

example

December 23, 2024

1 Usage example of issmoex package

```
[ ]: # Installation of package
      #!pip install ISSMOEX==1.1.1.5
```

```
[ ]: # Initialization
      from issmoex import ISSMOEX
      iss = ISSMOEX()
```

The structure of ISS MOEX suggests the segmentation according to engine and markets. Each engine has specific set of markets.

```
[ ]: print('ISS engines:')
      print(iss.engines)

      print('\nISS engines and corresponding markets:')
      print(iss.engines_markets)
```

ISS engines:

	id	name	title
0	1	stock	
1	2	state	()
2	3	currency	
3	4	futures	
4	5	commodity	
5	6	interventions	
6	7	offboard	-
7	9	agro	
8	1012	otc	
9	1282	quotes	

ISS engines and corresponding markets:

```
{'stock': ['index', 'shares', 'bonds', 'ndm', 'otc', 'ccp', 'deposit', 'repo',
'qnv', 'mamc', 'foreignshares', 'foreignndm', 'moexboard', 'gcc', 'credit',
'nonresndm', 'nonresrepo', 'nonresccp', 'standard', 'classica'], 'state':
['index', 'bonds', 'repo', 'ndm'], 'currency': ['selt', 'futures', 'index',
'otc'], 'futures': ['main', 'forts', 'options', 'fortsiqs', 'optionsiqs'],
'commodity': ['futures'], 'interventions': ['grain'], 'offboard': ['bonds'],
```

```
'agro': ['sugar', 'auctions', 'buyauctions'], 'otc': ['shares', 'bonds',
'sharesndm', 'ndm'], 'quotes': ['bonds']}]
```

For example, there is an interest in the assets at shares market. Hence, we choose `engine = stock` and `market = shares`.

```
[ ]: shares_market = iss.securities_market(engine = 'stock', market = 'shares')
shares_market.head()
```

```
[ ]: SECID BOARDID  SHORTNAME  PREVPRICE  LOTSIZE  FACEVALUE  STATUS  \
0  ABIO    TQBR    i          99.500        10         0.10      A
1  ABRD    TQBR          279.000        10         1.00      A
2  ACKO    TQBR          3.580        100         1.00      A
3  AFKS    TQBR          26.191        100         0.09      A
4  AFLT    TQBR          57.800        10         1.00      A
```

```
BOARDNAME  DECIMALS  SECNAME  ...  \
0  +:    -    .        2          "  "  ...
1  +:    -    .        1          -    ...
2  +:    -    .        2          ...
3  +:    -    .        3          "  "  ...
4  +:    -    .        2    -  .    (  )  ...
```

```
PREVDATE  ISSUESIZE  ISIN  LATNAME  REGNUMBER  \
0  2024-06-10  92645451  RU000A0JNAB6  ARTGEN ao  1-01-08902-A
1  2024-06-10  98000184  RU000A0JS5T7  Abrau-Durso ao  1-02-12500-A
2  2024-06-10  536000000  RU000A0JXS91  ASKO ao  1-01-52065-Z
3  2024-06-10  9650000000  RU000A0DQZE3  AFK Sistema  1-05-01669-A
4  2024-06-10  3975771215  RU0009062285  Aeroflot  1-01-00010-A
```

```
PREVLEGALCLOSEPRICE  CURRENCYID  SECTYPE  LISTLEVEL  SETTLEDATE
0          99.300          SUR        1          2  2024-06-13
1         279.000          SUR        1          3  2024-06-13
2          3.580          SUR        1          3  2024-06-13
3         25.915          SUR        1          1  2024-06-13
4         57.200          SUR        1          1  2024-06-13
```

[5 rows x 27 columns]

```
[ ]: # History prices of SBER
hist_price_sber = iss.history_prices(engine = 'stock',market = 'shares',isin = 'SBER',date = '2024-01-01',show_progress=True)
hist_price_sber.head()
```

Fetching data: 100%| | 1/1 [00:00<00:00, 4.78it/s]

```
[ ]: BOARDID  TRADEDATE  SHORTNAME  SECID  NUMTRADES  VALUE  OPEN  \
0  TQBR  2024-01-03  SBER  50912  5.631305e+09  271.90
1  TQBR  2024-01-04  SBER  33045  3.218188e+09  274.67
```

