

Kaspersky MLAD:
 ?????????????? ??????????????
 API

1. ?????????? ??????????

?????? ???????? ????????? ???????? ???????????? ? ???????????? API Kaspersky MLAD ??? ??????????
 ??????? (?????????? ??????, ????????????? ?? ?????????????? ? ??????????? ??????? ?????????).

2. ????? ?????????

?? ?????? ?????????? ??????? ????????????? ? Kaspersky MLAD ?? ????????????? ????????????? ???????? API,
 ?????? ?????????? ?????? ????? ?? ?????????? ??????? ?? ?????????? ??????????. ??? ?? ?????? ? ??????????
 ?????????????????? API, ?????????????????? ?????????????????? ?????????????, ? ?????????? ??? ?? ?????? ?????????????????? ???????.

?????????: ?????????? ? ?????? ?????????? ?????????? ?????????????? ?? ?????????? ??????????????
 ?????????????? Kaspersky MLAD, ? ?????????????????? ?? ?????? ?????????????????????? ??? ???
 ?????????????????? ??? ?????????????????? ? ?????????? ?????????????.

3. ???????????



????? ???????, ?? ? ?????????? ?????????? ?????? ?????? ?????????? ?????????????????? ?????? ??
 ?????? ????????? API.

???? ????? ????????? ????????? ????????? python ??? ????????? ?????????. ?? ????? ??? ??????????
 ????????? ????????????????????? ?????.

?????????? ???? ???????

????? ????????? ? ?????????? ???, ?????? ?? ?? ????????? ?????????? ?????????????
 ????????, ?? ?????? ??? ?? (?? ??????, ?? ????? ?? ?????????).

??????????, ??? ? ??? ???? ????? ?????????????, ?? ???? ?????????? ????? ????????????? ???????.

???????? ?????????? ??????????:

“ - ????????? ???????? ?????? ? MLAD
- ?????????? "?????????????? ?????????????? / developer tools" (F12 ??? ?????? ????????)

```

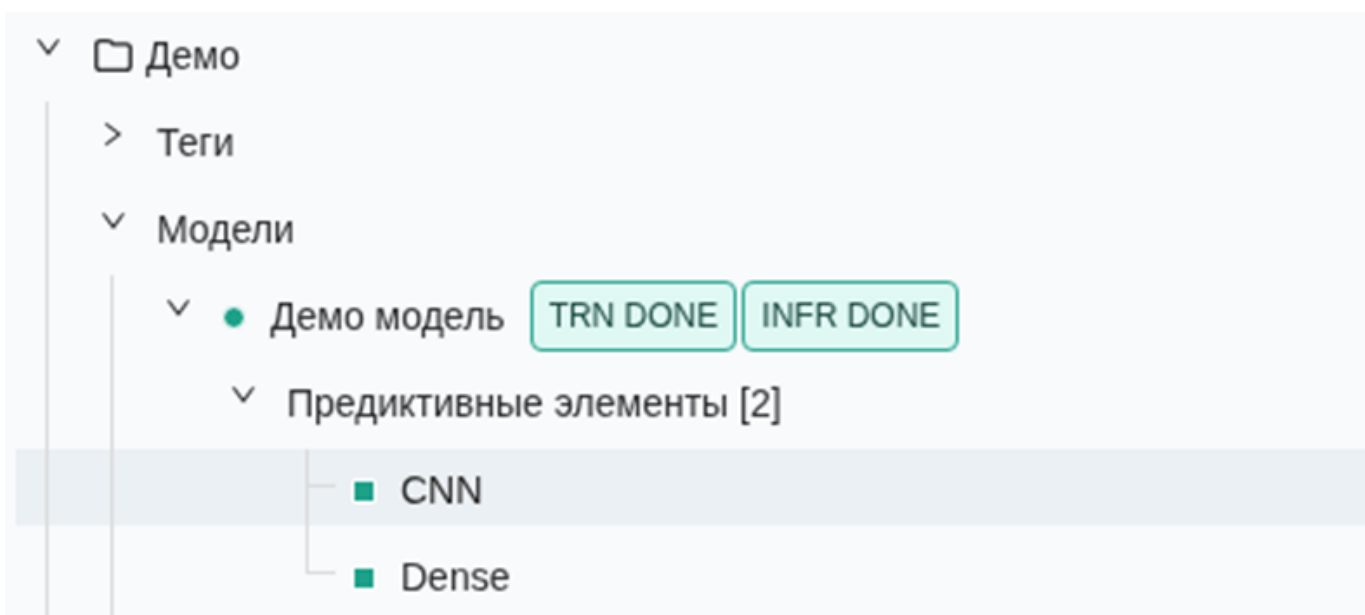
???? ? ?????? ?????? ?????????? -> "????????????? ??? / Inspect")
- ? ?????????????? ?????????????? ?????????? ?????????? "Network" / "?????"
- ??????? ?????????? ??????? "?????????" "Preserve log", ?????? ?? ?????????? ?????????? ??????????
- ?????????????????? ??? ?????? ??????????????????
- ????? ? ????? ??????? ?????????????? (/api/authorize/)
- ? "payload" ?????? ?????????? ???? ???????, ?????????? ?? ?????? ?????????????????? ???
????????????? ? ?????????????? API "?????????".
- ??????? ?????? ? ?????? ?????, ??????????:
"7DFcwe0I6Ult8w422gQgWg2TPUXKPYXIkRFIWvhkrPKAs8kfWcInbHo7UYiKMVYUa/rHQ
7K7GtEvP8S2ZLJQAw=="
- ?????????????? ???? ??? ? ?????????????? 'USER_PASSWORD'? ?????????? ??????????

