
'pct_top5', 'manager', 'investment_manager', 'custodian',
'guarantor', 'market_maker']

Bond Maturity Analysis

تحلیل سررسید اوراق بدھی – تفکیک بر اساس نوع و نزدیکترین سررسیدها

```
import algotik_tse as att

bonds = att.list_bonds()
print(f"Total bonds/sukuk: {len(bonds)}")
print(bonds['BondType'].value_counts())

# 10 bonds with nearest maturity
near = bonds[bonds['DaysToMaturity'] > 0].nsmallest(10, 'DaysToMaturity')
print(near[['Symbol', 'Ticker', 'BondType', 'MaturityJalali', 'DaysToMaturity', 'Close']])
```

Output

Total bonds/sukuk: 349

BondType	
murabaha	283
ijara	45
treasury	21

# 10 bonds with nearest maturity:	Symbol	Ticker	BondType	MaturityJalali	DaysToMaturity	Close
	اخر41	20841	اخز41	treasury	1404/12/11	11 989017
	اخر41	20841	اخز41	treasury	1404/12/11	11 997710
	اراد4	184	اراد4	murabaha	1404/12/24	24 991510
	مقدم4	05	مقدم4	murabaha	1405/02/01	61 1000000
	اراد4	904	اراد4	murabaha	1405/02/17	77 971110
	اخز41	2011	اخز41	treasury	1405/03/25	116 898210
	كرمان4	531	كرمان4	murabaha	1405/03/27	118 1000000
	كرمان4	532	كرمان4	murabaha	1405/03/27	118 1000000
	مارينا4	05	مارينا4	ijara	1405/04/05	127 1000000
	اراد4	1664	اراد4	murabaha	1405/04/19	141 962402

Institutional Money Flow

خالص خرید و فروش حقوقی – شناسایی نمادهایی که حقوقی‌ها در حال خرید یا فروش هستند

```

import algotik_tse as att

ct = att.get_market_client_type()
data = att.get_market_snapshot()

# Merge for symbol names
stocks = data['stocks'][data['stocks']['InstrumentType'].isin([300, 303, 309])][[['InsCode', 'Symbol']]]
merged = ct.merge(stocks, on='InsCode', how='inner')

# Top institutional buyers & sellers
top_buy = merged.nlargest(10, 'Net_N_Volume')[['Symbol', 'Net_N_Volume']]
top_sell = merged.nsmallest(10, 'Net_N_Volume')[['Symbol', 'Net_N_Volume']]
print(top_buy)
print(top_sell)

```

Output

Top 10 institutional net buyers:

Symbol	Net_N_Volume
2770366877	خودرو
2082818640	خسآپا
366854539	ویملت
181253875	شتران
152961055	شبندر
146242076	فولاد
102337120	وبصادر
63769330	ونوین
58828670	شپنا
33587581	وتجارت

Top 10 institutional net sellers:

Symbol	Net_N_Volume
175200000-	فاذر
133683202-	وسحان
123021255-	ذوب
74580959-	ورنا
60574000-	کسراء
58559183-	اخبار
35444977-	واسآپا
31000000-	تبکو
22456287-	شگستر
22145336-	های وب

All Asset Types Overview

نمایش تمام انواع ابزارهای بازار – سهام، حق تقدم، صندوق، اوراق، اختیار، تسهیلات مسکن، کالا و انرژی

```

import algotik_tse as att

all_syms = att.get_symbols(
    bourse=True, farabourse=True, payeh=True,
    haghe_taqadom=True, sandogh=True,
    bonds=True, options=True, mortgage=True,
    commodity=True, energy=True
)
print(f"Total instruments: {len(all_syms)}")
print(all_syms['asset_type'].value_counts())

```

Output

```
Total instruments: 5086
```

```
asset_type
option      2498
stock       901
bond        728
right       482
fund        321
mortgage    87
commodity   48
energy      21
```

Intraday Candle Analysis

کندل‌های ۵ دقیقه‌ای و ۱ ساعته – مشاهده حرکات قیمت در طول روز

```
import algotik_tse as att

# 5-minute candles
candles = att.get_intraday('شتران', interval='5min')
print(f"5-min candles: {len(candles)} rows")
print(candles.tail(5))

# 1-hour candles
candles_1h = att.get_intraday('شتران', interval='1h')
print(candles_1h)
```

Output

```
# 5-min candles (last 5):
          Open  High   Low Close    Volume TradeCount
DateTime
2026-02-19 12:05:00  3916  3916  3916  3916    609524       43
2026-02-19 12:10:00  3916  3916  3916  3916    742435       27
2026-02-19 12:15:00  3916  3916  3916  3916   30783482      293
2026-02-19 12:20:00  3916  3916  3916  3916   39369865      428
2026-02-19 12:25:00  3916  3917  3916  3916   13706189      205

# 1-hour candles:
          Open  High   Low Close    Volume TradeCount
DateTime
2026-02-19 09:00:00  4079  4118  3966  3966   70752848     1993
2026-02-19 10:00:00  3965  3988  3916  3916  163093506     3051
2026-02-19 11:00:00  3916  3916  3916  3916   8004606      330
2026-02-19 12:00:00  3916  3917  3916  3916   85675975     1018
```

Stock Detail & Shareholders

اطلاعات کامل شرکت و سهامداران عمدہ

```
import algotik_tse as att

info = att.get_info('شتران')
holders = att.get_shareholders('شتران')
print(info)
print(holders.head(5))
```

Output

```
# Stock info for شتران:
                                         value
key
lVal18AFC                      شتران
lVal30                          پالیش نفت تهران
lVal18                          Palayesh Tehran
cIsin                           IRO1PTEH0007
flowTitle                        بازار بورس
cgrValCotTitle      بازار اول (تابلوی اصلی) بورس
eps_estimatedEPS                1018
eps_sectorPE                     5
zTitad                          539,500,000,000
minYear                         1,875
maxYear                          4,938

# Major shareholders (top 5):
                                         share_holder_name  number_of_shares  percentage
0                               بانک صادرات ایران           32,344,984,668
1       شرکت سرمایه گذاری ایرانیان -سهامی خاص-
2       شرکت سرمایه گذاری .ا.تهران -سهامی عام-- م ک م ف ع
3       شرکت .س. سهام عدالت .ا.خراسان رضوی -س ع -- م ک م ف ع
4                               PRX3                  17,974,075,372    -33322 -76894 موس
```

Data Sources

Source	URL	Data
TSETMC	old.tsetmc.com / cdn.tsetmc.com	Stock prices, indices, shareholders, capital increases, intraday, market watch
TGJU	api.tgju.org	Currency & coin prices

Contributing

Contributions are welcome! Please see [CONTRIBUTING.rst](#) for guidelines.

License

This project is licensed under the **GNU General Public License v3 (GPLv3)**. See [LICENSE](#) for the full license text.

Credits

- **Author:** Mohsen Alipour (alipour@algotik.ir)
- **Website:** algotik.com
- **Telegram:** [@algotik](https://t.me/algotik)
- Inspired by [finpy-tse](#) and [tsemodule5](#)