2	TQBR	2024-01-05	SBER	26234	2.634960e+09	274.30
3	TQBR	2024-01-08	SBER	70828	5.924626e+09	273.60
4	TQBR	2024-01-09	SBER	40508	5.601231e+09	276.97

	LOW	HIGH	LEGALCLOSEPRICE	...	MARKETPRICE2	MARKETPRICE3	\
0	271.00	274.70	273.64	...	273.48	273.48	
1	273.70	275.48	274.04	...	274.42	274.42	
2	272.80	274.69	273.46	...	273.46	273.46	
3	273.53	277.00	275.81	...	275.61	275.61	
4	274.71	278.00	276.00	...	275.71	275.71	

	ADMITTEDQUOTE	MP2VALTRD	MARKETPRICE3TRADESVALUE	ADMITTEDVALUE	WAVAL	\
0	NaN	4.994800e+09	4.994800e+09	NaN	0	
1	NaN	2.822724e+09	2.822724e+09	NaN	0	
2	NaN	2.246293e+09	2.246293e+09	NaN	0	
3	NaN	5.204970e+09	5.204970e+09	NaN	0	
4	NaN	4.962470e+09	4.962470e+09	NaN	0	

	TRADINGSESSION	CURRENCYID	TRENDCLSPR
0	3	SUR	1.38
1	3	SUR	-0.16
2	3	SUR	-0.18
3	3	SUR	1.15
4	3	SUR	-0.53

[5 rows x 23 columns]

Suppose there is a necessity to load the history prices for several shares: SBER, GAZP and AFKS.

```
[ ]: isins = ['SBER','GAZP','AFKS']
hist_prices = iss.history_prices(engine = 'stock',market = 'shares',isin =_
↪isins,date = '2024-01-01',show_progress=True)
hist_prices.head()
```

Fetching data: 100%| | 3/3 [00:00<00:00, 11.57it/s]

[]:	BOARDID	TRADEDATE	SHORTNAME	SECID	NUMTRADES	VALUE	OPEN	\
0	TQBR	2024-01-03	SBER	50912	5.631305e+09	271.90		
1	TQBR	2024-01-04	SBER	33045	3.218188e+09	274.67		
2	TQBR	2024-01-05	SBER	26234	2.634960e+09	274.30		
3	TQBR	2024-01-08	SBER	70828	5.924626e+09	273.60		
4	TQBR	2024-01-09	SBER	40508	5.601231e+09	276.97		

	LOW	HIGH	LEGALCLOSEPRICE	...	MARKETPRICE2	MARKETPRICE3	\
0	271.00	274.70	273.64	...	273.48	273.48	
1	273.70	275.48	274.04	...	274.42	274.42	
2	272.80	274.69	273.46	...	273.46	273.46	
3	273.53	277.00	275.81	...	275.61	275.61	

```
4 274.71 278.00          276.00 ...          275.71          275.71
```

```

      ADMITTEDQUOTE      MP2VALTRD  MARKETPRICE3TRADESVALUE  ADMITTEDVALUE  WAAVAL  \
0          NaN  4.994800e+09          4.994800e+09          NaN      0
1          NaN  2.822724e+09          2.822724e+09          NaN      0
2          NaN  2.246293e+09          2.246293e+09          NaN      0
3          NaN  5.204970e+09          5.204970e+09          NaN      0
4          NaN  4.962470e+09          4.962470e+09          NaN      0

```

```

      TRADINGSESSION  CURRENCYID  TRENDCLSPR
0                3          SUR          1.38
1                3          SUR         -0.16
2                3          SUR         -0.18
3                3          SUR          1.15
4                3          SUR         -0.53

```

[5 rows x 23 columns]

Also ISS provides the information about best offer and best bid with time lag of 15 minutes for different boards

```
[ ]: iss.market_data(engine = 'stock',market = 'shares',isin = 'SBER')
```

```
[ ]:  SECID  BOARDID  BID  BIDDEPTH  OFFER  OFFERDEPTH  SPREAD  BIDDEPTH  \
0  SBER    SMAL   NaN    NaN    NaN    NaN    0    0
1  SBER    SPEQ   NaN    NaN    NaN    NaN    0    0
2  SBER    TQBR   NaN    NaN    NaN    NaN    0    0