```

?????? ??????? UUID

?????? ??? UUID ??????? ? ??????? ?????? ??????????? ?????? ?????????????? ? ???-????????????? MLAD.

?????????? ??????? ?????????????????? ?????????? ? ??????? ??????????:



?????? ?????? ?????? ??????????????? UUID ?? ?????????? ??????????:

https://localhost/models_constructor/model_element/ca6e5966-1c7f-11f1-8bb3-72a1019c5619/

?????????????, ?????? ??????? ?????? ?????????? ??????:

https://localhost/models_constructor/tag/76/

????????? ?????????? ?????? ? ?????????????? ?????? API

????? ?????????? ??????? ?????????? ??? ?????????? ?????????? ?????? ? ?????????????? ??????? ??????????:

```

“ # импортируем нужные модули
import requests
import json
import logging
import time
from datetime import datetime

```

```
# учетная запись пользователя, от имени которого происходит
подключение
USER_EMAIL = test@admin.com
# пароль в хешированном виде - см. инструкцию по извлечению хеша
пароля выше
USER_PASSWORD =
"z24y0qIWIn/zfnbl8YJlO3K+E38GcKtY8kRTyU2moNagGLUABgl9HeJrHWWblGdOE
9cNmey2s3szM0ly3v9l4w==" #hashed password
# IP адрес или имя хоста Kaspersky MLAD
MLAD_HOST = "192.168.179.131"

requests.packages.urllib3.disable_warnings()

logging.basicConfig(
    level=logging.INFO,
    format='%(asctime)s - %(levelname)s - %(message)s',
    datefmt='%Y-%m-%d %H:%M:%S'
)
logging.info(f"{20*'-'}START{20*'-'}")

auth_response = requests.post(
    url=f'https://{MLAD_HOST}/api/authorize/',
    json={
        'username': USER_EMAIL,
        'password': USER_PASSWORD,
    },
    verify=False,
)
logging.info(json.dumps(auth_response.json(), indent=4))

# timestamp - момент времени, от которого все будет рассчитываться,
задается в миллисекундах
# для отправки запроса в реальном времени можно брать текущую метку
времени:
# int(time()*1000)

timestamp_string = '2026-03-12 12:00:00'
timestamp_format = '%Y-%m-%d %H:%M:%S'
timestamp_obj = datetime.strptime(timestamp_string, timestamp_format)
timestamp_ms = int(timestamp_obj.timestamp() * 1000)
logging.info(f'{timestamp_obj} = {timestamp_ms} ms')

# period (ms)- Период, за который планируется достать данные из истории
в миллисекундах. На странице истории выбирается из выпадающего
```

списка.

