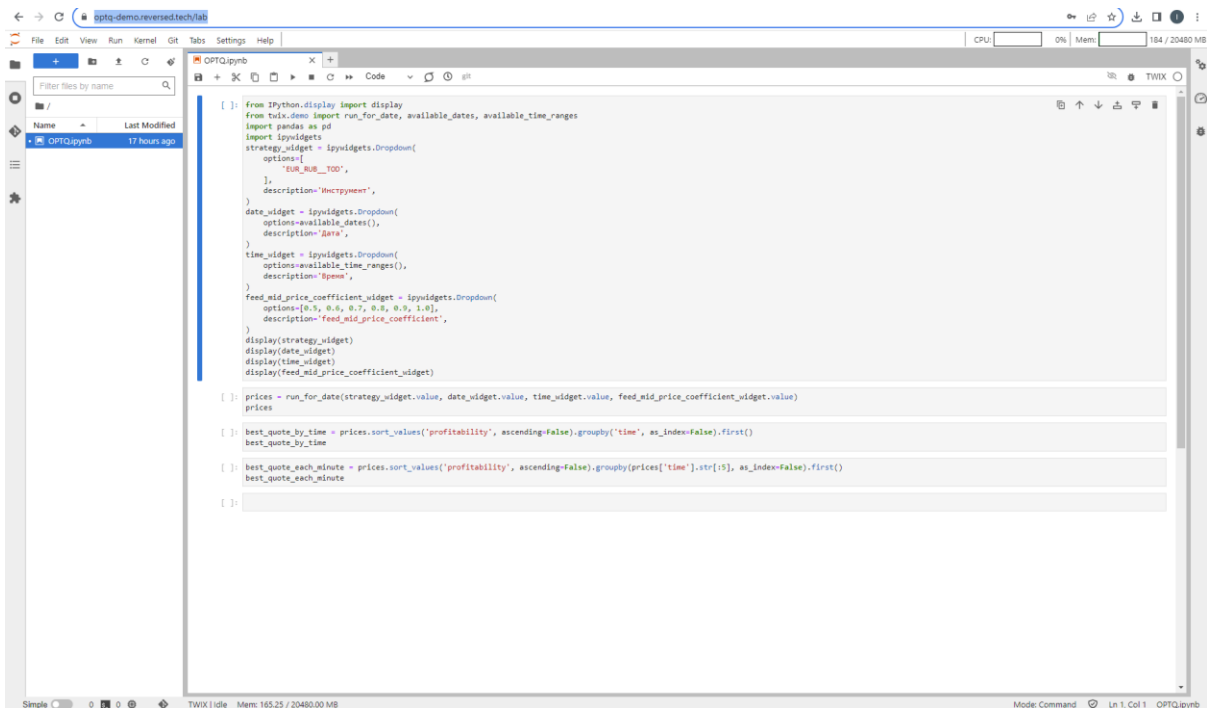


Инструкция по эксплуатации экземпляра ПО «ОРТQ / ОПТИКЬЮ», предоставленного для проведения экспертной проверки

Программа «ОРТQ / ОПТИКЬЮ» (далее Программа) представляет собой дистрибутив (файл tar.gz), внутри которого находится код, написанный на языке C++. Данная программа должна быть встроена в ПО клиента для анализа биржевых данных или торговли на Московской Бирже. Для эксплуатации ПО требуется иметь доступ до данных с биржевой информацией. Для проведения экспертной проверки предоставлен удалённый доступ до демонстрационной платформы, на которой развёрнута скомпилированная версия Программы и реализован доступ до биржевых данных за июль 2023 года. Таким образом, демонстрируется возможная работа Программы в комплексе с клиентским ПО для анализа биржевых данных.

Эксперт получает возможность запуска ПО на различных торговых инструментах, с различными наборами параметров на выбранных датах. Интерфейс демонстрационной платформы является стандартным Jupyter Notebook под названием ОРТQ.ipynb. Подробнее о функционале Jupyter Notebook можно прочесть, например, [здесь](#).