```

```

      OFFERDEPTH  OPEN  ...      SEQNUM      SYSTIME  \
0            0  316.93  ...  20240612000500  2024-06-12 00:05:00
1            0    NaN  ...  20240612000500  2024-06-12 00:05:00
2            0  317.50  ...  20240612000500  2024-06-12 00:05:00

```

```

      CLOSINGAUCTIONPRICE  CLOSINGAUCTIONVOLUME  ISSUECAPITALIZATION  \
0            NaN            NaN            6860332074400
1            NaN            NaN            6860332074400
2            317.9            68340.0            6860332074400

```

```

      ISSUECAPITALIZATION_UPDATETIME  ETFSETTLECURRENCY  VALTODAY_RUR  \
0                23:48:56            NaN            13391
1                23:48:56            NaN              0
2                23:48:56            NaN            6377952743

```

```

      TRADINGSESSION  TRENDISSUECAPITALIZATION
0            NaN            11225212960
1            NaN            11225212960
2            NaN            11225212960

```

[3 rows x 56 columns]

Now, it's unclear what is BIDDEPTH feature. The function - tables_description can help to get description for every table

```
[ ]: descriptions = iss.tables_description(engine = 'stock', market = 'shares')
marketdata_desc = descriptions['marketdata']
marketdata_desc.head(20)
```

```
[ ]:      id      name      short_title \
0   1866      SECID
1   1865    BOARDID
2   1868      BID
3   1869    BIDDEPTH
4   1872    OFFER
5   1873  OFFERDEPTH
6   1914    SPREAD
7   1870  BIDDEPTHT
8   1874  OFFERDEPTHT
9   1876    OPEN
10  1878    LOW
11  1877    HIGH
12  1879    LAST
13  1913  LASTCHANGE      , .
14  1881  LASTCHANGEPRCNT      , %
15  1884      QTY
16  1890    VALUE
17  1891  VALUE_USD      , .
18  1892    WAPRICE      . . .
19  1883  LASTCNGTOLASTWAPRICE

      title  is_ordered  is_system \
0         1          0
1         0          0
2         0          0
3      , ...      0          0
4         0          0
5      , ...      0          0
6         0          0
7         0          0
8         0          0
9         0          0
10        0          0
11        0          0
12        0          0
13      ...      0          0
14      ...      1          0
15      ,      0          0
```

```

16         ,      .      0      0
17         ,      .      0      0
18         ,      .      0      0
19         ...      0      0

```

	is_hidden	trend_by	is_signed	has_percent	type	precision	is_linked
0	0	NaN	0	0	string	NaN	0
1	0	NaN	0	0	string	NaN	0
2	0	NaN	0	0	number	NaN	0
3	0	NaN	0	0	number	0.0	0
4	0	NaN	0	0	number	NaN	0
5	0	NaN	0	0	number	0.0	0
6	0	NaN	0	0	number	NaN	0
7	0	NaN	0	0	number	0.0	0
8	0	NaN	0	0	number	0.0	0
9	0	NaN	0	0	number	NaN	0
10	0	NaN	0	0	number	NaN	0
11	0	NaN	0	0	number	NaN	0
12	0	1915.0	0	0	number	NaN	0
13	0	1913.0	0	0	number	NaN	0
14	0	1881.0	1	1	number	2.0	0
15	0	NaN	0	0	number	0.0	0
16	0	NaN	0	0	number	NaN	0
17	1	NaN	0	0	number	2.0	0
18	0	1893.0	0	0	number	NaN	0
19	1	1883.0	0	0	number	NaN	0

Also ISS provides an information about candels for different instruments.

```

[ ]: # 1-minute candels for SBER from 2024-11-20
candles = iss.candles('stock','shares','SBER','2024-11-20',show_progress=False)
candles.head()