если мы хотим значение за конкретную метку времени, то можно поставить значение значительно ниже дискретности, с которой приходят данные, например, 10мс

interval (ms) - интервал группировки (ресемпл) данных. Если данные приходят очень часто, то позволит группировать данные по окнам

Если мы хотим получать одно значение раз в 5 минут, но не знаем точные метки времени, можно отправлять запрос каждые ($T < 5\text{min}$) минут и указать период равный T. В таком случае будет возвращаться либо одно значение, либо ни одного.

timestamp_position - Позиционирование центра времени, т.к. timestamp - обычно `center` т.е. значения будут браться по обе стороны от него.

Т.е. если мы говорим "дай данные за 1 минуту", значит возьмется 30 секунд до заданного timestamp и 30 секунд после. В целом можно не менять.

element_uuid_list - UUID элементов моделей, чьи артефакты необходимо получить наравне с тегами

```
request_data = {
    'period': 300000,
    'interval': 150000,
    'tag_id_list': [76], # список с номерами тегов – можно посмотреть во вкладке
    Активы
    'timestamp_position': 'center',
    'timestamp': timestamp_ms,
    # element_uuid_list – список с UID моделей, из которых мы хотим брать
    предсказания для тегов из списка tag_id_list. Можно посмотреть в адресной
    строке браузера после выбора соответствующей модели в веб-интерфейсе
    – см инструкцию выше
    'element_uuid_list': [
        'ca6e5966-1c7f-11f1-8bb3-72a1019c5619'
    ],
}
```

```
history_data = requests.post(
    url=f'https://{MLAD_HOST}/api/trends/',
    json=request_data,
    verify=False,
    headers={
        'Token': auth_response.json()['token'],
    }
)
```

```
logging.info(json.dumps(history_data.json(), indent=4))
```

```
# Пример вывода:
# 2026-03-13 13:25:29 - INFO - 2026-03-12 12:00:00 = 1773306000000 ms
# 2026-03-13 13:25:29 - INFO - {
#   "REAL": {
#     "76": [
#       {
#         "timestamp": 1773306080000000000,
#         "y": 997.8019959509448
#       }
#     ]
#   },
#   "PRED": {
#     "ca6e5966-1c7f-11f1-8bb3-72a1019c5619": {
#       "76": [
#         {
#           "timestamp": 1773306080000000000,
#           "y": 1002.8917846679688
#         }
#       ]
#     }
#   },
#   "DIFF": {
#     "ca6e5966-1c7f-11f1-8bb3-72a1019c5619": {
#       "76": [
#         {
#           "timestamp": 1773306080000000000,
#           "y": 0.002571807693983069
#         }
#       ],
#       "ca6e5966-1c7f-11f1-8bb3-72a1019c5619": [
#         {
#           "timestamp": 1773306080000000000,
#           "y": 0.6879152193599999
#         }
#       ]
#     }
#   },
#   "tags_order": [
#     76
#   ],
#   "tag_independent": [
#     "ca6e5966-1c7f-11f1-8bb3-72a1019c5619"
#   ]
# }
```

#- REAL - Измеренные (реальные) значения по запрошенным тегам
#- PRED - Предсказания значения тегов, сделанные конкретными элементами модели (сгруппированные по UUID элементам модели)
#- DIFF - Артефакты элементов модели, так же сгруппированы по UUID элемента модели. Могут внутри содержать как артефакты по тегам, так и собственные
#- RULE - Артефакты элементов типа "Правило"
#- tags_order - порядок тегов, нужно для UI.
#- tag_independent - перечисление всех артефактов, нужно для UI.

#Некоторых ключей может не быть, если при заданных параметрах, не окажется данных.

?????? ?????????? ?????? API

?????? ???? ??? ??????? ??????????:

```
“ start_date = datetime(2024, 1, 1, 0, 0, 0).isoformat()
  end_date = datetime(2024, 12, 31, 23, 59, 59).isoformat()
  incidents = requests.get(
    f'https://mlad/api/incidents/external?start_date={start_date}&end_date={end_date}',
    verify=False,
    headers={
      'Token': auth_response.json()["token"],
    },
  )
```

Revision #2

Created 25 September 2026 10:08:58 by Эльдар Юсуфов

Updated 25 September 2026 10:17:24 by Эльдар Юсуфов