Демонстрационная платформа перед стартом работы выглядит следующим образом:



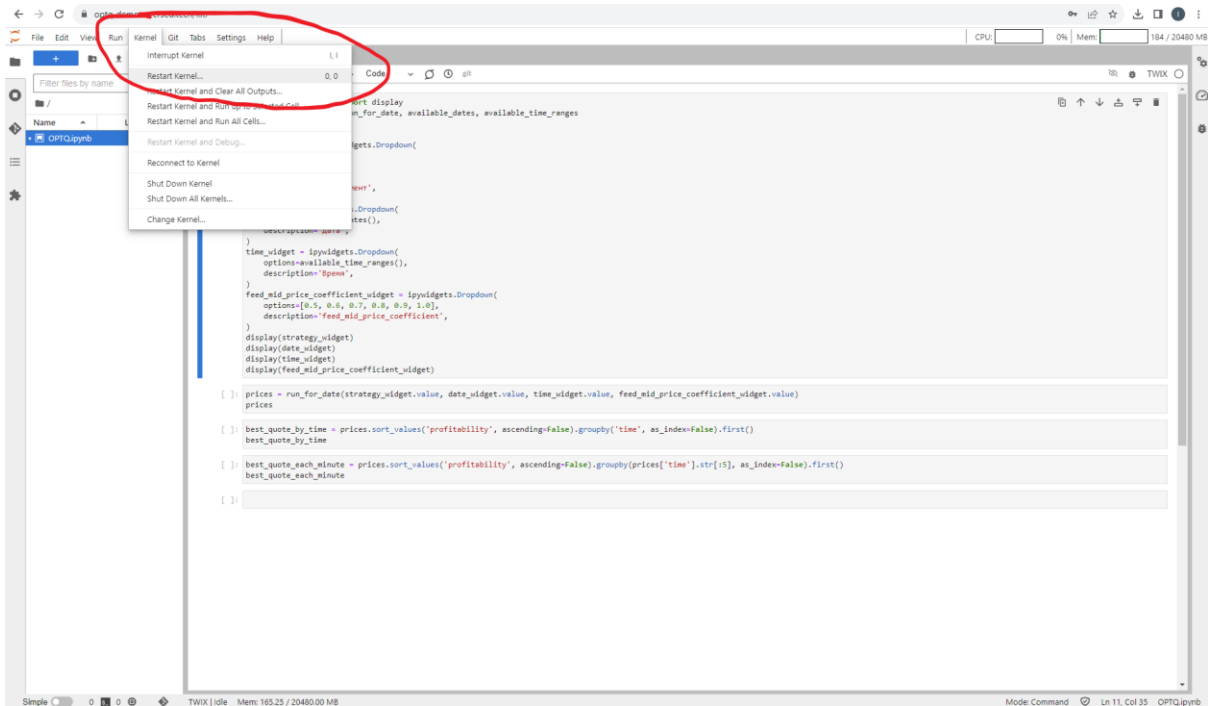
```
from IPython.display import display
from twix_demo import run_for_date, available_dates, available_time_ranges
import pandas as pd
import ipynbwidgets
strategy_widget = ipynbwidgets.Dropdown(
    options=[
        "EUR_RUB__TOD",
    ],
    description="Инструмент",
)
date_widget = ipynbwidgets.Dropdown(
    options=available_dates(),
    description="Дата",
)
time_widget = ipynbwidgets.Dropdown(
    options=available_time_ranges(),
    description="Время",
)
feed_mid_price_coefficient_widget = ipynbwidgets.Dropdown(