```

```

[ ]:      open   close   high   low      value  volume      begin  \
0  241.82  241.82  241.82  241.82  27286968.8  112840  2024-11-20  09:59:00
1  241.80  241.89  242.28  241.55  166152508.1  686900  2024-11-20  10:00:00
2  241.81  241.73  242.19  241.67  123881188.3  512060  2024-11-20  10:01:00
3  241.72  242.03  242.08  241.60   64168699.3  265340  2024-11-20  10:02:00
4  242.04  241.89  242.28  241.72   79630783.9  329050  2024-11-20  10:03:00

      end
0  2024-11-20  09:59:59
1  2024-11-20  10:00:59
2  2024-11-20  10:01:59
3  2024-11-20  10:02:59
4  2024-11-20  10:03:59

```

ISS also provides information about different indecies at MOEX and their components.

```
[ ]: indices = iss.indices()
indices.head()

index = 'IMOEX'
IMOEX = iss.index_components(index = index, date = '2024-06-11')
IMOEX.head(20)
```

```
[ ]:      indexid  tradedate ticker  shortnames secids  weight  tradingession
0      IMOEX  2024-06-11  AFKS          AFKS    0.89          3
1      IMOEX  2024-06-11  AFLT          AFLT    0.83          3
2      IMOEX  2024-06-11  AGRO    AGRO-    AGRO    0.86          3
3      IMOEX  2024-06-11  ALRS          ALRS    1.34          3
4      IMOEX  2024-06-11  BSPB          BSPB    0.45          3
5      IMOEX  2024-06-11  CBOM          CBOM    0.63          3
6      IMOEX  2024-06-11  CHMF      -    CHMF    2.51          3
7      IMOEX  2024-06-11  ENPG    +    ENPG    0.43          3
8      IMOEX  2024-06-11  FEES          FEES    0.42          3
9      IMOEX  2024-06-11  FIVE    FIVE-    FIVE    1.11          3
10     IMOEX  2024-06-11  FLOT          FLOT    0.60          3
11     IMOEX  2024-06-11  GAZP          GAZP    8.67          3
12     IMOEX  2024-06-11  GLTR    GLTR-    GLTR    0.25          3
13     IMOEX  2024-06-11  GMKN          GMKN    6.10          3
14     IMOEX  2024-06-11  HYDR          HYDR    0.23          3
15     IMOEX  2024-06-11  IRAO          IRAO    1.51          3
16     IMOEX  2024-06-11  LKOH          LKOH   15.19          3
17     IMOEX  2024-06-11  MAGN          MAGN    1.32          3
18     IMOEX  2024-06-11  MGNT          MGNT    1.31          3
19     IMOEX  2024-06-11  MOEX          MOEX    1.33          3
```

Also, ISS MOEX provides the time schedule of bonds' coupons and amortization payments. Let's consider based on following example:

```
[ ]: bonds_market = iss.securities_market(engine = 'stock', market = 'bonds')
bonds_market.head()
```

```
[ ]:      SECID  BOARDID  SHORTNAME  PREVWAPPRICE  YIELDATPREVWAPPRICE  \
0  KG000A3LSC78    TQCB    . 01      98.999          16.99
1  KG000A3LSJ06    TQCB    . 02      97.410          17.20
2  RU000A0JQ7Z2    TQCB   -19      99.830          10.29
3  RU000A0JQAM6    TQCB    . 15     106.800          -5.58
4  RU000A0JQRD9    TQCB   -23      95.530          16.54

      COUPONVALUE  NEXTCOUPON  ACCRUEDINT  PREVPRICE  LOTSIZE  ...  REGNUMBER  \
0           3.88  2024-06-25    3.434903    98.999         1  ...         NaN
1           3.63  2024-06-26    3.101389    97.050         1  ...         NaN
2          39.14  2024-07-08   33.770000    99.860         1  ...  4-19-65045-D
3          93.26  2024-09-15   45.620000   106.800         1  ...  4-15-00739-A
4          39.14  2024-07-18   31.620000    96.050         1  ...  4-23-65045-D
```

	CURRENCYID	ISSUESIZEPLACED	LISTLEVEL	SECTYPE	COUPONPERCENT	OFFERDATE	\
0	SUR	NaN	3	6	15.50	NaN	
1	SUR	NaN	3	6	14.50	NaN	
2	SUR	10000000.0	2	6	7.85	NaN	
3	SUR	7000000.0	1	6	18.50	NaN	
4	SUR	15000000.0	2	6	7.85	NaN	

	SETTLEDATE	LOTVALUE	FACEVALUEONSETTLEDATE
0	2024-06-13	100.0	100.0
1	2024-06-13	100.0	100.0
2	2024-06-13	1000.0	1000.0
3	2024-06-13	1000.0	1000.0
4	2024-06-13	1000.0	1000.0

[5 rows x 40 columns]

```
[ ]: bonds_market.columns
```

```
[ ]: Index(['SECID', 'BOARDID', 'SHORTNAME', 'PREVWAPPRICE', 'YIELDATPREVWAPPRICE',
          'COUPONVALUE', 'NEXTCOUPON', 'ACCRUEDINT', 'PREVPRICE', 'LOTSIZE',
          'FACEVALUE', 'BOARDNAME', 'STATUS', 'MATDATE', 'DECIMALS',
          'COUPONPERIOD', 'ISSUESIZE', 'PREVLEGALCLOSEPRICE', 'PREVDATE',
          'SECNAME', 'REMARKS', 'MARKETCODE', 'INSTRID', 'SECTORID', 'MINSTEP',
          'FACEUNIT', 'BUYBACKPRICE', 'BUYBACKDATE', 'ISIN', 'LATNAME',
          'REGNUMBER', 'CURRENCYID', 'ISSUESIZEPLACED', 'LISTLEVEL', 'SECTYPE',
          'COUPONPERCENT', 'OFFERDATE', 'SETTLEDATE', 'LOTVALUE',
          'FACEVALUEONSETTLEDATE'],
          dtype='object')
```

```
[ ]: url = 'https://iss.moex.com/iss/apps/infogrid/emission/rates.csv?iss.
          ↪dp=comma&iss.df=%25d.%25m.%25Y&iss.tf=%25H:%25M:%25S&iss.dtf=%25d.%25m.
          ↪%25Y%20%25H:%25M:%25S&iss.only=rates&limit=unlimited&lang=ru'
market_full = pd.read_csv(url, skiprows=1, delimiter=';',
          ↪encoding='windows-1251') # Get all tradable bonds
market_full['INN']
```

```
[ ]:
```

```
[ ]: 0      5403
      1     97053
      2     3850
      3     3850
      4     3850
      ...
     3185    10035220
     3186    10035220
```



```

3187    10035220
3188    10035220
3189    10035220
Name: INN, Length: 3190, dtype: int64

```

```

[ ]: isins = bonds_market['SECID'].iloc[:20].to_list()
      bonds_coupons = iss.bonds_coupons(isin = isins)
      bonds_amort = iss.bonds_amort(isin = isins)

```

```

Fetching data: 100%|          | 20/20 [00:01<00:00, 15.98it/s]
Fetching data: 100%|          | 20/20 [00:03<00:00,  6.17it/s]

```

```

[ ]: bonds_coupons.head()

```

```

[ ]:
      isin      name  issuevalue  coupondate  recorddate \
0  RU000A0JS4Z7  .  .  . 21  15000000000  2012-09-11      NaT
1  RU000A0JS4Z7  .  .  . 21  15000000000  2013-03-12      NaT
2  RU000A0JS4Z7  .  .  . 21  15000000000  2013-09-10      NaT
3  RU000A0JS4Z7  .  .  . 21  15000000000  2014-03-11      NaT
4  RU000A0JS4Z7  .  .  . 21  15000000000  2014-09-09      NaT

```

```

      startdate  initialfacevalue  facevalue  faceunit  value  valueprc \
0  2012-03-13                1000        1000      RUB  41.88      8.4
1  2012-09-11                1000        1000      RUB  41.88      8.4
2  2013-03-12                1000        1000      RUB  41.88      8.4
3  2013-09-10                1000        1000      RUB  41.88      8.4
4  2014-03-11                1000        1000      RUB  41.88      8.4

```

```

      value_rub      secid  primary_boardid
0      41.88  RU000A0JS4Z7      TQCB
1      41.88  RU000A0JS4Z7      TQCB
2      41.88  RU000A0JS4Z7      TQCB
3      41.88  RU000A0JS4Z7      TQCB
4      41.88  RU000A0JS4Z7      TQCB

```

```

[ ]: bonds_amort.head()

```

```

[ ]:
      isin      name  issuevalue  amortdate \
0  KG000A3LSJ06      02  1000000000  2025-12-26
0  RU000A0JQAM6  .  ( )  .  15  7000000000  2028-09-15
0  RU000A0JRZ74      -03  18000000000  2025-07-16
0  RU000A0JS4J1  .  .  . 03  5000000000  2032-02-06
0  KG000A3LSC78      01  1000000000  2025-12-25

```

```

      facevalue  initialfacevalue  faceunit  valueprc  value  value_rub \
0         100                100      RUB    100.0    100    100.00
0        1000                1000      RUB    100.0   1000   1000.00
0        1000                1000      RUB    100.0   1000   1000.00

```

0	1000	1000	RUB	100.0	1000	1000.00
0	100	100	KGS	100.0	100	102.33

	data_source	secid	primary_boardid
0	maturity	KG000A3LSJ06	TQCB
0	maturity	RU000AOJQAM6	TQCB
0	maturity	RU000AOJRZ74	TQCB
0	maturity	RU000A0JS4J1	TQCB
0	maturity	KG000A3LSC78	TQCB

The package provides an opportunity to operate with custom urls from ISS.

```
[ ]: # Accrued interest on the end of month
# Single page version without
url = 'https://iss.moex.com/iss/statistics/engines/stock/markets/bonds/
      monthendaccints.html'
iss.iss_url(url)
```

[]:	tradedate	secid	name	shortname	\
0	2024-03-31	AT0000A2UF10	Raiffeisen Bank Int AG	Raiff BIAG	
1	2024-03-31	CH0248531110	VTB CAPITAL S.A. 24 CHF	VTB-24 CHF	
2	2024-03-31	CH0419041618	RZD CAPITAL PLC 0.898 03/10/25	RZD-25 CHF	
3	2024-03-31	CH0419041634	RZD CAPITAL PLC 03/04/28	RZD-28 CHF	
4	2024-03-31	CH1100259816	RZD Capital PLC VAR UNDT	RZD-p CHF	
...	
3033	2024-03-31	XS2420560869	BCS SP Plc Series 239	BCS01/25	
3034	2024-03-31	XS2423361190	BCS SP Plc Series 240	BCS02/27-4	
3035	2024-03-31	XS2429208486	BCS SP Plc Series 243	BCS05/25	
3036	2024-03-31	XS2439218640	BCS SP Plc Series 244	BCS06/25-B	
3037	2024-03-31	XS2446844321	BCS SP Plc Series 245	BCS06/25-C	

	regnumber	accint
0	NaN	NaN
1	NaN	88.803125
2	NaN	22.200556
3	NaN	59.418056
4	NaN	120.225694
...
3033	NaN	NaN
3034	NaN	NaN
3035	NaN	NaN
3036	NaN	NaN
3037	NaN	NaN

[3038 rows x 6 columns]

```
[ ]: # Market Data of Yields
# Example of custom url function iterable over defined number of parameters
url_func = lambda isin: f'https://iss.moex.com/iss/engines/stock/markets/bonds/
↳securities/{isin}.html?iss.only=marketdata_yields'
bonds_market_yields = iss.iss_url(url = url_func, parameters = isins)
bonds_market_yields.head()
```

```
[ ]:
SECID BOARDID PRICE YELDDATE ZCYCMOMENT \
0 RU000A0JS4K9 TQCB 105.99 2032-02-06 2024-06-11 16:36:06
0 RU000A0JS4Z7 TQCB 95.20 2032-02-17 2024-06-11 14:16:51
0 RU000A0JS4L7 TQCB 116.30 2032-02-06 2024-06-10 18:39:59
0 KG000A3LSC78 TQCB 99.00 2025-12-25 2024-06-11 10:17:53
0 RU000A0JSQ58 TQCB 99.99 2024-07-31 2024-06-11 11:39:58

YELDDATETYPE EFFECTIVEYIELD DURATION ZSPREADBP GSPREADBP ... \
0 MATDATE 8.9387 1992 -618 -629 ...
0 MATDATE 10.4502 1989 -463 -471 ...
0 MATDATE 7.1708 2033 -794 -804 ...
0 MATDATE 16.9939 492 106 105 ...
0 OFFER 7.9261 47 -702 -702 ...

DURATIONWAPRICE IR ICPI BEI CBR YIELDTOOFFER YIELDLASTCOUPON \
0 1992.0 NaN NaN NaN NaN NaN NaN
0 1990.0 NaN NaN NaN NaN NaN NaN
0 2033.0 NaN NaN NaN NaN NaN NaN
0 NaN NaN NaN NaN NaN NaN NaN
0 47.0 NaN NaN NaN NaN NaN 7.9261 NaN

TRADEMOMENT SEQNUM SYSTIME
0 2024-06-11 16:36:37 20240611165100 2024-06-11 16:51:00
0 2024-06-11 14:16:52 20240611143100 2024-06-11 14:31:00
0 2024-06-06 23:59:59 20240611062613 2024-06-11 06:26:13
0 2024-06-11 10:17:57 20240611103200 2024-06-11 10:32:00
0 2024-06-11 11:47:30 20240611120200 2024-06-11 12:02:00
```

[5 rows x 22 columns]

```
[ ]: date = '2024-01-01'
url_func = lambda isin: f'https://iss.moex.com/iss/history/engines/stock/
↳markets/bonds/yields/{isin}.html?from={date}&limit=100'
bonds_market_yields_history = iss.iss_url(url = url_func,
parameters = isins,
pages = True, #to iterate over all
↳page entries

show_progress = True,
records_per_pages = 100 )#depicted
↳number of rows at one iteration)
```

```
bonds_market_yields_history.head()
```

Fetching data: 100%| | 20/20 [00:01<00:00, 14.39it/s]

```
[ ]:   TRADEDATE      SECID BOARDID YELDDATE YELDDATETYPE  PRICE  ACCINT  \
0  2024-01-03  RU000A0JRZ74    TQCB      NaN      NaN    NaN    NaN
1  2024-01-04  RU000A0JRZ74    TQCB      NaN      NaN    NaN    NaN
2  2024-01-05  RU000A0JRZ74    TQCB      NaN      NaN    NaN    NaN
3  2024-01-08  RU000A0JRZ74    TQCB      NaN      NaN    NaN    NaN
4  2024-01-09  RU000A0JRZ74    TQCB      NaN      NaN    NaN    NaN

   ZCYCMOMENT  EFFECTIVEYIELD  DURATION  ...  WAPRICE  EFFECTIVEYIELDWAPRICE  \
0          NaN              NaN        NaN  ...    NaN                      NaN
1          NaN              NaN        NaN  ...    NaN                      NaN
2          NaN              NaN        NaN  ...    NaN                      NaN
3          NaN              NaN        NaN  ...    NaN                      NaN
4          NaN              NaN        NaN  ...    NaN                      NaN

   DURATIONWAPRICE  IR  ICPI  BEI  CBR  YIELDTOOFFER  YIELDLASTCOUPON  \
0              NaN NaN   NaN  NaN  NaN          NaN              NaN
1              NaN NaN   NaN  NaN  NaN          NaN              NaN
2              NaN NaN   NaN  NaN  NaN          NaN              NaN
3              NaN NaN   NaN  NaN  NaN          NaN              NaN
4              NaN NaN   NaN  NaN  NaN          NaN              NaN

   TRADEMOMENT
0          NaN
1          NaN
2          NaN
3          NaN
4          NaN
```

[5 rows x 22 columns]

The package has docstring description. For convince use the following query to get information about function and variables

```
[ ]: ?iss.iss_url
```

1.0.1 Advantages of package

The package has several advantages:

1. Simplifies work with ISS MOEX, as most frequent queries are defined in the easy reading functions
2. Significantly speed up the urls requests, because of asynchronous framework usage
3. Allows to work with custom ISS requests, accounting for all specificity of the requests

1.0.2 Time difference with usual requests

The comparison will be done with `pd.read_html()` method, which is the easiest method to interact with html tables.

The design of experiment will be following:

1. 20 stocks from IMOEX
2. History prices from 2023-01-01

```
[ ]: IMOEX.sort_values(by = 'weight',ascending = False, inplace = True)
stocks = IMOEX['secids'].iloc[:20].to_list()
IMOEX.head(20)
```

```
[ ]:   indexid  tradedate ticker  shortnames secids  weight  tradingseason
16   IMOEX  2024-06-11  LKOH           LKOH   15.19           3
34   IMOEX  2024-06-11  SBER           SBER   14.05           3
11   IMOEX  2024-06-11  GAZP           GAZP    8.67           3
13   IMOEX  2024-06-11  GMKN           GMKN    6.10           3
41   IMOEX  2024-06-11  TATN      3  TATN     6.00           3
25   IMOEX  2024-06-11  NVTK           NVTK    3.65           3
40   IMOEX  2024-06-11  SNGSP     -  SNGSP    3.43           3
39   IMOEX  2024-06-11  SNGS           SNGS    3.21           3
35   IMOEX  2024-06-11  SBERP     -  SBERP    2.71           3
29   IMOEX  2024-06-11  PLZL           PLZL    2.60           3
6    IMOEX  2024-06-11  CHMF     -  CHMF    2.51           3
31   IMOEX  2024-06-11  ROSN           ROSN    2.34           3
28   IMOEX  2024-06-11  PIKK           PIKK    1.84           3
24   IMOEX  2024-06-11  NLMK           NLMK    1.67           3
15   IMOEX  2024-06-11  IRAO           IRAO    1.51           3
33   IMOEX  2024-06-11  RUAL           RUAL    1.45           3
26   IMOEX  2024-06-11  OZON  OZON-  OZON    1.40           3
3    IMOEX  2024-06-11  ALRS           ALRS    1.34           3
19   IMOEX  2024-06-11  MOEX           MOEX    1.33           3
17   IMOEX  2024-06-11  MAGN           MAGN    1.32           3
```

```
[ ]: ## Function for downloading shares' history in standard manner
import pandas as pd
from tqdm import tqdm
def shares_history(isin):

    df_full = pd.DataFrame()
    cond = True
    k=0
    while cond:
        url = 'https://iss.moex.com/iss/history/engines/stock/markets/shares/
↪securities/{}.html?iss.
↪only=history&marketprice_board=1&from=2023-01-01&limit=100&start={}'.
↪format(isin,k)
```

```

df = pd.read_html(url,encoding = 'utf-8')[0]
df.columns = [i.split(' ')[0] for i in df.columns]
if df.empty:
    cond = False

else:
    df_full = pd.concat([df,df_full],axis=0)
    k+=100
df_full = df_full.drop_duplicates()
return df_full

```

[]: *## Execution of standard download*

```

import time

start_time_pd = time.time()
usual_history = pd.DataFrame()
for j in tqdm(range(len(stocks))):
    stock = stocks[j]
    stock_df = shares_history(stock)
    usual_history = pd.concat([usual_history,stock_df],axis=0)
end_time_pd = time.time() - start_time_pd

```

100%| | 20/20 [00:27<00:00, 1.36s/it]

[]: *## Execution of package function*

```

start_time_lib = time.time()
hist_prices = iss.history_prices(engine = 'stock',market = 'shares',isin = stocks,
    ↪date = '2023-01-01',show_progress=True)
hist_prices.head()

end_time_lib = time.time() - start_time_lib

```

Fetching data: 100%| | 20/20 [00:03<00:00, 6.43it/s]

```

[ ]: print(f'Execution time of standard requests: {round(end_time_pd,3)} seconds')
print(f'Execution time of package function: {round(end_time_lib,3)} seconds')

```

Execution time of standard requests: 27.137 seconds

Execution time of package function: 3.114 seconds

Thus, the package function is **9 times faster** than the standard request. This advantage was reached by using asynchronous framework