    options=[0.5, 0.6, 0.7, 0.8, 0.9, 1.0],
    description="feed_mid_price_coefficient",
)
display(strategy_widget)
display(date_widget)
display(time_widget)
display(feed_mid_price_coefficient_widget)

prices = run_for_date(strategy_widget.value, date_widget.value, time_widget.value, feed_mid_price_coefficient_widget.value)
prices

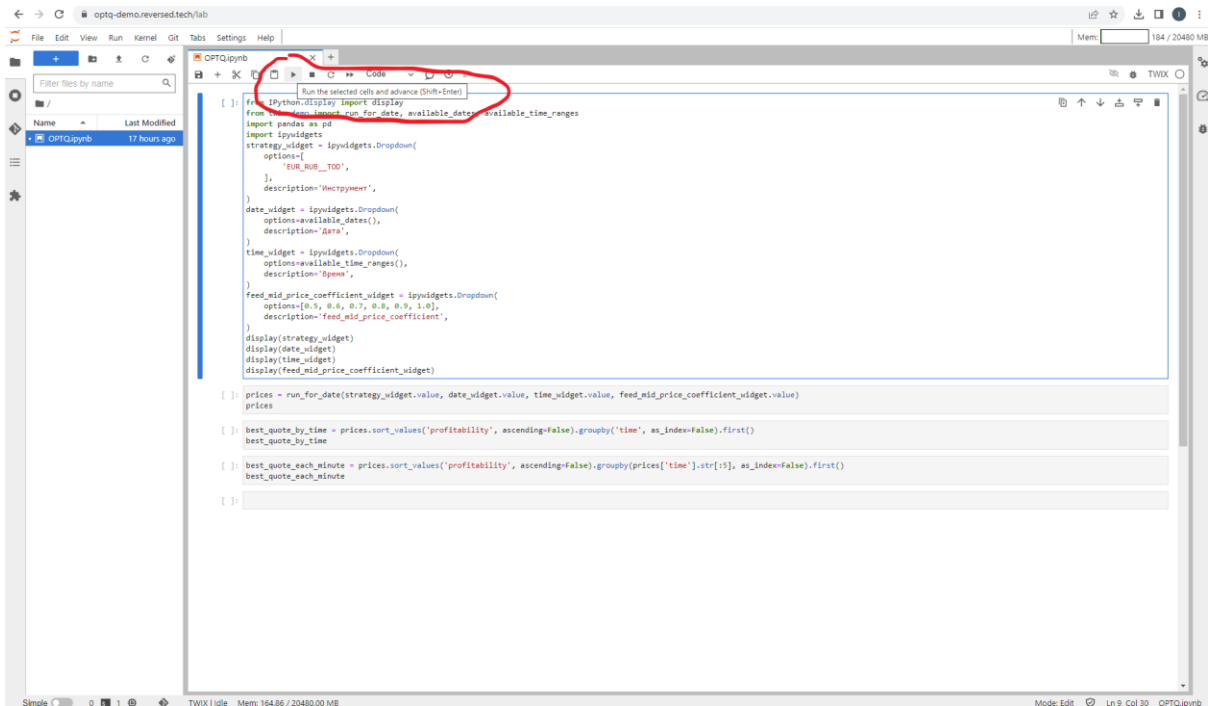
best_quote_by_time = prices.sort_values('profitability', ascending=False).groupby('time', as_index=False).first()
best_quote_by_time

best_quote_each_minute = prices.sort_values('profitability', ascending=False).groupby(prices['time'].str[5], as_index=False).first()
best_quote_each_minute
```

В случае, если доступ до демонстрационной платформы с использованием заданного пароля будет осуществлять несколько экспертов, при очередном запуске платформы могут быть видны результаты предыдущих запусков. В таком случае, требуется перед стартом работы очистить ноутбук командой Kernel → Restart Kernel and Clear All Outputs...



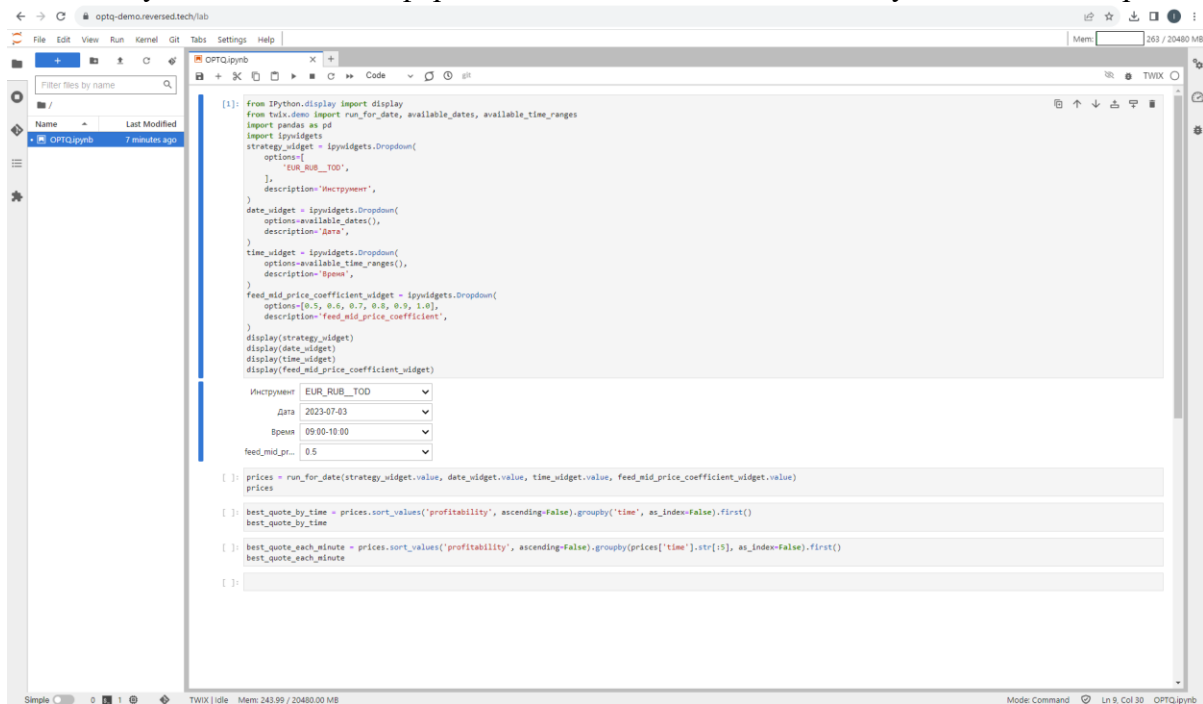
Первая клетка Jupyter ноутбука OPTQ.ipynb задаёт параметры для запуска ПО. Для запуска каждой клетки Jupyter ноутбука требуется нажать Shift+Enter или кнопку Play.



Перед запуском ПО требуется заполнить несколько форм, определяющих параметры, с которыми будет осуществляться запуск.

- **Инструмент.** Эксперту предлагается совершить выбор одного из доступных на платформе инструментов валютного рынка Московской Биржи. ПО будет получать биржевую информацию по всем инструментам валютного рынка и на их основе оценивать профитность ценовых уровней для котирования на выбранном инструменте. Доступен выбор одной из двух опций: 1) торговля евро за российские рубли с расчётами T0 (инструмент EUR_RUB_TOD); 2) торговля евро за российские рубли с расчётами T1 (инструмент EUR_RUB_TOM)
- **Дата.** Доступен список дней из июля 2023 года, когда осуществлялась биржевая торговля выбранными инструментами на площадках ПАО «Московская биржа».
- **Время.** Промежуток времени, для которого осуществляется расчёт. После запуска ПО будет сформирована таблица со значениями оценок профитности котировок выбранного инструмента за заданный промежуток времени.
- **feed_mid_price_coefficient.** Параметр ПО, отвечающий за то, насколько сильно на расчёт профитности влияют котировки по другим, скоррелированным инструментам.

По умолчанию формы заполнены следующим образом:



```
[1]: from IPython.display import display
from twix.demo import run_for_date, available_dates, available_time_ranges
import pandas as pd
import ipynbwidgets
strategy_widget = ipynbwidgets.Dropdown(
    options=[
        'EUR_RUB_TOD',
    ],
    description='Инструмент',
)
date_widget = ipynbwidgets.Dropdown(
    options=available_dates(),
    description='Дата',
)
time_widget = ipynbwidgets.Dropdown(
    options=available_time_ranges(),
    description='Время',
)
feed_mid_price_coefficient_widget = ipynbwidgets.Dropdown(
    options=[0.5, 0.6, 0.7, 0.8, 0.9, 1.0],
    description='feed_mid_price_coefficient',
)
display(strategy_widget)
display(date_widget)
display(time_widget)
display(feed_mid_price_coefficient_widget)

Инструмент EUR_RUB_TOD
Дата 2023-07-03
Время 09:00-10:00
feed_mid_pr_ 0.5

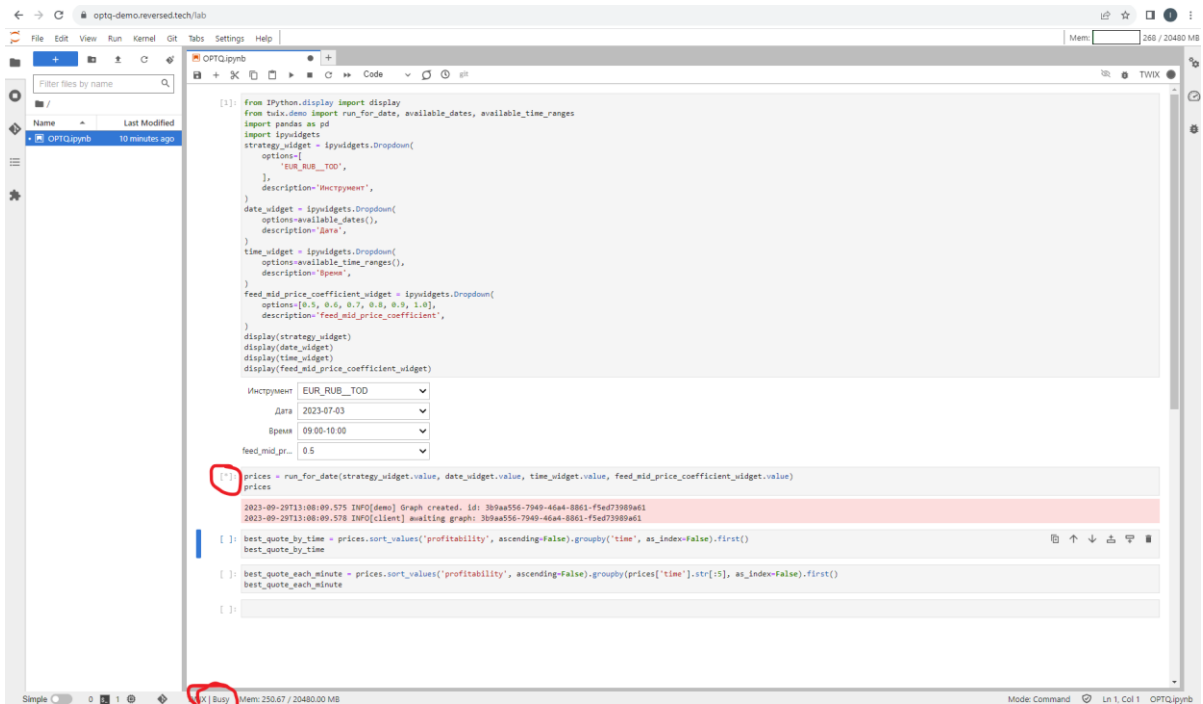
[ ]: prices = run_for_date(strategy_widget.value, date_widget.value, time_widget.value, feed_mid_price_coefficient_widget.value)
prices

[ ]: best_quote_by_time = prices.sort_values('profitability', ascending=False).groupby('time', as_index=False).first()
best_quote_by_time

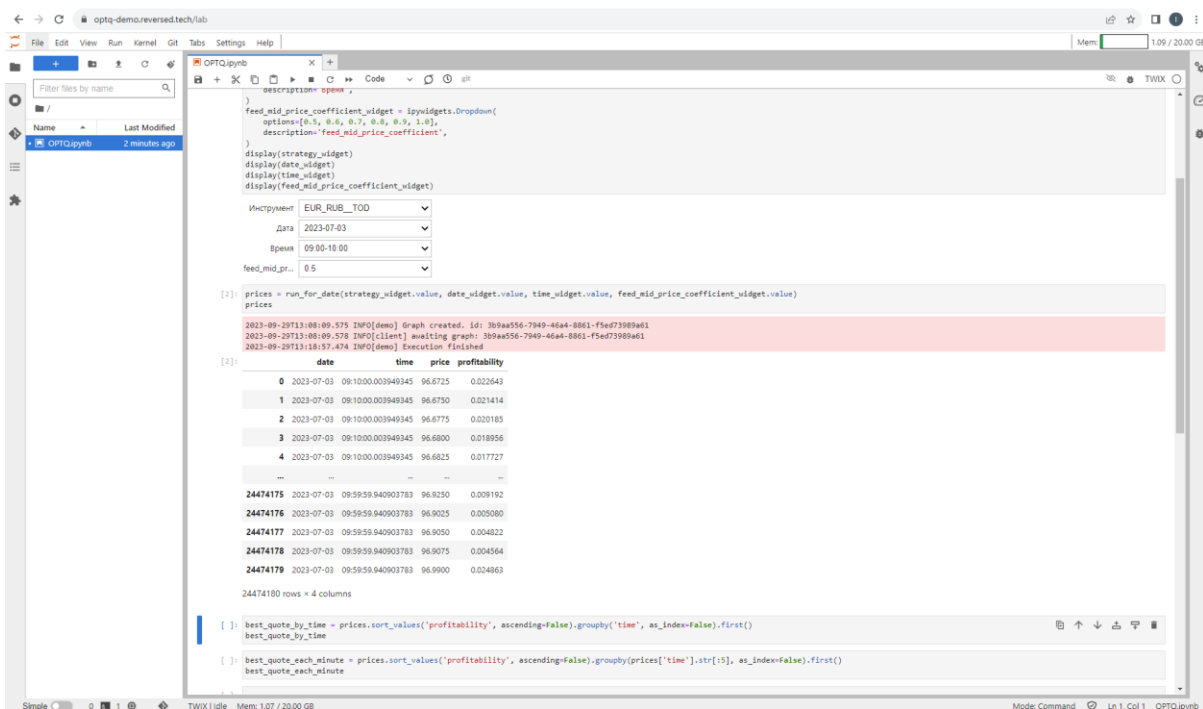
[ ]: best_quote_each_minute = prices.sort_values('profitability', ascending=False).groupby(prices['time'].str[15], as_index=False).first()
best_quote_each_minute

[ ]:
```

Вторая клетка Jupyter ноутбука OPTQ.ipynb запускает ПО, которое получает биржевую информацию за указанный промежуток времени и выдаёт расчёт оценок прибыльности котировок выбранного инструмента после каждого обновления биржевой информации. Расчёт оценок занимает несколько минут, т. к. в течение торгового дня происходят сотни тысяч обновлений биржевой информации. При проведении расчётов демонстрационная платформа выглядит следующим образом:



Результатом работы ПО является таблица со значениями оценок прибыльности котировок выбранного инструмента за заданный промежуток времени. Полученная таблица это pandas.DataFrame, подробнее о данном функционале можно прочесть [здесь](#).



В третьей и четвертой клетках Jupyter ноутбука OPTQ.ipynb добавлены примеры возможного дальнейшего использования полученного набора оценок. Так, в клетке 3 показана котировка с наивысшей оценкой прибыльности после каждого обновления биржевой информации, а в клетке 4 -- котировка с наивысшей оценкой прибыльности за каждую минуту. При желании эксперты имеют возможность провести собственный анализ таблицы со значениями предсказаний, используя стандартный инструмент Jupyter notebook.

```
[1]: best_quote_by_time = prices.sort_values('profitability', ascending=False).groupby('time', as_index=False).first()
best_quote_by_time
```

	time	date	price	profitability
0	09:10:00.003949345	2023-07-03	96.6725	0.022643
1	09:10:00.004175491	2023-07-03	96.6725	0.025018
2	09:10:00.004193289	2023-07-03	96.6725	0.025018
3	09:10:00.004778899	2023-07-03	96.6725	0.025018
4	09:10:00.004829511	2023-07-03	96.6725	0.025018
...
956088	09:59:59.907236933	2023-07-03	96.9300	0.031781
956089	09:59:59.908322123	2023-07-03	96.9300	0.031781
956090	09:59:59.909104081	2023-07-03	96.9300	0.031780
956091	09:59:59.909555145	2023-07-03	96.9300	0.031780
956092	09:59:59.940903783	2023-07-03	96.9300	0.031785

```
[1]: best_quote_each_minute = prices.sort_values('profitability', ascending=False).groupby(prices['time'].str[5:], as_index=False).first()
best_quote_each_minute
```

```
[4]: best_quote_each_minute = prices.sort_values('profitability', ascending=False).groupby(prices['time'].str[5:], as_index=False).first()
best_quote_each_minute
```

	date	time	price	profitability
0	2023-07-03	09:10:53.541719481	96.8100	0.136160
1	2023-07-03	09:11:32.5348979045	96.7325	0.147461
2	2023-07-03	09:12:03.767886511	96.6775	0.090088
3	2023-07-03	09:13:58.544109703	96.5975	0.071901
4	2023-07-03	09:14:59.742562905	96.5975	0.068083
5	2023-07-03	09:15:29.353894055	96.7025	0.106883
6	2023-07-03	09:16:16.915510893	96.7350	0.107300
7	2023-07-03	09:17:56.412444711	96.6350	0.076359
8	2023-07-03	09:18:53.581078253	96.6350	0.096863
9	2023-07-03	09:19:39.315311543	96.6350	0.120898
10	2023-07-03	09:20:48.69425663	96.7150	0.153277
11	2023-07-03	09:21:09.633881623	96.7150	0.133095
12	2023-07-03	09:22:18.167724505	96.7475	0.131539
13	2023-07-03	09:23:12.772703379	96.7300	0.157369
14	2023-07-03	09:24:30.673101173	96.7100	0.129655
15	2023-07-03	09:25:54.828960791	96.6475	0.062916
16	2023-07-03	09:26:11.659923657	96.6475	0.059431
17	2023-07-03	09:27:43.599471063	96.5575	0.062091
18	2023-07-03	09:28:58.09645291	96.7300	0.095189
19	2023-07-03	09:29:14.275073733	96.8125	0.124134
20	2023-07-03	09:30:42.999609973	96.8725	0.118462
21	2023-07-03	09:31:16.657488965	96.9025	0.133248
22	2023-07-03	09:32:41.350315801	96.8825	0.128009
23	2023-07-03	09:33:11.250087841	96.8825	0.138961
24	2023-07-03	09:34:14.818373137	96.9075	0.142270
25	2023-07-03	09:35:01.058158859	96.9075	0.099947
26	2023-07-03	09:36:09.261615611	97.0075	0.101486
27	2023-07-03	09:37:34.677480229	97.0475	0.062568
28	2023-07-03	09:38:06.556783259	97.2475	0.207571

Напоминаем, что в случае, если доступ до демонстрационной платформы с использованием заданного пароля будут осуществлять несколько экспертов, то каждому из экспертов после завершения работы требуется сбросить все результаты запросов с помощью команды Kernel → Restart Kernel and Clear All Outputs...:

