

Анализ проекта http-api-3.1 от 2024-12-01 02:46:07.694943 UTC+3

Дата последнего изменения проекта : 2024-12-01 02:46:07.694943 UTC+3
Общее количество ошибок : 1418 Ошибки логирования : 16; Ошибки стандарта
PER : 1402

Ошибки логирования

print_statements

Расположение файла в дереве проекта

./path/to/http-api-3.1/http-api-3.1/restapi/rest/response.py

Номер -> Строка

```
| 115 -> print(url.query)
```

Расположение файла в дереве проекта

./path/to/http-api-3.1/http-api-3.1/restapi/server.py

Номер -> Строка

```
| 81 -> print("Disconnection completed")
```

Расположение файла в дереве проекта

./path/to/http-api-3.1/http-api-3.1/restapi/server.py

Номер -> Строка

```
| 365 -> print(e)
```

Расположение файла в дереве проекта

./path/to/http-api-3.1/http-api-3.1/restapi/utilities/logs.py

Номер -> Строка

```
| 25 -> print(
```

Расположение файла в дереве проекта

./path/to/http-api-3.1/http-api-3.1/restapi/utilities/logs.py

Номер -> Строка

```
| 34 -> print(
```

logging_imports

Расположение файла в дереве проекта

./path/to/http-api-3.1/http-api-3.1/restapi/server.py

Номер -> Строка

```
| 4 ->
```

basic_config_usage

non_lazy_logging

Расположение файла в дереве проекта

./path/to/http-api-3.1/http-api-3.1/restapi/connectors/neo4j/__init__.py

Номер -> Строка

```
| 245 -> log.error(f"Failed to execute Cypher Query\n{e}")
```

Расположение файла в дереве проекта

./path/to/http-api-3.1/http-api-3.1/restapi/utilities/faker.py

Номер -> Строка

```
| 126 -> log.warning(f"Today I'm {FAKER_LOCALES.get(loc)}")
```

json_logger_usage

global_loggers

scoped_loggers

Ошибки стандарта PEP

./path/to/http-api-3.1//http-api-3.1/docs/conf.py:
Строка 3: Слишком длинный комментарий

```
This file only contains a selection of the most common options. For a full
```

./path/to/http-api-3.1//http-api-3.1/docs/conf.py:
Строка 7: Слишком длинный комментарий

```
-- Path setup -----
```

./path/to/http-api-3.1//http-api-3.1/docs/conf.py:
Строка 9: Слишком длинный комментарий

```
If extensions (or modules to document with autodoc) are in another directory,
```

./path/to/http-api-3.1//http-api-3.1/docs/conf.py:
Строка 10: Слишком длинный комментарий

```
add these directories to sys.path here. If the directory is relative to the
```

./path/to/http-api-3.1//http-api-3.1/docs/conf.py:
Строка 11: Слишком длинный комментарий

```
documentation root, use os.path.abspath to make it absolute, like shown here.
```

./path/to/http-api-3.1//http-api-3.1/docs/conf.py:
Строка 19: Слишком длинный комментарий

```
-- Project information -----
```

./path/to/http-api-3.1//http-api-3.1/docs/conf.py:
Строка 29: Слишком длинный комментарий

```
-- General configuration -----
```

./path/to/http-api-3.1//http-api-3.1/docs/conf.py:
Строка 60: Слишком длинный комментарий

```
-- Options for HTML output -----
```

./path/to/http-api-3.1//http-api-3.1/docs/conf.py:
Строка 62: Слишком длинный комментарий

```
The theme to use for HTML and HTML Help pages. See the documentation for
```

./path/to/http-api-3.1//http-api-3.1/docs/conf.py:
Строка 67: Слишком длинный комментарий

```
Add any paths that contain custom static files (such as style sheets) here,
```

./path/to/http-api-3.1//http-api-3.1/docs/conf.py:
Строка 68: Слишком длинный комментарий

```
relative to this directory. They are copied after the builtin static files,
```

./path/to/http-api-3.1//http-api-3.1/restapi/alembic
_env.py: Строка 89: Превышена максимальная длина
(>79 символов)

```
| 89 -> context.configure(connection=connection, target_metadata=target_metadata)
```

./path/to/http-api-3.1//http-api-3.1/restapi/config.
py: Переменная API_URL: Некорректное именован
ие
API_URL

./path/to/http-api-3.1//http-api-3.1/restapi/config.
py: Переменная AUTH_URL: Некорректное именован
ие
AUTH_URL

./path/to/http-api-3.1//http-api-3.1/restapi/config.
py: Переменная APP_MODE: Некорректное именован
ие
APP_MODE

./path/to/http-api-3.1//http-api-3.1/restapi/config.
py: Переменная FORCE_PRODUCTION_TESTS: Некорректное
именование
FORCE_PRODUCTION_TESTS

./path/to/http-api-3.1//http-api-3.1/restapi/config.
py: Переменная TESTING: Некорректное именован
ие
TESTING

./path/to/http-api-3.1//http-api-3.1/restapi/config.
py: Переменная PRODUCTION: Некорректное именован
PRODUCTION

./path/to/http-api-3.1//http-api-3.1/restapi/config.
py: Переменная STACKTRACE: Некорректное именован
STACKTRACE

./path/to/http-api-3.1//http-api-3.1/restapi/config.
py: Переменная REMOVE_DATA_AT_INIT_TIME:
Некорректное именован
REMOVE_DATA_AT_INIT_TIME

./path/to/http-api-3.1//http-api-3.1/restapi/config.
py: Переменная HOSTNAME: Некорректное именован
HOSTNAME

./path/to/http-api-3.1//http-api-3.1/restapi/config.
py: Переменная MAIN_SERVER_NAME: Некорректное
именование
MAIN_SERVER_NAME

./path/to/http-api-3.1//http-api-3.1/restapi/config.
py: Переменная BACKEND_HOSTNAME: Некорректное
именование
BACKEND_HOSTNAME

./path/to/http-api-3.1//http-api-3.1/restapi/config.
py: Переменная FLOWER_HOSTNAME: Некорректное
именование
FLOWER_HOSTNAME

./path/to/http-api-3.1//http-api-3.1/restapi/config.
py: Переменная CELERYBEAT_HOSTNAME: Некорректное
именование

CELERYBEAT_HOSTNAME

./path/to/http-api-3.1//http-api-3.1/restapi/config.
py: Переменная CELERY_HOSTNAME: Некорректное
именование

CELERY_HOSTNAME

./path/to/http-api-3.1//http-api-3.1/restapi/config.
py: Переменная DOCS: Некорректное имя

DOCS

./path/to/http-api-3.1//http-api-3.1/restapi/config.
py: Переменная HOST_TYPE: Некорректное имя

HOST_TYPE

./path/to/http-api-3.1//http-api-3.1/restapi/config.
py: Переменная DEFAULT_HOST: Некорректное имя

DEFAULT_HOST

./path/to/http-api-3.1//http-api-3.1/restapi/config.
py: Переменная DEFAULT_PORT: Некорректное имя

DEFAULT_PORT

./path/to/http-api-3.1//http-api-3.1/restapi/config.
py: Переменная DATA_PATH: Некорректное имя

DATA_PATH

./path/to/http-api-3.1//http-api-3.1/restapi/config.
py: Переменная IMPORT_PATH: Некорректное имя

IMPORT_PATH

./path/to/http-api-3.1//http-api-3.1/restapi/config.
py: Переменная CODE_DIR: Некорректное имя

CODE_DIR

./path/to/http-api-3.1//http-api-3.1/restapi/config.
py: Переменная APP_SECRETS: Некорректное именование
APP_SECRETS

./path/to/http-api-3.1//http-api-3.1/restapi/config.
py: Переменная JWT_SECRET_FILE: Некорректное
именование
JWT_SECRET_FILE

./path/to/http-api-3.1//http-api-3.1/restapi/config.
py: Переменная TOTP_SECRET_FILE: Некорректное
именование
TOTP_SECRET_FILE

./path/to/http-api-3.1//http-api-3.1/restapi/config.
py: Переменная SSL_CERTIFICATE: Некорректное
именование
SSL_CERTIFICATE

./path/to/http-api-3.1//http-api-3.1/restapi/config.
py: Переменная SSL_SECRET: Некорректное именование
SSL_SECRET

./path/to/http-api-3.1//http-api-3.1/restapi/config.
py: Переменная DOMAIN: Некорректное именование
DOMAIN

./path/to/http-api-3.1//http-api-3.1/restapi/config.
py: Переменная PROXIED_CONNECTION: Некорректное
именование
PROXIED_CONNECTION

./path/to/http-api-3.1//http-api-3.1/restapi/config.
py: Переменная MODELS_DIR: Некорректное именование
MODELS_DIR

./path/to/http-api-3.1//http-api-3.1/restapi/config.
py: Переменная CONF_PATH: Некорректное именован
CONF_PATH

./path/to/http-api-3.1//http-api-3.1/restapi/config.
py: Переменная EXTENDED_PROJECT_DISABLED:
Некорректное именован
EXTENDED_PROJECT_DISABLED

./path/to/http-api-3.1//http-api-3.1/restapi/config.
py: Переменная BACKEND_PACKAGE: Некорректное
именование
BACKEND_PACKAGE

./path/to/http-api-3.1//http-api-3.1/restapi/config.
py: Переменная CUSTOM_PACKAGE: Некорректное
именование
CUSTOM_PACKAGE

./path/to/http-api-3.1//http-api-3.1/restapi/config.
py: Переменная EXTENDED_PACKAGE: Некорректное
именование
EXTENDED_PACKAGE

./path/to/http-api-3.1//http-api-3.1/restapi/config.
py: Переменная SENTRY_URL: Некорректное именован
SENTRY_URL

./path/to/http-api-3.1//http-api-3.1/restapi/config.
py: Переменная ABS_RESTAPI_PATH: Некорректное
именование
ABS_RESTAPI_PATH

./path/to/http-api-3.1//http-api-3.1/restapi/config.
py: Переменная GZIP_ENABLE: Некорректное именован

GZIP_ENABLE

./path/to/http-api-3.1//http-api-3.1/restapi/config.
py: Переменная GZIP_THRESHOLD: Некорректное
именование

GZIP_THRESHOLD

./path/to/http-api-3.1//http-api-3.1/restapi/config.
py: Переменная GZIP_LEVEL: Некорректное имя

GZIP_LEVEL

./path/to/http-api-3.1//http-api-3.1/restapi/config.
py: Переменная BACKEND_URL: Некорректное имя

BACKEND_URL

./path/to/http-api-3.1//http-api-3.1/restapi/config.
py: Переменная BACKEND_PREFIX: Некорректное
именование

BACKEND_PREFIX

./path/to/http-api-3.1//http-api-3.1/restapi/config.
py: Переменная FRONTEND_URL: Некорректное имя

FRONTEND_URL

./path/to/http-api-3.1//http-api-3.1/restapi/config.
py: Переменная FRONTEND_PREFIX: Некорректное
именование

FRONTEND_PREFIX

./path/to/http-api-3.1//http-api-3.1/restapi/config.
py: Переменная BACKEND_PREFIX: Некорректное
именование

BACKEND_PREFIX

./path/to/http-api-3.1//http-api-3.1/restapi/config.py: Переменная FRONTEND_PREFIX: Некорректное именование

FRONTEND_PREFIX

./path/to/http-api-3.1//http-api-3.1/restapi/connectors/celery/beat.py: Строка 4: Превышена максимальная длина (>79 символов)

| 4 -> <http://blog.miguelgrinberg.com/post/celery-and-the-flask-application-factory-pattern>

./path/to/http-api-3.1//http-api-3.1/restapi/connectors/celery/beat.py: Строка 27: Превышена максимальная длина (>79 символов)

| 27 -> # Reload Flask app code for the worker (needed to have the app context available)

./path/to/http-api-3.1//http-api-3.1/restapi/connectors/celery/beat.py: Строка 19: Слишком длинный комментарий

Explicit init_app is needed because the app is created directly from Flask

./path/to/http-api-3.1//http-api-3.1/restapi/connectors/celery/beat.py: Строка 27: Слишком длинный комментарий

Reload Flask app code for the worker (needed to have the app context available)

./path/to/http-api-3.1//http-api-3.1/restapi/connectors/celery/worker.py: Строка 11: Превышена максимальная длина (>79 символов)

| 11 -> # Reload Flask app code for the worker (needed to have the app context available)

./path/to/http-api-3.1//http-api-3.1/restapi/connectors/celery/worker.py: Строка 11: Слишком длинный комментарий

Reload Flask app code for the worker (needed to have the app context available)

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/celery/__init__.py: Строка 16: Превышена
максимальная длина (>79 символов)

```
| 16 -> from restapi.config import CUSTOM_PACKAGE, DOCS, HOST_TYPE,  
SSL_CERTIFICATE, TESTING
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/celery/__init__.py: Строка 20: Превышена
максимальная длина (>79 символов)

```
| 20 -> from restapi.connectors.smtp.notifications import  
send_celery_error_notification
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/celery/__init__.py: Строка 43: Превышена
максимальная длина (>79 символов)

```
| 43 -> def mark_task_as_failed(self: Any, name: str, exception: Exception) ->  
NoReturn:
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/celery/__init__.py: Строка 85: Превышена
максимальная длина (>79 символов)

```
| 85 -> def mark_task_as_failed_ignore(self: Any, name: str, exception:  
Exception) -> NoReturn:
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/celery/__init__.py: Строка 92: Превышена
максимальная длина (>79 символов)

```
| 92 -> log.warning("Celery task {} ({{}}) failed: {{}}", task_id, task_name,  
exception)
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/celery/__init__.py: Строка 139: Превышена
максимальная длина (>79 символов)

```
| 139 -> "Celery task {} ({{}}) failed due to: {{}}, " "but will be retried (fail  
#{{}}/{{}})",
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/celery/___init___.py: Строка 177: Превышена
максимальная длина (>79 символов)

| 177 -> Wrapper of the celery task decorator for a smooth integration in RAPIDo.

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/celery/___init___.py: Строка 180: Превышена
максимальная длина (>79 символов)

| 180 -> If idempotent (i.e. will not cause unintended effects even if called

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/celery/___init___.py: Строка 183: Превышена
максимальная длина (>79 символов)

| 183 -> Note: you can interrupt an idempotent task with
task.revoke(terminate=True)

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/celery/___init___.py: Строка 185: Превышена
максимальная длина (>79 символов)

| 185 -> the default, if not specified, is the name of the decorated function.

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/celery/___init___.py: Строка 186: Превышена
максимальная длина (>79 символов)

| 186 -> :param autoretry_for: A tuple of exception classes. If any of these
exceptions

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/celery/___init___.py: Строка 197: Превышена
максимальная длина (>79 символов)

| 197 -> # This decorator is not covered by tests because can't be tested on
backend

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/celery/___init___.py: Строка 204: Превышена
максимальная длина (>79 символов)

| 204 -> # autoretries will be delayed following via an exponential backoff

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/celery/___init___.py: Строка 208: Превышена
максимальная длина (>79 символов)

```
| 208 -> # to prevent all tasks in the queue from being executed simultaneously.
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/celery/___init___.py: Строка 209: Превышена
максимальная длина (>79 символов)

```
| 209 -> # the delay value will be a random number between zero and retry_backoff
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/celery/___init___.py: Строка 211: Превышена
максимальная длина (>79 символов)

```
| 211 -> # If enabled, messages for this task will be acknowledged after the task
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/celery/___init___.py: Строка 213: Превышена
максимальная длина (>79 символов)

```
| 213 -> # This means the task may be executed multiple times if the worker
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/celery/___init___.py: Строка 221: Превышена
максимальная длина (>79 символов)

```
| 221 -> # app is officially Optional[Flask]... but can never be None at runtime
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/celery/___init___.py: Строка 265: Превышена
максимальная длина (>79 символов)

```
| 265 -> def get_redis_url(variables: dict[str, str], protocol: str, db: int) ->  
str:
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/celery/___init___.py: Строка 290: Превышена
максимальная длина (>79 символов)

```
| 290 -> # ssl_cert_reqs (required): one of the SSLContext.verify_mode values:
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/celery/__init__.py: Строка 298: Превышена
максимальная длина (>79 символов)

```
| 298 -> server_hostname = RabbitExt.get_hostname(service_vars.get("host", ""))
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/celery/__init__.py: Строка 330: Превышена
максимальная длина (>79 символов)

```
| 330 -> print_and_exit("Unable to start Celery: unknown broker service: {}",  
broker)
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/celery/__init__.py: Строка 399: Превышена
максимальная длина (>79 символов)

```
| 399 -> # https://docs.celeryproject.org/en/master/faq.html#does-celery-support-  
task-priorities
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/celery/__init__.py: Строка 416: Превышена
максимальная длина (>79 символов)

```
| 416 -> # of concurrent processes. The default is 4 (four messages for each  
process).
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/celery/__init__.py: Строка 417: Превышена
максимальная длина (>79 символов)

```
| 417 -> # The default setting is usually a good choice, however - if you have  
very
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/celery/__init__.py: Строка 418: Превышена
максимальная длина (>79 символов)

```
| 418 -> # long running tasks waiting in the queue and you have to start the  
workers,
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/celery/__init__.py: Строка 419: Превышена
максимальная длина (>79 символов)

```
| 419 -> # note that the first worker to start will receive four times the number
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/celery/__init__.py: Строка 420: Превышена
максимальная длина (>79 символов)

```
| 420 -> # of messages initially. Thus the tasks may not be fairly distributed to
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/celery/__init__.py: Строка 421: Превышена
максимальная длина (>79 символов)

```
| 421 -> # the workers. To disable prefetching, set worker_prefetch_multiplier to  
1.
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/celery/__init__.py: Строка 422: Превышена
максимальная длина (>79 символов)

```
| 422 -> # Changing that setting to 0 will allow the worker to keep consuming as  
many
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/celery/__init__.py: Строка 426: Превышена
максимальная длина (>79 символов)

```
| 426 -> # Introduced in Celery 5.1: on connection loss cancels all currently  
executed
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/celery/__init__.py: Строка 432: Превышена
максимальная длина (>79 символов)

```
| 432 -> self.celery_app.conf.worker_cancel_long_running_tasks_on_connection_loss  
= True
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/celery/__init__.py: Строка 434: Превышена
максимальная длина (>79 символов)

```
| 434 -> # Automatically try to establish the connection to the AMQP broker on  
Celery startup
```

```
./path/to/http-api-3.1//http-api-3.1/restapi/connect  
ors/celery/__init__.py: Строка 449: Превышена  
максимальная длина (>79 символов)
```

```
| 449 -> log.info("Celery-beat connected to Redis: {}", obfuscate_url(url))
```

```
./path/to/http-api-3.1//http-api-3.1/restapi/connect  
ors/celery/__init__.py: Строка 452: Превышена  
максимальная длина (>79 символов)
```

```
| 452 -> "Cannot configure celery beat scheduler with backend: {}", backend
```

```
./path/to/http-api-3.1//http-api-3.1/restapi/connect  
ors/celery/__init__.py: Строка 542: Превышена  
максимальная длина (>79 символов)
```

```
| 542 -> f"Unsupported celery-beat scheduler: {cls.CELERYBEAT_SCHEDULER}"
```

```
./path/to/http-api-3.1//http-api-3.1/restapi/connect  
ors/celery/__init__.py: Строка 582: Превышена  
максимальная длина (>79 символов)
```

```
| 582 -> f"Unsupported celery-beat scheduler: {cls.CELERYBEAT_SCHEDULER}"
```

```
./path/to/http-api-3.1//http-api-3.1/restapi/connect  
ors/celery/__init__.py: Переменная  
REDBEAT_KEY_PREFIX: Некорректное именование
```

```
REDBEAT_KEY_PREFIX
```

```
./path/to/http-api-3.1//http-api-3.1/restapi/connect  
ors/celery/__init__.py: Переменная F: Некорректное  
именование
```

```
F
```

```
./path/to/http-api-3.1//http-api-3.1/restapi/connect  
ors/celery/__init__.py: Переменная  
CELERYBEAT_SCHEDULER: Некорректное именование
```


CELERYBEAT_SCHEDULER

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/celery/__init__.py: Переменная MAX_RETRIES:
Некорректное именование

MAX_RETRIES

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/celery/__init__.py: Строка 197: Слишком длинный
комментарий

This decorator is not covered by tests because can't be tested on backend

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/celery/__init__.py: Строка 416: Слишком длинный
комментарий

of concurrent processes. The default is 4 (four messages for each process).

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/celery/__init__.py: Строка 418: Слишком длинный
комментарий

long running tasks waiting in the queue and you have to start the workers,

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/celery/__init__.py: Строка 421: Слишком длинный
комментарий

the workers. To disable prefetching, set worker_prefetch_multiplier to 1.

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/celery/__init__.py: Строка 422: Слишком длинный
комментарий

Changing that setting to 0 will allow the worker to keep consuming as many

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/celery/__init__.py: Строка 426: Слишком длинный
комментарий

Introduced in Celery 5.1: on connection loss cancels all currently executed

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/celery/__init__.py: Строка 434: Слишком длинный
комментарий

```
Automatically try to establish the connection to the AMQP broker on Celery  
startup
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/ftp/__init__.py: Строка 51: Превышена
максимальная длина (>79 символов)

```
| 51 -> # Exception raised when a reply is received from the server that does not  
fit the
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/ftp/__init__.py: Класс FTP_TLS_SharedSession:
Некорректное именование

FTP_TLS_SharedSession

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/ftp/__init__.py: Строка 51: Слишком длинный
комментарий

```
Exception raised when a reply is received from the server that does not fit the
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/neo4j/models.py: Строка 47: Превышена
максимальная длина (>79 символов)

```
| 47 -> members = RelationshipFrom(UserModelPath, "BELONGS_TO",  
cardinality=ZeroOrMore)
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/neo4j/models.py: Строка 64: Превышена
максимальная длина (>79 символов)

```
| 64 -> tokens = RelationshipTo(TokenModelPath, "HAS_TOKEN",  
cardinality=ZeroOrMore)
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/neo4j/models.py: Строка 66: Превышена

максимальная длина (>79 символов)

```
| 66 -> belongs_to = RelationshipTo(GroupModelPath, "BELONGS_TO",  
cardinality=ZeroOrOne)
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/neo4j/models.py: Строка 67: Превышена
максимальная длина (>79 символов)

```
| 67 -> logins = RelationshipTo(LoginModelPath, "HAS_LOGIN",  
cardinality=ZeroOrMore)
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/neo4j/models.py: Строка 79: Превышена
максимальная длина (>79 символов)

```
| 79 -> emitted_for = RelationshipFrom(UserModelPath, "HAS_TOKEN",  
cardinality=ZeroOrOne)
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/neo4j/models.py: Строка 85: Превышена
максимальная длина (>79 символов)

```
| 85 -> privileged = RelationshipFrom(UserModelPath, "HAS_ROLE",  
cardinality=OneOrMore)
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/neo4j/models.py: Переменная UserCustomClass:
Некорректное именование

UserCustomClass

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/neo4j/models.py: Переменная GroupCustomClass:
Некорректное именование

GroupCustomClass

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/neo4j/models.py: Переменная GroupModelPath:
Некорректное именование

GroupModelPath

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/neo4j/models.py: Переменная LoginModelPath:
Некорректное именованение

LoginModelPath

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/neo4j/models.py: Переменная RoleModelPath:
Некорректное именованение

RoleModelPath

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/neo4j/models.py: Переменная TokenModelPath:
Некорректное именованение

TokenModelPath

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/neo4j/models.py: Переменная UserModelPath:
Некорректное именованение

UserModelPath

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/neo4j/models.py: Переменная IP: Некорректное
именование

IP

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/neo4j/models.py: Переменная IP: Некорректное
именование

IP

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/neo4j/parser.py: Строка 14: Превышена
максимальная длина (>79 символов)

```
| 14 -> f"Wrong field syntax, type is missing in {f}. Expected {f}:TYPE. "
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/neo4j/parser.py: Строка 82: Превышена
максимальная длина (>79 символов)

```
| 82 -> def switch_label(cls, old_label: str, new_label: str, limit: int = 10000)
-> None:
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/neo4j/parser.py: Строка 131: Превышена
максимальная длина (>79 символов)

```
| 131 -> "Deleted {} (:{}-[]->(:{})) relationships", num, label1, relation,
label2
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/neo4j/parser.py: Строка 169: Превышена
максимальная длина (>79 символов)

```
| 169 -> LOAD CSV WITH HEADERS FROM 'file:///{}' AS line
FIELDTERMINATOR '\t'
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/neo4j/parser.py: Строка 240: Превышена
максимальная длина (>79 символов)

```
| 240 -> LOAD CSV WITH HEADERS FROM 'file:///{}' AS line
FIELDTERMINATOR '\t'
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/neo4j/parser.py: Переменная NODE1_LABEL:
Некорректное именование

NODE1_LABEL

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/neo4j/parser.py: Переменная NODE2_LABEL:
Некорректное именование

NODE2_LABEL

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/neo4j/parser.py: Переменная labelsOrTypes:
Некорректное именование

labelsOrTypes

./path/to/http-api-3.1//http-api-3.1/restapi/connectors/neo4j/types.py: Строка 19: Превышена максимальная длина (>79 символов)

```
| 19 -> # These are not compatible with marshmallow that serializes with
        UUID(value) so that
```

./path/to/http-api-3.1//http-api-3.1/restapi/connectors/neo4j/types.py: Строка 19: Слишком длинный комментарий

```
These are not compatible with marshmallow that serializes with UUID(value) so
that
```

./path/to/http-api-3.1//http-api-3.1/restapi/connectors/neo4j/__init__.py: Строка 2: Превышена максимальная длина (>79 символов)

```
| 2 -> Neo4j connector with automatic integration in rapydo framework, based on
        neomodel
```

./path/to/http-api-3.1//http-api-3.1/restapi/connectors/neo4j/__init__.py: Строка 72: Превышена максимальная длина (>79 символов)

```
| 72 -> rf"Node\([0-9]+\) {t} `(.+)` and property `(.+)` = '(.+)'", str(e)
```

./path/to/http-api-3.1//http-api-3.1/restapi/connectors/neo4j/__init__.py: Строка 82: Превышена максимальная длина (>79 символов)

```
| 82 -> # Can't be tested, should never happen except in case of new neo4j version
```

./path/to/http-api-3.1//http-api-3.1/restapi/connectors/neo4j/__init__.py: Строка 84: Превышена максимальная длина (>79 символов)

```
| 84 -> raise DatabaseDuplicatedEntry("Duplicated entry") from e # pragma: no
        cover
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/neo4j/__init__.py: Строка 93: Превышена
максимальная длина (>79 символов)

```
| 93 -> message = f"Missing property {missing_property} required by {model}"
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/neo4j/__init__.py: Строка 104: Превышена
максимальная длина (>79 символов)

```
| 104 -> # Caught in case of re-raise for example RequiredProperty -> BadRequest
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/neo4j/__init__.py: Строка 133: Превышена
максимальная длина (>79 символов)

```
| 133 -> # https://github.com/neo4j/neo4j-python-driver/blob/d36334e80a66d57b3262  
1d319032751d2204ef67/neo4j/addressing.py#L112
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/neo4j/__init__.py: Строка 148: Превышена
максимальная длина (>79 символов)

```
| 148 -> #  
https://neomodel.readthedocs.io/en/latest/getting_started.html#connecting
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/neo4j/__init__.py: Строка 189: Превышена
максимальная длина (>79 символов)

```
| 189 -> # With Neo4j 4.3 remove all labels on empty DB started to fail with:
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/neo4j/__init__.py: Строка 191: Превышена
максимальная длина (>79 символов)

```
| 191 -> # File "/usr/local/lib/python3.9/dist-packages/neomodel/core.py", ...
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/neo4j/__init__.py: Строка 192: Превышена
максимальная длина (>79 символов)

```
| 192 -> # ... line 62, in drop_indexes
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/neo4j/__init__.py: Строка 202: Превышена
максимальная длина (>79 символов)

```
| 202 -> # install_all_labels can fail when models are cross-referenced between
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/neo4j/__init__.py: Строка 205: Превышена
максимальная длина (>79 символов)

```
| 205 -> # {code: Neo.ClientError.Schema.EquivalentSchemaRuleAlreadyExists}
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/neo4j/__init__.py: Строка 207: Превышена
максимальная длина (>79 символов)

```
| 207 -> # 'Constraint( type='UNIQUENESS', schema=(XYZ {uuid}), [...]
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/neo4j/__init__.py: Строка 231: Превышена
максимальная длина (>79 символов)

```
| 231 -> # Argument 1 to "update_properties" becomes "Any" due to an unfollowed  
import
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/neo4j/__init__.py: Строка 232: Превышена
максимальная длина (>79 символов)

```
| 232 -> def update_properties(instance: StructuredNode, properties: dict[str,  
Any]) -> None: # type: ignore # noqa
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/neo4j/__init__.py: Строка 252: Превышена
максимальная длина (>79 символов)

```
| 252 -> Strip and clean up terms from special characters. To be used in fuzzy  
search
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/neo4j/__init__.py: Строка 254: Превышена
максимальная длина (>79 символов)


```
| 254 -> return term.strip().replace("*", "").replace("'", "\\'").replace("~",
"")
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/neo4j/__init__.py: Строка 386: Превышена
максимальная длина (>79 символов)

```
| 386 -> self.custom_user_properties_post(user, userdata, extra_userdata,
self.db)
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/neo4j/__init__.py: Строка 421: Превышена
максимальная длина (>79 символов)

```
| 421 -> self, user: User, token: str, payload: Payload, token_type:
Optional[str] = None
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/neo4j/__init__.py: Строка 429: Превышена
максимальная длина (>79 символов)

```
| 429 -> exp = payload.get("exp", now +
timedelta(seconds=self.DEFAULT_TOKEN_TTL))
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/neo4j/__init__.py: Строка 530: Превышена
максимальная длина (>79 символов)

```
| 530 -> def save_login(self, username: str, user: Optional[User], failed: bool)
-> None:
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/neo4j/__init__.py: Строка 554: Превышена
максимальная длина (>79 символов)

```
| 554 -> logins = self.db.Login.nodes.filter(username=username, flushed=False)
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/neo4j/__init__.py: Строка 561: Превышена
максимальная длина (>79 символов)

```
| 561 -> for login in self.db.Login.nodes.filter(username=username,
flushed=False):
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/neo4j/__init__.py: Переменная F: Некорректное
именование

F

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/neo4j/__init__.py: Переменная USER: Некорректное
именование

USER

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/neo4j/__init__.py: Переменная PWD: Некорректное
именование

PWD

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/neo4j/__init__.py: Переменная HOST: Некорректное
именование

HOST

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/neo4j/__init__.py: Переменная PORT: Некорректное
именование

PORT

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/neo4j/__init__.py: Переменная DATABASE:
Некорректное имя

DATABASE

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/neo4j/__init__.py: Переменная URI: Некорректное
именование

URI

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/neo4j/__init__.py: Строка 133: Слишком длинный
комментарий

```
https://github.com/neo4j/neo4j-python-driver/blob/d36334e80a66d57b32621d319032751  
d2204ef67/neo4j/addressing.py
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/neo4j/__init__.py: Строка 231: Слишком длинный
комментарий

```
Argument 1 to "update_properties" becomes "Any" due to an unfollowed import
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/rabbitmq/__init__.py: Строка 2: Превышена
максимальная длина (>79 символов)

```
| 2 -> RabbitMQ connector with automatic integration in rapydo framework, based  
on pika
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/rabbitmq/__init__.py: Строка 35: Превышена
максимальная длина (>79 символов)

```
| 35 -> self.channel: Optional[pika.adapters.blocking_connection.BlockingChannel]  
= None
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/rabbitmq/__init__.py: Строка 76: Превышена
максимальная длина (>79 символов)

```
| 76 -> # -> Cannot create a client socket with a PROTOCOL_TLS_SERVER context
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/rabbitmq/__init__.py: Строка 103: Превышена
максимальная длина (>79 символов)

```
| 103 -> context=context, server_hostname=self.get_hostname(host)
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/rabbitmq/__init__.py: Строка 129: Превышена
максимальная длина (>79 символов)

```
| 129 -> # Stream connection lost: ConnectionResetError(104, 'Connection reset by peer')
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/rabbitmq/__init__.py: Строка 138: Превышена
максимальная длина (>79 символов)

```
| 138 -> # this will verify if channel is still open and will try to recreate it
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/rabbitmq/__init__.py: Строка 139: Превышена
максимальная длина (>79 символов)

```
| 139 -> # In some conditions the connection is open but in a wrong state and the
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/rabbitmq/__init__.py: Строка 141: Превышена
максимальная длина (>79 символов)

```
| 141 -> # This way the connection will be considered as closed and invalidated
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/rabbitmq/__init__.py: Строка 153: Превышена
максимальная длина (>79 символов)

```
| 153 -> Host can be the internal rabbit.dockerized.io url or any other external  
url
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/rabbitmq/__init__.py: Строка 154: Превышена
максимальная длина (>79 символов)

```
| 154 -> if the host is an external address it will be assumed to match the cert  
host
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/rabbitmq/__init__.py: Строка 156: Превышена
максимальная длина (>79 символов)

```
| 156 -> inspected to verify in the certificate will probably be a self signed  
cert
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/rabbitmq/__init__.py: Строка 174: Превышена
максимальная длина (>79 символов)

```
| 174 -> exchange=exchange, exchange_type="direct", durable=True,  
auto_delete=False
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/rabbitmq/__init__.py: Строка 218: Превышена
максимальная длина (>79 символов)

```
| 218 -> # url-encode unsafe characters by also including / (thanks to safe  
parameter)
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/rabbitmq/__init__.py: Строка 224: Превышена
максимальная длина (>79 символов)

```
| 224 -> log.error("Invalid rabbitmq username or password, can't retrieve  
bindings")
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/rabbitmq/__init__.py: Строка 230: Превышена
максимальная длина (>79 символов)

```
| 230 ->  
f"{schema}{host}:{port}/api/exchanges/{vhost}/{exchange}/bindings/source",
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/rabbitmq/__init__.py: Строка 269: Превышена
максимальная длина (>79 символов)

```
| 269 -> channel.queue_bind(queue=queue, exchange=exchange,  
routing_key=routing_key)
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/rabbitmq/__init__.py: Строка 271: Превышена
максимальная длина (>79 символов)

```
| 271 -> def queue_unbind(self, queue: str, exchange: str, routing_key: str) ->  
None:
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/rabbitmq/__init__.py: Строка 273: Превышена

максимальная длина (>79 символов)

```
| 273 -> channel.queue_unbind(queue=queue, exchange=exchange,  
routing_key=routing_key)
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/rabbitmq/__init__.py: Строка 300: Превышена
максимальная длина (>79 символов)

```
| 300 -> If this message should be json-encoded please use .send_json()
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/rabbitmq/__init__.py: Строка 129: Слишком
длинный комментарий

```
Stream connection lost: ConnectionResetError(104, 'Connection reset by peer')
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/rabbitmq/__init__.py: Строка 218: Слишком
длинный комментарий

```
url-encode unsafe characters by also including / (thanks to safe parameter)
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/redis/__init__.py: Переменная DB: Некорректное
именование

DB

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/redis/__init__.py: Переменная CELERY_BACKEND_DB:
Некорректное имя

CELERY_BACKEND_DB

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/redis/__init__.py: Переменная CACHE_DB:
Некорректное имя

CACHE_DB

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/redis/__init__.py: Переменная CELERY_BEAT_DB:

Некорректное именованние

CELERY_BEAT_DB

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/redis/__init__.py: Переменная CELERY_BROKER_DB:
Некорректное именованние

CELERY_BROKER_DB

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/smtp/mailmock.py: Строка 33: Превышена
максимальная длина (>79 символов)

```
| 33 -> # -> return False if the exception is not None (because it is not handled  
here)
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/smtp/mailmock.py: Строка 77: Превышена
максимальная длина (>79 символов)

```
| 77 -> log.info("Mail mock sending email from {} to {}", from_address,  
dest_addresses)
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/smtp/mailmock.py: Строка 80: Превышена
максимальная длина (>79 символов)

```
| 80 -> log.info("Mail mock sent email from {} to {}", from_address,  
dest_addresses)
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/smtp/mailmock.py: Строка 91: Превышена
максимальная длина (>79 символов)

```
| 91 -> payload = first_payload.get_payload(decode=True).decode("utf-8") # type:  
ignore
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/smtp/mailmock.py: Строка 96: Превышена
максимальная длина (>79 символов)

```
| 96 -> payload = b.get_payload(decode=True).decode("utf-8") # type: ignore
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/smtp/mailmock.py: Класс SMTP_SSL: Некорректное
именование

SMTP_SSL

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/smtp/mailmock.py: Переменная T: Некорректное
именование

T

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/smtp/mailmock.py: Строка 33: Слишком длинный
комментарий

```
-> return False if the exception is not None (because it is not handled here)
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/smtp/notifications.py: Строка 51: Превышена
максимальная длина (>79 символов)

```
| 51 -> def _get_html_template(template_file: str, replaces: dict[str, Any]) ->  
Optional[str]:
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/smtp/notifications.py: Строка 71: Превышена
максимальная длина (>79 символов)

```
| 71 -> templateLoader = jinja2.FileSystemLoader(searchpath=template_path.parent)
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/smtp/notifications.py: Строка 72: Превышена
максимальная длина (>79 символов)

```
| 72 -> templateEnv = jinja2.Environment(loader=templateLoader, autoescape=True)
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/smtp/notifications.py: Строка 185: Превышена
максимальная длина (>79 символов)

```
| 185 -> data={"events": events, "duration": seconds_to_human(duration), "url":  
url},
```


./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/smtp/notifications.py: Строка 203: Превышена
максимальная длина (>79 символов)

```
| 203 -> def notify_update_credentials_to_user(user: User, unhashed_password:  
str) -> None:
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/smtp/notifications.py: Строка 216: Превышена
максимальная длина (>79 символов)

```
| 216 -> task_id: str, task_name: str, arguments: str, error_stack: Any,  
retry_num: int
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/smtp/notifications.py: Переменная
templateLoader: Некорректное именование
templateLoader

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/smtp/notifications.py: Переменная templateEnv:
Некорректное именование
templateEnv

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/smtp/__init__.py: Строка 39: Превышена
максимальная длина (>79 символов)

```
| 39 -> # This way the send method will be able to use variables overridden in  
connect
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/smtp/__init__.py: Строка 109: Превышена
максимальная длина (>79 символов)

```
| 109 -> args=[body, subject, to_address, from_address, cc, bcc, plain_body],
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/smtp/__init__.py: Строка 167: Превышена
максимальная длина (>79 символов)

```
| 167 -> log.error("Skipping send email: destination address not configured")
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/smtp/__init__.py: Переменная MAX_NUM_RETRIES:
Некорректное именование

MAX_NUM_RETRIES

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/smtp/__init__.py: Строка 39: Слишком длинный
комментарий

This way the send method will be able to use variables overridden in connect

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/sqlalchemy/models.py: Строка 51: Превышена
максимальная длина (>79 символов)

```
| 51 -> email: Mapped[str] = mapped_column(String(100), unique=True,  
nullable=False)
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/sqlalchemy/models.py: Строка 52: Превышена
максимальная длина (>79 символов)

```
| 52 -> name: Mapped[str] = mapped_column(String(255, collation=None),  
nullable=False)
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/sqlalchemy/models.py: Строка 53: Превышена
максимальная длина (>79 символов)

```
| 53 -> surname: Mapped[str] = mapped_column(String(255, collation=None),  
nullable=False)
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/sqlalchemy/models.py: Строка 57: Превышена
максимальная длина (>79 символов)

```
| 57 -> first_login: Mapped[Optional[datetime]] =  
mapped_column(DateTime(timezone=True))
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/sqlalchemy/models.py: Строка 58: Превышена

максимальная длина (>79 символов)

```
| 58 -> last_login: Mapped[Optional[datetime]] =  
mapped_column(DateTime(timezone=True))
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/sqlalchemy/models.py: Строка 64: Превышена
максимальная длина (>79 символов)

```
| 64 -> expiration: Mapped[Optional[datetime]] =  
mapped_column(DateTime(timezone=True))
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/sqlalchemy/models.py: Строка 103: Превышена
максимальная длина (>79 символов)

```
| 103 -> creation: Mapped[datetime] = mapped_column(DateTime(timezone=True),  
nullable=False)
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/sqlalchemy/models.py: Строка 104: Превышена
максимальная длина (>79 символов)

```
| 104 -> expiration: Mapped[Optional[datetime]] =  
mapped_column(DateTime(timezone=True))
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/sqlalchemy/models.py: Строка 105: Превышена
максимальная длина (>79 символов)

```
| 105 -> last_access: Mapped[Optional[datetime]] =  
mapped_column(DateTime(timezone=True))
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/sqlalchemy/models.py: Строка 125: Превышена
максимальная длина (>79 символов)

```
| 125 -> shortname: Mapped[str] = mapped_column(String(64), unique=True,  
nullable=False)
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/sqlalchemy/models.py: Строка 143: Превышена
максимальная длина (>79 символов)

максимальная длина (>79 символов)

```
| 132 -> log.critical("Raised unknown exception {}: {}", e.__class__.__name__, e)
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/sqlalchemy/__init__.py: Строка 192: Превышена
максимальная длина (>79 символов)

```
| 192 -> self._session.commit = catch_db_exceptions(self._session.commit) # type:  
ignore
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/sqlalchemy/__init__.py: Строка 193: Превышена
максимальная длина (>79 символов)

```
| 193 -> self._session.flush = catch_db_exceptions(self._session.flush) # type:  
ignore
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/sqlalchemy/__init__.py: Строка 194: Превышена
максимальная длина (>79 символов)

```
| 194 -> Connection.execute = catch_db_exceptions(Connection.execute) # type:  
ignore
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/sqlalchemy/__init__.py: Строка 196: Превышена
максимальная длина (>79 символов)

```
| 196 -> Connection._execute_context =  
catch_db_exceptions(Connection._execute_context) # type: ignore # noqa
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/sqlalchemy/__init__.py: Строка 280: Превышена
максимальная длина (>79 символов)

```
| 280 -> self.custom_user_properties_post(user, userdata, extra_userdata,  
self.db)
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/sqlalchemy/__init__.py: Строка 458: Превышена
максимальная длина (>79 символов)

```
| 458 -> self, user: User, token: str, payload: Payload, token_type:  
Optional[str] = None
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/sqlalchemy/__init__.py: Строка 466: Превышена
максимальная длина (>79 символов)

```
| 466 -> exp = payload.get("exp", now +  
timedelta(seconds=self.DEFAULT_TOKEN_TTL))
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/sqlalchemy/__init__.py: Строка 594: Превышена
максимальная длина (>79 символов)

```
| 594 -> def save_login(self, username: str, user: Optional[User], failed: bool)  
-> None:
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/sqlalchemy/__init__.py: Переменная F:
Некорректное именован

F

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/sqlalchemy/__init__.py: Переменная
DB_INITIALIZING: Некорректное именован

DB_INITIALIZING

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/__init__.py: Строка 25: Превышена максимальная
длина (>79 символов)

```
| 25 -> from restapi.services.authentication import BaseAuthentication,  
NoAuthentication
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/__init__.py: Строка 32: Превышена максимальная
длина (>79 символов)

```
| 32 -> # https://mypy.readthedocs.io/en/latest/generics.html#generic-methods-and-generic-self
```

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/__init__.py: Строка 65: Превышена максимальная

длина (>79 символов)

```
| 65 -> # This is the folder name corresponding to the connector name (neo4j, celery, )
```

./path/to/http-api-3.1//http-api-3.1/restapi/connectors/__init__.py: Строка 91: Превышена максимальная длина (>79 символов)

```
| 91 -> log.debug("Authentication service: {}", Connector.authentication_service)
```

./path/to/http-api-3.1//http-api-3.1/restapi/connectors/__init__.py: Строка 148: Превышена максимальная длина (>79 символов)

```
| 148 -> print_and_exit("Missing initialize method in {}", self.__class__.__name__)
```

./path/to/http-api-3.1//http-api-3.1/restapi/connectors/__init__.py: Строка 155: Превышена максимальная длина (>79 символов)

```
| 155 -> def load_connectors(cls, path: Path, module: str, services: Services) -> Services:
```

./path/to/http-api-3.1//http-api-3.1/restapi/connectors/__init__.py: Строка 169: Превышена максимальная длина (>79 символов)

```
| 169 -> # This is the only exception... we should rename sqlalchemy as alchemy
```

./path/to/http-api-3.1//http-api-3.1/restapi/connectors/__init__.py: Строка 185: Превышена максимальная длина (>79 символов)

```
| 185 -> # HOST found in variables but empty... never happens during tests
```

./path/to/http-api-3.1//http-api-3.1/restapi/connectors/__init__.py: Строка 211: Превышена максимальная длина (>79 символов)

```
| 211 -> # i.e. to verify instance and get_instance in the connector module
```

./path/to/http-api-3.1//http-api-3.1/restapi/connectors/__init__.py: Строка 232: Превышена максимальная длина (>79 символов)

```
| 232 -> def get_class(connector_module: Optional[ModuleType]) ->
Optional[type[Any]]:
```

./path/to/http-api-3.1//http-api-3.1/restapi/connectors/__init__.py: Строка 258: Превышена максимальная длина (>79 символов)

```
| 258 -> BaseAuthentication, Connector._authentication_module.Authentication()
```

./path/to/http-api-3.1//http-api-3.1/restapi/connectors/__init__.py: Строка 272: Превышена максимальная длина (>79 символов)

```
| 272 -> "Auth service '{_}' is not available", Connector.authentication_service
```

./path/to/http-api-3.1//http-api-3.1/restapi/connectors/__init__.py: Строка 329: Превышена максимальная длина (>79 символов)

```
| 329 -> base_models = Meta.import_models(self.name, BACKEND_PACKAGE,
mandatory=True)
```

./path/to/http-api-3.1//http-api-3.1/restapi/connectors/__init__.py: Строка 420: Превышена максимальная длина (>79 символов)

```
| 420 -> raise ServiceUnavailable(f"Service {self.name} is not available") from e
```

./path/to/http-api-3.1//http-api-3.1/restapi/connectors/__init__.py: Строка 472: Превышена максимальная длина (>79 символов)

```
| 472 -> return self.initialize_connection(expiration, verification, **kwargs)
```

./path/to/http-api-3.1//http-api-3.1/restapi/connectors/__init__.py: Строка 500: Превышена максимальная длина (>79 символов)

```
| 500 -> "{_} is no longer connected, connector invalidated", self.name
```


./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/__init__.py: Переменная T: Некорректное
именование

T

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/__init__.py: Переменная CONNECTORS_FOLDER:
Некорректное имя

CONNECTORS_FOLDER

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/__init__.py: Переменная NO_AUTH: Некорректное
именование

NO_AUTH

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/__init__.py: Переменная DEFAULT_DATETIME:
Некорректное имя

DEFAULT_DATETIME

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/__init__.py: Переменная Services: Некорректное
именование

Services

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/__init__.py: Переменная ExceptionsList:
Некорректное имя

ExceptionsList

./path/to/http-api-3.1//http-api-3.1/restapi/connect
ors/__init__.py: Строка 65: Слишком длинный
комментарий

This is the folder name corresponding to the connector name (neo4j, celery,)

./path/to/http-api-3.1//http-api-3.1/restapi/customizer.py: Строка 87: Превышена максимальная длина (>79 СИМВОЛОВ)

```
| 87 -> def get_custom_output_fields(request: FlaskRequest) -> Props: # pragma: no cover
```

./path/to/http-api-3.1//http-api-3.1/restapi/customizer.py: Переменная FlaskApp: Некорректное именование FlaskApp

./path/to/http-api-3.1//http-api-3.1/restapi/customizer.py: Переменная ADMIN: Некорректное именование ADMIN

./path/to/http-api-3.1//http-api-3.1/restapi/customizer.py: Переменная PROFILE: Некорректное именование PROFILE

./path/to/http-api-3.1//http-api-3.1/restapi/customizer.py: Переменная REGISTRATION: Некорректное именование REGISTRATION

./path/to/http-api-3.1//http-api-3.1/restapi/decorators.py: Строка 33: Превышена максимальная длина (>79 СИМВОЛОВ)

```
| 33 -> from restapi.rest.bearer import HTTPTokenAuth as auth # imported as alias for endpoints
```

./path/to/http-api-3.1//http-api-3.1/restapi/decorators.py: Строка 46: Превышена максимальная длина (>79 СИМВОЛОВ)

```
| 46 -> #  
https://github.com/jmcarp/flask-apispec/blob/master/flask\_apispec/annotations.py
```

./path/to/http-api-3.1//http-api-3.1/restapi/decorators.py: Строка 90: Превышена максимальная длина (>79 СИМВОЛОВ)

```
| 90 -> if not normalized_path.startswith(API_URL) and not
normalized_path.startswith(
```

./path/to/http-api-3.1//http-api-3.1/restapi/decorators.py: Строка 116: Превышена максимальная длина (>79 СИМВОЛОВ)

```
| 116 -> # parameters annotated as Any are always accepted, regardless the
runtime type
```

./path/to/http-api-3.1//http-api-3.1/restapi/decorators.py: Строка 132: Превышена максимальная длина (>79 СИМВОЛОВ)

```
| 132 -> # TypeError: Subscripted generics cannot be used with class and instance
checks
```

./path/to/http-api-3.1//http-api-3.1/restapi/decorators.py: Строка 136: Превышена максимальная длина (>79 СИМВОЛОВ)

```
| 136 -> # This can happens if static type is an instance, instead of a Type or a
typing
```

./path/to/http-api-3.1//http-api-3.1/restapi/decorators.py: Строка 142: Превышена максимальная длина (>79 СИМВОЛОВ)

```
| 142 -> match_types(arg_type, runtime_value) for arg_type in
get_args(static_type)
```

./path/to/http-api-3.1//http-api-3.1/restapi/decorators.py: Строка 150: Превышена максимальная длина (>79 СИМВОЛОВ)

```
| 150 -> # This function takes as input the callback function signature and
extracts from both
```

./path/to/http-api-3.1//http-api-3.1/restapi/decorators.py: Строка 152: Превышена максимальная длина

(>79 СИМВОЛОВ)

```
| 152 -> # If the parameter is not found in view_args and kwargs or it is found  
with a wrong
```

./path/to/http-api-3.1//http-api-3.1/restapi/decorat
ors.py: Строка 153: Превышена максимальная длина
(>79 СИМВОЛОВ)

```
| 153 -> # type, then a None is returned and the preload decorator will raise a  
ServerError
```

./path/to/http-api-3.1//http-api-3.1/restapi/decorat
ors.py: Строка 165: Превышена максимальная длина
(>79 СИМВОЛОВ)

```
| 165 -> "Missing (endpoint: EndpointResource) parameter in {}", callback_name
```

./path/to/http-api-3.1//http-api-3.1/restapi/decorat
ors.py: Строка 180: Превышена максимальная длина
(>79 СИМВОЛОВ)

```
| 180 -> # endpoint will be injected by the preload decorator and it is not  
expected
```

./path/to/http-api-3.1//http-api-3.1/restapi/decorat
ors.py: Строка 197: Превышена максимальная длина
(>79 СИМВОЛОВ)

```
| 197 -> # Parameter is missing but it has a default value, so can be safely  
skipped
```

./path/to/http-api-3.1//http-api-3.1/restapi/decorat
ors.py: Строка 225: Превышена максимальная длина
(>79 СИМВОЛОВ)

```
| 225 -> # The callback is expected to have a first argument of type  
EndpointResource and then
```

./path/to/http-api-3.1//http-api-3.1/restapi/decorat
ors.py: Строка 226: Превышена максимальная длина
(>79 СИМВОЛОВ)

```
| 226 -> # optional additional parameters automatically injected from kwargs and  
url parameters
```

./path/to/http-api-3.1//http-api-3.1/restapi/decorators.py: Строка 260: Превышена максимальная длина (>79 символов)

```
| 260 -> # callback can raise exceptions to stop the execution and e.g. implement
```

./path/to/http-api-3.1//http-api-3.1/restapi/decorators.py: Строка 294: Превышена максимальная длина (>79 символов)

```
| 294 -> # Function name is not expanded by any token that could be provided (are ignored)
```

./path/to/http-api-3.1//http-api-3.1/restapi/decorators.py: Строка 298: Превышена максимальная длина (>79 символов)

```
| 298 -> # If the token is validated, the function name is expanded by a token-dependent key
```

./path/to/http-api-3.1//http-api-3.1/restapi/decorators.py: Строка 314: Превышена максимальная длина (>79 символов)

```
| 314 -> print_and_exit("Cannot load @decorators.cache, is Redis connector available?")
```

./path/to/http-api-3.1//http-api-3.1/restapi/decorators.py: Строка 383: Превышена максимальная длина (>79 символов)

```
| 383 -> validate=validate.OneOf(["asc", "desc"]), required=False, load_default="asc"
```

./path/to/http-api-3.1//http-api-3.1/restapi/decorators.py: Строка 389: Превышена максимальная длина (>79 символов)

```
| 389 -> def verify_parameters(self, data: dict[str, Any], **kwargs: Any) -> dict[str, Any]:
```

./path/to/http-api-3.1//http-api-3.1/restapi/decorators.py: Строка 439: Превышена максимальная длина (>79 символов)

```
| 439 -> # This decorator is automatically added to every endpoints... do not use it explicitly
```

./path/to/http-api-3.1//http-api-3.1/restapi/decorators.py: Строка 440: Превышена максимальная длина (>79 символов)

```
| 440 -> def catch_exceptions(**kwargs: Any) -> Callable[[EndpointFunction], EndpointFunction]:
```

./path/to/http-api-3.1//http-api-3.1/restapi/decorators.py: Строка 499: Превышена максимальная длина (>79 символов)

```
| 499 -> "Caught {} exception with ID {}: {}".format(excname, error_id, message)
```

./path/to/http-api-3.1//http-api-3.1/restapi/decorators.py: Строка 509: Превышена максимальная длина (>79 символов)

```
| 509 -> {excname: f"There was an unexpected error. ErrorID: {error_id}"},
```

./path/to/http-api-3.1//http-api-3.1/restapi/decorators.py: Переменная mimeType: Некорректное именование mimeType

./path/to/http-api-3.1//http-api-3.1/restapi/decorators.py: Переменная lastModified: Некорректное именование lastModified

./path/to/http-api-3.1//http-api-3.1/restapi/decorators.py: Строка 46: Слишком длинный комментарий

https://github.com/jmcarp/flask-apispec/blob/master/flask_apispec/annotations.py

`./path/to/http-api-3.1//http-api-3.1/restapi/decorators.py`: Строка 47: Слишком длинный комментарий

`TODO: the original function is not type-hinted... to be fixed in a future`

`./path/to/http-api-3.1//http-api-3.1/restapi/decorators.py`: Строка 116: Слишком длинный комментарий

`parameters annotated as Any are always accepted, regardless the runtime type`

`./path/to/http-api-3.1//http-api-3.1/restapi/decorators.py`: Строка 132: Слишком длинный комментарий

`TypeError: Subscripted generics cannot be used with class and instance checks`

`./path/to/http-api-3.1//http-api-3.1/restapi/decorators.py`: Строка 136: Слишком длинный комментарий

`This can happens if static type is an instance, instead of a Type or a typing`

`./path/to/http-api-3.1//http-api-3.1/restapi/decorators.py`: Строка 150: Слишком длинный комментарий

`This function takes as input the callback function signature and extracts from both`

`./path/to/http-api-3.1//http-api-3.1/restapi/decorators.py`: Строка 151: Слишком длинный комментарий

`view_args and kwargs the corresponding parameters to be injected at runtime`

`./path/to/http-api-3.1//http-api-3.1/restapi/decorators.py`: Строка 152: Слишком длинный комментарий

`If the parameter is not found in view_args and kwargs or it is found with a wrong`

`./path/to/http-api-3.1//http-api-3.1/restapi/decorators.py`: Строка 153: Слишком длинный комментарий

`type, then a None is returned and the preload decorator will raise a ServerError`

`./path/to/http-api-3.1//http-api-3.1/restapi/decorators.py`: Строка 180: Слишком длинный комментарий

endpoint will be injected by the preload decorator and it is not expected

./path/to/http-api-3.1//http-api-3.1/restapi/decorators.py: Строка 197: Слишком длинный комментарий

Parameter is missing but it has a default value, so can be safely skipped

./path/to/http-api-3.1//http-api-3.1/restapi/decorators.py: Строка 225: Слишком длинный комментарий

The callback is expected to have a first argument of type EndpointResource and then

./path/to/http-api-3.1//http-api-3.1/restapi/decorators.py: Строка 226: Слишком длинный комментарий

optional additional parameters automatically injected from kwargs and url parameters

./path/to/http-api-3.1//http-api-3.1/restapi/decorators.py: Строка 294: Слишком длинный комментарий

Function name is not expanded by any token that could be provided (are ignored)

./path/to/http-api-3.1//http-api-3.1/restapi/decorators.py: Строка 298: Слишком длинный комментарий

If the token is validated, the function name is expanded by a token-dependent key

./path/to/http-api-3.1//http-api-3.1/restapi/decorators.py: Строка 439: Слишком длинный комментарий

This decorator is automatically added to every endpoints... do not use it explicitly

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/admin_groups.py: Строка 87: Превышена максимальная длина (>79 символов)

```
| 87 -> def put(self, group_id: str, group: Group, user: User, **kwargs: Any) -> Response:
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/admin_groups.py: Строка 109: Превышена

максимальная длина (>79 символов)

```
| 109 -> f"Cannot delete this group, it is assigned to {len(members)} user(s)"
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/admin_server_stats.py: Строка 27: Превышена максимальная длина (>79 символов)

```
| 27 -> # This is the average system load calculated over a given period of time
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/admin_server_stats.py: Строка 29: Превышена максимальная длина (>79 символов)

```
| 29 -> # In our case, we will show the load average over a period of 15 minutes.
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/admin_server_stats.py: Строка 64: Превышена максимальная длина (>79 символов)

```
| 64 -> boot_time = datetime.fromtimestamp(int(vmstat_out2[24].strip().split(" ")[0]))
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/admin_server_stats.py: Строка 71: Превышена максимальная длина (>79 символов)

```
| 71 -> # Here we will ping google at an interval of five seconds for five times
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/admin_server_stats.py: Строка 85: Превышена максимальная длина (>79 символов)

```
| 85 -> # in: The number of interrupts per second, including the clock.
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/admin_server_stats.py: Строка 91: Превышена максимальная длина (>79 символов)

```
| 91 -> # us: Time spent running non-kernel code. (user time and nice time)
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/admin_tokens.py: Строка 82: Превышена

максимальная длина (>79 символов)

```
| 82 -> # Returning Any from function declared to return "_SupportsLessThan"
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/admin_tokens.py: Строка 93: Превышена максимальная длина (>79 символов)

```
| 93 -> log.error("Found a token without any user assigned: {}", t["id"])
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/admin_tokens.py: Строка 112: Превышена максимальная длина (>79 символов)

```
| 112 -> raise BadRequest(f"Failed token invalidation: {token}") # pragma: no cover
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/admin_users.py: Строка 21: Превышена максимальная длина (>79 символов)

```
| 21 -> def inject_user(endpoint: EndpointResource, user_id: str, user: User) -> dict[str, Any]:
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/admin_users.py: Строка 27: Превышена максимальная длина (>79 символов)

```
| 27 -> if endpoint.auth.is_admin(target_user) and not endpoint.auth.is_admin(user):
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/admin_users.py: Строка 93: Превышена максимальная длина (>79 символов)

```
| 93 -> if not self.auth.is_admin(user) and Role.ADMIN in roles: # pragma: no cover
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/admin_users.py: Строка 148: Превышена максимальная длина (>79 символов)

```
| 148 -> if not self.auth.is_admin(user) and Role.ADMIN in roles: # pragma: no cover
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/admin_users.py: Строка 175: Превышена максимальная длина (>79 символов)

```
| 175 -> # Can't be reached because group_id is prefiltered by marshmallow
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/admin_users.py: Строка 176: Превышена максимальная длина (>79 символов)

```
| 176 -> raise NotFound("This group cannot be found") # pragma: no cover
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/admin_users.py: Строка 186: Превышена максимальная длина (>79 символов)

```
| 186 -> # In both cases tokens should be invalidated to prevent to have tokens
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/admin_users.py: Строка 189: Превышена максимальная длина (>79 символов)

```
| 189 -> # dt_lower (alias for date_lower_than) is a comparison fn that ignores tz
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/admin_users.py: Строка 190: Превышена максимальная длина (>79 символов)

```
| 190 -> if not prev_expiration or dt_lower(target_user.expiration, prev_expiration):
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/admin_users.py: Строка 192: Превышена максимальная длина (>79 символов)

```
| 192 -> # Invalidate all tokens with expiration after the account expiration
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/login.py: Строка 45: Превышена максимальная длина (>79 символов)

```
| 45 -> token, payload, user = self.auth.make_login(username, password, totp_code)
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/login.py: Строка 53: Превышена максимальная длина (>79 символов)

```
| 53 -> message["errors"].append("You do not provided a valid verification code")
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/login_unlock.py: Строка 22: Превышена максимальная длина (>79 символов)

```
| 22 -> description="The unlock sent by email is validated here to restore  
credentials",
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/login_unlock.py: Строка 35: Превышена максимальная длина (>79 символов)

```
| 35 -> token, raiseErrors=True, token_type=self.auth.UNLOCK_CREDENTIALS
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/login_unlock.py: Строка 57: Превышена максимальная длина (>79 символов)

```
| 57 -> # Cannot be tested, this is an extra test to prevent any unauthorized  
access...
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/login_unlock.py: Строка 60: Превышена максимальная длина (>79 символов)

```
| 60 -> raise BadRequest("Invalid unlock token: this request is no longer valid")
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/login_unlock.py: Строка 63: Превышена максимальная длина (>79 символов)

```
| 63 -> if self.auth.count_failed_login(user.email) <  
self.auth.MAX_LOGIN_ATTEMPTS:
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/login_unlock.py: Строка 65: Превышена максимальная длина (>79 символов)

```
| 65 -> raise BadRequest("Invalid unlock token: this request is no longer valid")
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/login_unlock.py: Строка 70: Превышена максимальная длина (>79 символов)

```
| 70 -> "{} provided a valid unlock token and credentials block is now revoked",
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/login_unlock.py: Строка 57: Слишком длинный комментарий

```
Cannot be tested, this is an extra test to prevent any unauthorized access...
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/profile.py: Строка 4: Превышена максимальная длина (>79 символов)

```
| 4 -> from restapi.endpoints.schemas import NewPassword, profile_output,  
profile_patch_input
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/profile.py: Строка 44: Превышена максимальная длина (>79 символов)

```
| 44 -> data = mem.customizer.manipulate_profile(ref=self, user=user, data=data)
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/profile.py: Строка 68: Превышена максимальная длина (>79 символов)

```
| 68 -> self.auth.change_password(user, password, new_password, password_confirm)
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/profile_activation.py: Строка 24: Превышена максимальная длина (>79 символов)

```
| 24 -> 403: "Account temporarily blocked due to the number of failed logins",
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/profile_activation.py: Строка 58: Превышена максимальная длина (>79 символов)

```
| 58 -> # Cannot be tested, this is an extra test to prevent any unauthorized access...
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/profile_activation.py: Строка 89: Превышена максимальная длина (>79 символов)

```
| 89 -> 403: "Account temporarily blocked due to the number of failed logins",
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/profile_activation.py: Строка 117: Превышена максимальная длина (>79 символов)

```
| 117 -> user, activation_token, payload, token_type=auth.ACTIVATE_ACCOUNT
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/profile_activation.py: Строка 58: Слишком длинный комментарий

```
Cannot be tested, this is an extra test to prevent any unauthorized access...
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/profile_registration.py: Строка 68: Превышена максимальная длина (>79 символов)

```
| 68 -> user = self.auth.create_user(kwargs, [self.auth.default_role], group)
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/profile_registration.py: Строка 89: Превышена максимальная длина (>79 символов)

```
| 89 -> raise ServiceUnavailable("Error sending email, please retry")
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/profile_registration.py: Строка 91: Превышена максимальная длина (>79 символов)

```
| 91 -> user, activation_token, payload, token_type=auth.ACTIVATE_ACCOUNT
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/reset_password.py: Строка 22: Превышена максимальная длина (>79 символов)

```
| 22 -> depends_on = ["MAIN_LOGIN_ENABLE", "ALLOW_PASSWORD_RESET", "AUTH_ENABLE"]
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/reset_password.py: Строка 45: Превышена максимальная длина (>79 символов)

```
| 45 -> f"Sorry, {reset_email} is not recognized as a valid username",
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/reset_password.py: Строка 72: Превышена максимальная длина (>79 символов)

```
| 72 -> msg = "We'll send instructions to the email provided if it's associated "
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/reset_password.py: Строка 118: Превышена максимальная длина (>79 символов)

```
| 118 -> raise BadRequest("Invalid reset token: this request is expired") from e
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/reset_password.py: Строка 134: Превышена максимальная длина (>79 символов)

```
| 134 -> # Can't happen because the token is refused from verify_token function
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/reset_password.py: Строка 136: Превышена максимальная длина (>79 символов)

```
| 136 -> raise BadRequest("Invalid reset token: this request is no longer valid")
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/reset_password.py: Строка 145: Превышена максимальная длина (>79 символов)

```
| 145 -> # If user changed the pwd after the token emission invalidate the token
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/reset_password.py: Строка 146: Превышена максимальная длина (>79 символов)

```
| 146 -> # Can't happen because the change password also invalidated the token
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/reset_password.py: Строка 151: Превышена максимальная длина (>79 символов)

```
| 151 -> # Can't happen because the change password also invalidated the token
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/reset_password.py: Строка 169: Превышена максимальная длина (>79 символов)

```
| 169 -> raise BadRequest("New password does not match with confirmation")
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/reset_password.py: Строка 174: Превышена максимальная длина (>79 символов)

```
| 174 -> # I really don't know why this save is required... since it is already
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/reset_password.py: Строка 175: Превышена максимальная длина (>79 символов)

```
| 175 -> # in change_password ... But if I remove it the new pwd is not saved...
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/schemas.py: Строка 122: Превышена максимальная длина (>79 символов)

```
| 122 -> metadata={"description": "Body of your email. You can use html code here."},
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/schemas.py: Строка 124: Превышена максимальная длина (>79 символов)

```
| 124 -> to = fields.Email(required=True, metadata={"label": "Destination email address"})
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/schemas.py: Строка 181: Превышена максимальная длина (>79 символов)


```
| 181 -> attributes["first_login"] = fields.DateTime(allow_none=True,  
format=ISO8601UTC)
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/schemas.py: Строка 182: Превышена максимальная длина (>79 символов)

```
| 182 -> attributes["last_login"] = fields.DateTime(allow_none=True,  
format=ISO8601UTC)
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/schemas.py: Строка 189: Превышена максимальная длина (>79 символов)

```
| 189 -> attributes["expiration"] = fields.DateTime(allow_none=True,  
format=ISO8601UTC)
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/schemas.py: Строка 220: Превышена максимальная длина (>79 символов)

```
| 220 -> # Can't use request.method because it is not passed at loading time, i.e.  
the Specs will
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/schemas.py: Строка 369: Превышена максимальная длина (>79 символов)

```
| 369 -> attributes["expiration"] = fields.DateTime(allow_none=True,  
format=ISO8601UTC)
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/schemas.py: Строка 374: Превышена максимальная длина (>79 символов)

```
| 374 -> attributes["first_login"] = fields.DateTime(required=True,  
format=ISO8601UTC)
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/schemas.py: Строка 375: Превышена максимальная длина (>79 символов)

```
| 375 -> attributes["last_login"] = fields.DateTime(required=True,  
format=ISO8601UTC)
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/schemas.py: Переменная MarshmallowSchema: Некорректное именован

MarshmallowSchema

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/schemas.py: Переменная IP: Некорректное именован

IP

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/schemas.py: Переменная IP: Некорректное именован

IP

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/schemas.py: Строка 220: Слишком длинный комментарий

Can't use request.method because it is not passed at loading time, i.e. the Specs will

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/send_mail.py: Строка 5: Превышена максимальная длина (>79 символов)

| 5 -> from restapi.connectors.smtp.notifications import _get_html_template, convert_html2text

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/swagger_specs.py: Строка 12: Превышена максимальная длина (>79 символов)

| 12 -> def add_model(definitions: dict[str, bool], def_name: str, is_private: bool) -> None:

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/swagger_specs.py: Строка 86: Превышена максимальная длина (>79 символов)

| 86 -> for _, response in definition.get("responses", {}).items():

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/swagger_specs.py: Строка 121: Превышена максимальная длина (>79 символов)

```
| 121 -> # If a definition is referenced by other definitions, its visibility
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/swagger_specs.py: Строка 190: Превышена максимальная длина (>79 символов)

```
| 190 -> "Invalid {} definition, unable to determine the visibility {} {}",
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/swagger_specs.py: Строка 212: Превышена максимальная длина (>79 символов)

```
| 212 -> # If used in at least one public definition, let's consider it as public
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_authentication.py: Строка 15: Превышена максимальная длина (>79 символов)

```
| 15 -> summary="Only resp a fixed response, no authenticataion is required",
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_authentication.py: Строка 101: Превышена максимальная длина (>79 символов)

```
| 101 -> def verify_uuid_value(endpoint: EndpointResource, uuid: str) ->
dict[str, Any]:
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_authentication.py: Строка 106: Превышена максимальная длина (>79 символов)

```
| 106 -> # Returned values if any will be injected into the endpoint as fn
parameters
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_authentication.py: Строка 111: Превышена максимальная длина (>79 символов)

```
| 111 -> # Note: this endpoint has a preload callback to verify the uuid and inject the user
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_authentication.py: Строка 115: Превышена максимальная длина (>79 символов)

```
| 115 -> @decorators.use_kwargs({"test": fields.Bool(required=True)}, location="query")
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_authentication.py: Строка 123: Превышена максимальная длина (>79 символов)

```
| 123 -> def get(self, uuid: str, test: bool, user: User, target_user: User) -> Response:
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_authentication.py: Строка 106: Слишком длинный комментарий

```
Returned values if any will be injected into the endpoint as fn parameters
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_authentication.py: Строка 111: Слишком длинный комментарий

```
Note: this endpoint has a preload callback to verify the uuid and inject the user
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_autocomplete.py: Строка 16: Превышена максимальная длина (>79 символов)

```
| 16 -> # The frontend is expected to translate this field into an autocomplete field
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_autocomplete.py: Строка 28: Превышена максимальная длина (>79 символов)

```
| 28 -> # This will accept a list of elements provided by the autocomplete endpoint.
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_autocomplete.py: Строка 29: Превышена максимальная длина (>79 символов)

```
| 29 -> # The frontend is expected to translate this field into a multiple autocomplete
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_autocomplete.py: Строка 44: Превышена максимальная длина (>79 символов)

```
| 44 -> def get_element(name: str, surname: str, nickname: str) -> tuple[str, str]:
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_autocomplete.py: Строка 16: Слишком длинный комментарий

```
The frontend is expected to translate this field into an autocomplete field
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_autocomplete.py: Строка 28: Слишком длинный комментарий

```
This will accept a list of elements provided by the autocomplete endpoint.
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_autocomplete.py: Строка 29: Слишком длинный комментарий

```
The frontend is expected to translate this field into a multiple autocomplete
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_cache.py: Строка 17: Превышена максимальная длина (>79 символов)

```
| 17 -> counter = 0 # Increased at each request... except cached responses of course
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_cache.py: Строка 38: Превышена максимальная длина (>79 символов)

```
| 38 -> counter = 0 # Increased at each request... except cached responses of course
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_cache.py: Строка 73: Превышена максимальная длина (>79 символов)

```
| 73 -> counter = 0 # Increased at each request... except cached responses of course
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_cache.py: Строка 78: Превышена максимальная длина (>79 символов)

```
| 78 -> summary="Execute tests of cached responses from authenticated endpoints",
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_cache.py: Строка 97: Превышена максимальная длина (>79 символов)

```
| 97 -> counter = 0 # Increased at each request... except cached responses of course
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_cache.py: Строка 102: Превышена максимальная длина (>79 символов)

```
| 102 -> summary="Execute tests of cached responses from optionally auth. endpoints",
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_cache.py: Строка 116: Превышена максимальная длина (>79 символов)

```
| 116 -> Used to test cache of authenticated endpoints that accept access token param
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_cache.py: Строка 122: Превышена максимальная длина (>79 символов)

```
| 122 -> counter = 0 # Increased at each request... except cached responses of course
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_cache.py: Строка 127: Превышена максимальная длина (>79 символов)

```
| 127 -> summary="Execute tests of cached responses auth. endpoints with  
parameter",
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_cache.py: Строка 147: Превышена максимальная длина (>79 символов)

```
| 147 -> counter = 0 # Increased at each request... except cached responses of  
course
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_cache.py: Строка 152: Превышена максимальная длина (>79 символов)

```
| 152 -> summary="Execute tests of caches of optionally auth. endpoints with  
param",
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_databases.py: Строка 36: Превышена максимальная длина (>79 символов)

```
| 36 -> # create group above is intended to fail due to the missing property
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_databases.py: Строка 43: Превышена максимальная длина (>79 символов)

```
| 43 -> # As a side effect this endpoint will modify the fullname of the default
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_databases.py: Строка 44: Превышена максимальная длина (>79 символов)

```
| 44 -> # Group if the exception is not raised. Otherwise this modification will
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_databases.py: Строка 53: Превышена максимальная длина (>79 символов)

```
| 53 -> # Don't commit with alchemy or the transaction can't be rollbacked
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_databases.py: Строка 57: Превышена максимальная длина (>79 символов)

```
| 57 -> # This can fail if data is a duplicate of a already created group
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_databases.py: Строка 58: Превышена максимальная длина (>79 символов)

```
| 58 -> # In this case a DatabaseDuplicatedEntry exception will be raised and the
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_databases.py: Строка 59: Превышена максимальная длина (>79 символов)

```
| 59 -> # database_transaction decorator will undo the change on the default grp
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_databases.py: Строка 60: Превышена максимальная длина (>79 символов)

```
| 60 -> group = self.auth.create_group({"shortname": data, "fullname": data})
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_databases.py: Строка 61: Превышена максимальная длина (>79 символов)

```
| 61 -> # Don't commit with alchemy or the transaction can't be rollbacked
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_download.py: Строка 26: Превышена максимальная длина (>79 символов)

```
| 26 -> def get(self, folder: str, fname: str, stream: bool = False) -> Response:
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_download.py: Строка 31: Превышена максимальная длина (>79 символов)

```
| 31 -> return Downloader.send_file_streamed(fname, subfolder=subfolder)
```


./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_inputs.py: Строка 25: Превышена максимальная длина (>79 символов)

```
| 25 -> # lowercase key without label defined. label will be key.title() in schema
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_inputs.py: Строка 27: Превышена максимальная длина (>79 символов)

```
| 27 -> # non-lowercase key without label defined. label will be == to key in schema
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_inputs.py: Строка 34: Превышена максимальная длина (>79 символов)

```
| 34 -> max=datetime.now(pytz.utc).replace(hour=23, minute=59, second=59),
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_inputs.py: Строка 74: Превышена максимальная длина (>79 символов)

```
| 74 -> myequalstr = fields.Str(required=True, validate=validate.Length(equal=6))
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_inputs.py: Строка 79: Превышена максимальная длина (>79 символов)

```
| 79 -> # fields.Nested is a replacement of the default Nested field with the ability
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_inputs.py: Строка 83: Превышена максимальная длина (>79 символов)

```
| 83 -> mynullablenested = fields.Nested(Nested, required=True, allow_none=True)
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_inputs.py: Строка 85: Превышена максимальная длина (>79 символов)

```
| 85 -> # fields.List is a replacement of the default List field with the ability
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_inputs.py: Строка 93: Превышена максимальная длина (>79 символов)

```
| 93 -> # The type is key[] ... should be something more explicative like  
FieldName[]
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_inputs.py: Переменная CHOICES: Некорректное именован

CHOICES

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_inputs.py: Переменная MYDATE: Некорректное именован

MYDATE

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_inputs.py: Переменная MYDATETIME: Некорректное именован

MYDATETIME

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_inputs.py: Строка 27: Слишком длинный комментарий

```
non-lowercase key without label defined. label will be == to key in schema
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_inputs.py: Строка 79: Слишком длинный комментарий

```
fields.Nested is a replacement of the default Nested field with the ability
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_inputs.py: Строка 93: Слишком длинный комментарий

```
The type is key[] ... should be something more explicative like FieldName[]
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_neo4j.py: Строка 41: Превышена максимальная длина (>79 символов)

```
| 41 -> message="datetime.datetime.utcnow() is deprecated and scheduled for  
removal in a future version", # noqa
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_neo4j.py: Переменная CHOICES_tuple: Некорректное именование

CHOICES_tuple

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_neo4j.py: Переменная CHOICES_dict: Некорректное именование

CHOICES_dict

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_upload.py: Строка 25: Превышена максимальная длина (>79 символов)

```
| 25 -> # This is just to test the allowed exts without adding a new parameter..
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/test_upload.py: Строка 53: Превышена максимальная длина (>79 символов)

```
| 53 -> # This is just to test the allowed exts without adding a new parameter..
```

./path/to/http-api-3.1//http-api-3.1/restapi/endpoints/tokens.py: Строка 57: Превышена максимальная длина (>79 символов)

```
| 57 -> raise BadRequest(f"Failed token invalidation: {token}") # pragma: no  
cover
```

./path/to/http-api-3.1//http-api-3.1/restapi/exceptions.py: Переменная ExceptionType: Некорректное именование

ExceptionType

./path/to/http-api-3.1//http-api-3.1/restapi/mocks.p
у: Строка 2: Превышена максимальная длина (>79
СИМВОЛОВ)

```
| 2 -> Mock classes used to replace customization classes when building the  
documentation
```

./path/to/http-api-3.1//http-api-3.1/restapi/mocks.p
у: Строка 13: Превышена максимальная длина (>79
СИМВОЛОВ)

```
| 13 -> Implement the constructor to add operations performed one-time at  
initialization
```

./path/to/http-api-3.1//http-api-3.1/restapi/mocks.p
у: Строка 19: Превышена максимальная длина (>79
СИМВОЛОВ)

```
| 19 -> # This method is called after normal initialization if TESTING mode is  
enabled
```

./path/to/http-api-3.1//http-api-3.1/restapi/mocks.p
у: Строка 52: Превышена максимальная длина (>79
СИМВОЛОВ)

```
| 52 -> def manipulate_profile(ref: EndpointResource, user: User, data: Props) ->  
Props:
```

./path/to/http-api-3.1//http-api-3.1/restapi/mocks.p
у: Строка 89: Превышена максимальная длина (>79
СИМВОЛОВ)

```
| 89 -> this method is used to extend the output model of profile and admin users
```

./path/to/http-api-3.1//http-api-3.1/restapi/mocks.p
у: Строка 19: Слишком длинный комментарий

```
This method is called after normal initialization if TESTING mode is enabled
```

./path/to/http-api-3.1//http-api-3.1/restapi/models/
fields.py: Строка 13: Превышена максимальная длина
(>79 СИМВОЛОВ)

```
| 13 -> #####  
#####
```

./path/to/http-api-3.1//http-api-3.1/restapi/models/
fields.py: Строка 14: Превышена максимальная длина
(>79 символов)

```
| 14 -> # 1) Override some types with custom implementation to extend  
functionalities
```

./path/to/http-api-3.1//http-api-3.1/restapi/models/
fields.py: Строка 15: Превышена максимальная длина
(>79 символов)

```
| 15 -> #####  
#####
```

./path/to/http-api-3.1//http-api-3.1/restapi/models/
fields.py: Строка 54: Превышена максимальная длина
(>79 символов)

```
| 54 -> # this is the case when requests (or pytest) send some json-dumped lists
```

./path/to/http-api-3.1//http-api-3.1/restapi/models/
fields.py: Строка 72: Превышена максимальная длина
(>79 символов)

```
| 72 -> f"Expected at least {self.min_items} items, received {len(value)}"
```

./path/to/http-api-3.1//http-api-3.1/restapi/models/
fields.py: Строка 82: Превышена максимальная длина
(>79 символов)

```
| 82 -> # the above type is from marshmallow, but it fails with dict[str, Any]  
(imc)
```

./path/to/http-api-3.1//http-api-3.1/restapi/models/
fields.py: Строка 110: Превышена максимальная длина
(>79 символов)

```
| 110 -> # this is the case when requests (or pytest) send some json-dumped object
```

./path/to/http-api-3.1//http-api-3.1/restapi/models/
fields.py: Строка 117: Превышена максимальная длина
(>79 символов)

```
| 117 -> return super()._deserialize(value, attr, data, **kwargs) # type: ignore
```

./path/to/http-api-3.1//http-api-3.1/restapi/models/
fields.py: Строка 132: Превышена максимальная длина
(>79 символов)

```
| 132 -> # Note: Can't use self.unique otherwise the elements will be silently  
cleaned
```

./path/to/http-api-3.1//http-api-3.1/restapi/models/
fields.py: Строка 148: Превышена максимальная длина
(>79 символов)

```
| 148 -> values = super()._deserialize(value, attr, data, **kwargs) # type:  
ignore
```

./path/to/http-api-3.1//http-api-3.1/restapi/models/
fields.py: Строка 156: Превышена максимальная длина
(>79 символов)

```
| 156 -> #####  
#####
```

./path/to/http-api-3.1//http-api-3.1/restapi/models/
fields.py: Строка 158: Превышена максимальная длина
(>79 символов)

```
| 158 -> #####  
#####
```

./path/to/http-api-3.1//http-api-3.1/restapi/models/
fields.py: Строка 161: Превышена максимальная длина
(>79 символов)

```
| 161 -> # Should be extended to automatically include the choices as validation  
field to be
```

./path/to/http-api-3.1//http-api-3.1/restapi/models/
fields.py: Строка 205: Превышена максимальная длина

(>79 СИМВОЛОВ)

```
| 205 -> return super()._serialize(value.all(), attr, obj, **kwargs) # type: ignore
```

./path/to/http-api-3.1//http-api-3.1/restapi/models/fields.py: Строка 222: Превышена максимальная длина (>79 СИМВОЛОВ)

```
| 222 -> return super()._serialize(value.single(), attr, obj, **kwargs) # type: ignore
```

./path/to/http-api-3.1//http-api-3.1/restapi/models/fields.py: Строка 251: Превышена максимальная длина (>79 СИМВОЛОВ)

```
| 251 -> #####  
#####
```

./path/to/http-api-3.1//http-api-3.1/restapi/models/fields.py: Строка 253: Превышена максимальная длина (>79 СИМВОЛОВ)

```
| 253 -> #####  
#####
```

./path/to/http-api-3.1//http-api-3.1/restapi/models/fields.py: Строка 270: Превышена максимальная длина (>79 СИМВОЛОВ)

```
| 270 -> # https://github.com/marshmallow-code/marshmallow/blob/dev/src/marshmallow/fields.py
```

./path/to/http-api-3.1//http-api-3.1/restapi/models/fields.py: Переменная Field: Некорректное именование Field

./path/to/http-api-3.1//http-api-3.1/restapi/models/fields.py: Переменная Boolean: Некорректное именование Boolean

Boolean

./path/to/http-api-3.1//http-api-3.1/restapi/models/
fields.py: Переменная DateTime: Некорректное
именование

DateTime

./path/to/http-api-3.1//http-api-3.1/restapi/models/
fields.py: Переменная Dict: Некорректное имя

Dict

./path/to/http-api-3.1//http-api-3.1/restapi/models/
fields.py: Переменная Number: Некорректное
именование

Number

./path/to/http-api-3.1//http-api-3.1/restapi/models/
fields.py: Переменная AwareDateTime: Некорректное
именование

AwareDateTime

./path/to/http-api-3.1//http-api-3.1/restapi/models/
fields.py: Переменная Date: Некорректное имя

Date

./path/to/http-api-3.1//http-api-3.1/restapi/models/
fields.py: Переменная Decimal: Некорректное
именование

Decimal

./path/to/http-api-3.1//http-api-3.1/restapi/models/
fields.py: Переменная Email: Некорректное имя

Email

./path/to/http-api-3.1//http-api-3.1/restapi/models/
fields.py: Переменная Integer: Некорректное
именование

Integer

./path/to/http-api-3.1//http-api-3.1/restapi/models/
fields.py: Переменная Float: Некорректное именование

Float

./path/to/http-api-3.1//http-api-3.1/restapi/models/
fields.py: Переменная NaiveDateTime: Некорректное
ИМЕНОВАНИЕ

NaiveDateTime

./path/to/http-api-3.1//http-api-3.1/restapi/models/
fields.py: Переменная Url: Некорректное именование

Url

./path/to/http-api-3.1//http-api-3.1/restapi/models/
fields.py: Переменная UUID: Некорректное именование

UUID

./path/to/http-api-3.1//http-api-3.1/restapi/models/
fields.py: Переменная URL: Некорректное именование

URL

./path/to/http-api-3.1//http-api-3.1/restapi/models/
fields.py: Переменная Str: Некорректное именование

Str

./path/to/http-api-3.1//http-api-3.1/restapi/models/
fields.py: Переменная Bool: Некорректное именование

Bool

./path/to/http-api-3.1//http-api-3.1/restapi/models/
fields.py: Переменная Int: Некорректное именование

Int

./path/to/http-api-3.1//http-api-3.1/restapi/models/
fields.py: Строка 14: Слишком длинный комментарий

```
1) Override some types with custom implementation to extend functionalities
```

./path/to/http-api-3.1//http-api-3.1/restapi/models/
fields.py: Строка 18: Слишком длинный комментарий

```
REVIEW: Следовало делать свои расширенные типы, а не патчить существующие
```

./path/to/http-api-3.1//http-api-3.1/restapi/models/
fields.py: Строка 82: Слишком длинный комментарий

```
the above type is from marshmallow, but it fails with dict[str, Any] (imc)
```

./path/to/http-api-3.1//http-api-3.1/restapi/models/
fields.py: Строка 121: Слишком длинный комментарий

```
https://github.com/marshmallow-code/webargs/blob/dev/src/webargs/fields.py
```

./path/to/http-api-3.1//http-api-3.1/restapi/models/
fields.py: Строка 132: Слишком длинный комментарий

```
Note: Can't use self.unique otherwise the elements will be silently cleaned
```

./path/to/http-api-3.1//http-api-3.1/restapi/models/
fields.py: Строка 161: Слишком длинный комментарий

```
Should be extended to automatically include the choices as validation field to be
```

./path/to/http-api-3.1//http-api-3.1/restapi/models/
fields.py: Строка 270: Слишком длинный комментарий

```
https://github.com/marshmallow-code/marshmallow/blob/dev/src/marshmallow/fields.p  
y
```

./path/to/http-api-3.1//http-api-3.1/restapi/models/
schema.py: Строка 16: Превышена максимальная длина
(>79 символов)

```
| 16 -> def __init__(self, strip_required: bool = False, *args: Any, **kwargs:  
Any) -> None:
```

./path/to/http-api-3.1//http-api-3.1/restapi/models/
schema.py: Строка 68: Превышена максимальная длина
(>79 символов)

```
| 68 -> self, model: type[Any], fields: Optional[Any], *args: Any, **kwargs: Any
```

./path/to/http-api-3.1//http-api-3.1/restapi/models/
schema.py: Строка 98: Превышена максимальная длина
(>79 символов)

```
| 98 -> # Skip all parentes up to StructuredNode and StructuredRel (included)
```

./path/to/http-api-3.1//http-api-3.1/restapi/models/
schema.py: Строка 100: Превышена максимальная длина
(>79 символов)

```
| 100 -> # Found the starting point, next class will be descended up to model
```

./path/to/http-api-3.1//http-api-3.1/restapi/models/
schema.py: Строка 103: Превышена максимальная длина
(>79 символов)

```
| 103 -> # skip all parent up to StructuredNode and StructuredRel INCLUDED
```

./path/to/http-api-3.1//http-api-3.1/restapi/models/
schema.py: Строка 113: Превышена максимальная длина
(>79 символов)

```
| 113 -> # self.fields can be None when the special value * is given in input
```

./path/to/http-api-3.1//http-api-3.1/restapi/models/
schema.py: Строка 117: Превышена максимальная длина
(>79 символов)

```
| 117 -> # log.info("Including property {}.{}", model.__name__, attribute)
```

./path/to/http-api-3.1//http-api-3.1/restapi/models/
schema.py: Строка 142: Превышена максимальная длина
(>79 символов)

```
| 142 -> "Unsupport neomodel property: {}, fallback to StringProperty",
```

./path/to/http-api-3.1//http-api-3.1/restapi/models/schema.py: Переменная GET_SCHEMA_KEY: Некорректное именован

GET_SCHEMA_KEY

./path/to/http-api-3.1//http-api-3.1/restapi/models/__init__.py: Строка 11: Превышена максимальная длина (>79 символов)

```
| 11 -> from restapi.models.schema import GET_SCHEMA_KEY, Neo4jSchema,
PartialSchema, Schema
```

./path/to/http-api-3.1//http-api-3.1/restapi/models/__init__.py: Переменная ISO8601UTC: Некорректное именован

ISO8601UTC

./path/to/http-api-3.1//http-api-3.1/restapi/rest/annotations.py: Строка 6: Превышена максимальная длина (>79 символов)

```
| 6 -> def inject_apispec_docs(fn: Any, conf: Any, labels: Optional[list[str]])
-> None:
```

./path/to/http-api-3.1//http-api-3.1/restapi/rest/bearer.py: Строка 67: Превышена максимальная длина (>79 символов)

```
| 67 -> def is_session_user_admin(request: FlaskRequest, auth:
BaseAuthentication) -> bool:
```

./path/to/http-api-3.1//http-api-3.1/restapi/rest/bearer.py: Строка 100: Превышена максимальная длина (>79 символов)

```
| 100 -> # get_self_reference_from_args can return None, this case is quite
```

./path/to/http-api-3.1//http-api-3.1/restapi/rest/bearer.py: Строка 101: Превышена максимальная длина (>79 символов)

```
| 101 -> # impossible... however with None the server can't continue!
```

./path/to/http-api-3.1//http-api-3.1/restapi/rest/be
arer.py: Строка 103: Превышена максимальная длина
(>79 символов)

```
| 103 -> "Server misconfiguration, self reference can't be None!"
```

./path/to/http-api-3.1//http-api-3.1/restapi/rest/be
arer.py: Строка 114: Превышена максимальная длина
(>79 символов)

```
| 114 -> # Check authentication. Optional authentication is valid if:
```

./path/to/http-api-3.1//http-api-3.1/restapi/rest/be
arer.py: Строка 193: Превышена максимальная длина
(>79 символов)

```
| 193 -> # get_self_reference_from_args can return None, this case is quite
```

./path/to/http-api-3.1//http-api-3.1/restapi/rest/be
arer.py: Строка 194: Превышена максимальная длина
(>79 символов)

```
| 194 -> # impossible... however with None the server can't continue!
```

./path/to/http-api-3.1//http-api-3.1/restapi/rest/be
arer.py: Строка 196: Превышена максимальная длина
(>79 символов)

```
| 196 -> "Server misconfiguration, self reference can't be None!"
```

./path/to/http-api-3.1//http-api-3.1/restapi/rest/be
arer.py: Строка 203: Превышена максимальная длина
(>79 символов)

```
| 203 -> f", e.g. {HTTPAUTH_AUTH_FIELD}: '{HTTPAUTH_SCHEME} TOKEN'"
```

./path/to/http-api-3.1//http-api-3.1/restapi/rest/be
arer.py: Строка 207: Превышена максимальная длина
(>79 символов)

```
| 207 -> msg, code=401, headers=HTTPAUTH_ERR_HEADER, allow_html=True
```

./path/to/http-api-3.1//http-api-3.1/restapi/rest/bearer.py: Строка 211: Превышена максимальная длина (>79 символов)

```
| 211 -> # ignore headers and let go, avoid unwanted interactions with CORS
```

./path/to/http-api-3.1//http-api-3.1/restapi/rest/bearer.py: Переменная HTTPAUTH_SCHEME: Некорректное именован

HTTPAUTH_SCHEME

./path/to/http-api-3.1//http-api-3.1/restapi/rest/bearer.py: Переменная HTTPAUTH_AUTH_FIELD: Некорректное именован

HTTPAUTH_AUTH_FIELD

./path/to/http-api-3.1//http-api-3.1/restapi/rest/bearer.py: Переменная HTTPAUTH_ERR_HEADER: Некорректное именован

HTTPAUTH_ERR_HEADER

./path/to/http-api-3.1//http-api-3.1/restapi/rest/bearer.py: Переменная ALLOW_ACCESS_TOKEN_PARAMETER: Некорректное именован

ALLOW_ACCESS_TOKEN_PARAMETER

./path/to/http-api-3.1//http-api-3.1/restapi/rest/bearer.py: Переменная TOKEN_VALIDATED_KEY: Некорректное именован

TOKEN_VALIDATED_KEY

./path/to/http-api-3.1//http-api-3.1/restapi/rest/definition.py: Строка 51: Превышена максимальная длина (>79 символов)

```
| 51 -> "Deprecated use of self.get_user(), user is now injected in the endpoint"
```

./path/to/http-api-3.1//http-api-3.1/restapi/rest/definition.py: Строка 81: Превышена максимальная длина (>79 символов)

```
| 81 -> # If you accept the risk or you do not use marshalling unlock html responses
```

./path/to/http-api-3.1//http-api-3.1/restapi/rest/definition.py: Строка 85: Превышена максимальная длина (>79 символов)

```
| 85 -> content, headers = ResponseMaker.get_html(content, code, headers)
```

./path/to/http-api-3.1//http-api-3.1/restapi/rest/definition.py: Строка 101: Превышена максимальная длина (>79 символов)

```
| 101 -> # This function has to be coupled with a marshal_with(TotalSchema, code=206)
```

./path/to/http-api-3.1//http-api-3.1/restapi/rest/definition.py: Переменная CURRENTPAGE_KEY:
Некорректное именован

CURRENTPAGE_KEY

./path/to/http-api-3.1//http-api-3.1/restapi/rest/definition.py: Переменная DEFAULT_CURRENTPAGE:
Некорректное именован

DEFAULT_CURRENTPAGE

./path/to/http-api-3.1//http-api-3.1/restapi/rest/definition.py: Переменная PERPAGE_KEY: Некорректное именован

PERPAGE_KEY

./path/to/http-api-3.1//http-api-3.1/restapi/rest/definition.py: Переменная DEFAULT_PERPAGE:
Некорректное именован

DEFAULT_PERPAGE

./path/to/http-api-3.1//http-api-3.1/restapi/rest/definition.py: Строка 81: Слишком длинный комментарий

```
If you accept the risk or you do not use marshalling unlock html responses
```

./path/to/http-api-3.1//http-api-3.1/restapi/rest/definition.py: Строка 101: Слишком длинный комментарий

```
This function has to be coupled with a marshal_with(TotalSchema, code=206)
```

./path/to/http-api-3.1//http-api-3.1/restapi/rest/loader.py: Строка 26: Превышена максимальная длина (>79 символов)

```
| 26 -> ERR400 = {"description": "The request cannot be satisfied due to malformed syntax"}
```

./path/to/http-api-3.1//http-api-3.1/restapi/rest/loader.py: Строка 28: Превышена максимальная длина (>79 символов)

```
| 28 -> ERR404_AUTH = {"description": "The resource cannot be found or you are not authorized"}
```

./path/to/http-api-3.1//http-api-3.1/restapi/rest/loader.py: Строка 54: Превышена максимальная длина (>79 символов)

```
| 54 -> # Used by swagger spec endpoint to remove private endpoints from public requests
```

./path/to/http-api-3.1//http-api-3.1/restapi/rest/loader.py: Строка 123: Превышена максимальная длина (>79 символов)

```
| 123 -> def extract_endpoints(self, base_dir: Path) -> list[type[EndpointResource]]:
```

./path/to/http-api-3.1//http-api-3.1/restapi/rest/loader.py: Строка 229: Превышена максимальная длина (>79 символов)

```
| 229 -> "Missing user: {} parameter in {}.{} (function {})",
```


./path/to/http-api-3.1//http-api-3.1/restapi/rest/loader.py: Строка 238: Превышена максимальная длина (>79 символов)

```
| 238 -> "Invalid {} endpoint in {}: missing endpoint decorator",
```

./path/to/http-api-3.1//http-api-3.1/restapi/rest/loader.py: Строка 274: Превышена максимальная длина (>79 символов)

```
| 274 -> inject_apispec_docs(fn, {"responses": responses}, epclss.labels)
```

./path/to/http-api-3.1//http-api-3.1/restapi/rest/loader.py: Строка 280: Превышена максимальная длина (>79 символов)

```
| 280 -> self.private_endpoints[fn.uri].setdefault(method_fn, endpoint.private)
```

./path/to/http-api-3.1//http-api-3.1/restapi/rest/loader.py: Строка 318: Превышена максимальная длина (>79 символов)

```
| 318 -> "Endpoint redefinition: {} {} used from both {} and {}",
```

./path/to/http-api-3.1//http-api-3.1/restapi/rest/loader.py: Строка 337: Превышена максимальная длина (>79 символов)

```
| 337 -> # split url tokens and remove the first (always empty) token
```

./path/to/http-api-3.1//http-api-3.1/restapi/rest/loader.py: Строка 340: Превышена максимальная длина (>79 символов)

```
| 340 -> # If number of tokens differens, there cannot be any collision
```

./path/to/http-api-3.1//http-api-3.1/restapi/rest/loader.py: Строка 361: Превышена максимальная длина (>79 символов)

```
| 361 -> "Endpoint shadowing detected: {}({m} {}) and {}({m} {})",
```

./path/to/http-api-3.1//http-api-3.1/restapi/rest/loader.py: Переменная ERR401: Некорректное именование
ERR401

./path/to/http-api-3.1//http-api-3.1/restapi/rest/loader.py: Переменная ERR400: Некорректное именование
ERR400

./path/to/http-api-3.1//http-api-3.1/restapi/rest/loader.py: Переменная ERR404: Некорректное именование
ERR404

./path/to/http-api-3.1//http-api-3.1/restapi/rest/loader.py: Переменная ERR404_AUTH: Некорректное именование
ERR404_AUTH

./path/to/http-api-3.1//http-api-3.1/restapi/rest/loader.py: Переменная CONF_FOLDERS: Некорректное именование
CONF_FOLDERS

./path/to/http-api-3.1//http-api-3.1/restapi/rest/loader.py: Строка 54: Слишком длинный комментарий
Used by swagger spec endpoint to remove private endpoints from public requests

./path/to/http-api-3.1//http-api-3.1/restapi/rest/response.py: Строка 125: Превышена максимальная длина (>79 символов)

```
| 125 -> if request.mimetype in ["application/octet-stream",  
"multipart/form-data"]:
```

./path/to/http-api-3.1//http-api-3.1/restapi/rest/response.py: Строка 265: Превышена максимальная длина (>79 символов)

```
| 265 -> # Do not compress binary contents (like images) due to small benefits expected
```

./path/to/http-api-3.1//http-api-3.1/restapi/rest/response.py: Строка 277: Превышена максимальная длина (>79 символов)

```
| 277 -> # compresslevel: an integer from 0 to 9 controlling the level of compression;
```

./path/to/http-api-3.1//http-api-3.1/restapi/rest/response.py: Строка 308: Превышена максимальная длина (>79 символов)

```
| 308 -> # This a debug code use to detect content types that should not be compressed
```

./path/to/http-api-3.1//http-api-3.1/restapi/rest/response.py: Строка 311: Превышена максимальная длина (>79 символов)

```
| 311 -> "Small benefit due to gzip compression on Content-Type: {} ({:.2f} %)",
```

./path/to/http-api-3.1//http-api-3.1/restapi/rest/response.py: Строка 338: Превышена максимальная длина (>79 символов)

```
| 338 -> f["extra_descriptions"] = field_def.metadata["extra_descriptions"]
```

./path/to/http-api-3.1//http-api-3.1/restapi/rest/response.py: Строка 344: Превышена максимальная длина (>79 символов)

```
| 344 -> if autocomplete_endpoint := field_def.metadata.get("autocomplete_endpoint"):
```

./path/to/http-api-3.1//http-api-3.1/restapi/rest/response.py: Строка 350: Превышена максимальная длина (>79 символов)

```
| 350 -> if autocomplete_id_bind := field_def.metadata.get("autocomplete_id_bind"):
```

./path/to/http-api-3.1//http-api-3.1/restapi/rest/response.py: Строка 360: Превышена максимальная длина (>79 символов)

```
| 360 -> elif not isinstance(field_def.load_default, _Missing): # pragma: no cover
```

./path/to/http-api-3.1//http-api-3.1/restapi/rest/response.py: Строка 367: Превышена максимальная длина (>79 символов)

```
| 367 -> # activated in case of fields.List(fields.Something) with an inner validator
```

./path/to/http-api-3.1//http-api-3.1/restapi/rest/response.py: Строка 421: Превышена максимальная длина (>79 символов)

```
| 421 -> content = {"Server internal error": "Failed to retrieve input schema"}
```

./path/to/http-api-3.1//http-api-3.1/restapi/rest/response.py: Строка 426: Превышена максимальная длина (>79 символов)

```
| 426 -> field: str, schema: marshmallow_fields.Field, default: Optional[Any] = None
```

./path/to/http-api-3.1//http-api-3.1/restapi/rest/response.py: Строка 432: Превышена максимальная длина (>79 символов)

```
| 432 -> if isinstance(schema, fields.Bool) or isinstance(schema, fields.Boolean):
```

./path/to/http-api-3.1//http-api-3.1/restapi/rest/response.py: Строка 449: Превышена максимальная длина (>79 символов)

```
| 449 -> if isinstance(schema, fields.Int) or isinstance(schema, fields.Integer):
```

./path/to/http-api-3.1//http-api-3.1/restapi/rest/response.py: Строка 453: Превышена максимальная длина (>79 символов)

```
| 453 -> inner_type = ResponseMaker.get_schema_type(field, schema.inner,
default=key)
```

./path/to/http-api-3.1//http-api-3.1/restapi/rest/response.py: Переменная PROJECT_VERSION: Некорректное ИМЕНОВАНИЕ

PROJECT_VERSION

./path/to/http-api-3.1//http-api-3.1/restapi/rest/response.py: Строка 265: Слишком длинный комментарий

```
Do not compress binary contents (like images) due to small benefits expected
```

./path/to/http-api-3.1//http-api-3.1/restapi/rest/response.py: Строка 277: Слишком длинный комментарий

```
compresslevel: an integer from 0 to 9 controlling the level of compression;
```

./path/to/http-api-3.1//http-api-3.1/restapi/rest/response.py: Строка 308: Слишком длинный комментарий

```
This a debug code use to detect content types that should not be compressed
```

./path/to/http-api-3.1//http-api-3.1/restapi/rest/response.py: Строка 367: Слишком длинный комментарий

```
activated in case of fields.List(fields.Something) with an inner validator
```

./path/to/http-api-3.1//http-api-3.1/restapi/server.py: Строка 76: Превышена максимальная длина (>79 СИМВОЛОВ)

```
| 76 -> # This is needed to let connectors to complete the disconnection and
prevent
```

./path/to/http-api-3.1//http-api-3.1/restapi/server.py: Строка 93: Превышена максимальная длина (>79 СИМВОЛОВ)

```
| 93 -> if PRODUCTION and TESTING and not FORCE_PRODUCTION_TESTS: # pragma: no
cover
```

./path/to/http-api-3.1//http-api-3.1/restapi/server.
py: Строка 104: Превышена максимальная длина (>79
СИМВОЛОВ)

```
| 104 -> flask_app = Flask(name,  
template_folder=str(ABS_RESTAPI_PATH.joinpath("templates")))
```

./path/to/http-api-3.1//http-api-3.1/restapi/server.
py: Строка 112: Превышена максимальная длина (>79
СИМВОЛОВ)

```
| 112 -> # Beware, this only works because get_frontend_url never append a port
```

./path/to/http-api-3.1//http-api-3.1/restapi/server.
py: Строка 165: Превышена максимальная длина (>79
СИМВОЛОВ)

```
| 165 -> print_and_exit("Invalid Customizer class, it should inherit  
BaseCustomizer")
```

./path/to/http-api-3.1//http-api-3.1/restapi/server.
py: Строка 175: Превышена максимальная длина (>79
СИМВОЛОВ)

```
| 175 -> # Restful plugin with endpoint mapping (skipped in INIT|DESTROY|WORKER  
modes)
```

./path/to/http-api-3.1//http-api-3.1/restapi/server.
py: Строка 190: Превышена максимальная длина (>79
СИМВОЛОВ)

```
| 190 -> # DeprecationWarning: Base category for warnings about deprecated  
features when
```

./path/to/http-api-3.1//http-api-3.1/restapi/server.
py: Строка 191: Превышена максимальная длина (>79
СИМВОЛОВ)

```
| 191 -> # those warnings are intended for other Python developers
```

./path/to/http-api-3.1//http-api-3.1/restapi/server.
py: Строка 192: Превышена максимальная длина (>79
СИМВОЛОВ)

```
| 192 -> # SyntaxWarning: Base category for warnings about dubious syntactic features.
```

```
./path/to/http-api-3.1//http-api-3.1/restapi/server.  
py: Строка 193: Превышена максимальная длина (>79  
СИМВОЛОВ)
```

```
| 193 -> # RuntimeWarning: Base category for warnings about dubious runtime features.
```

```
./path/to/http-api-3.1//http-api-3.1/restapi/server.  
py: Строка 194: Превышена максимальная длина (>79  
СИМВОЛОВ)
```

```
| 194 -> # FutureWarning: Base category for warnings about deprecated features  
when those
```

```
./path/to/http-api-3.1//http-api-3.1/restapi/server.  
py: Строка 196: Превышена максимальная длина (>79  
СИМВОЛОВ)
```

```
| 196 -> # PendingDeprecationWarning: Base category for warnings about features  
that will
```

```
./path/to/http-api-3.1//http-api-3.1/restapi/server.  
py: Строка 197: Превышена максимальная длина (>79  
СИМВОЛОВ)
```

```
| 197 -> # be deprecated in the future (ignored by default).
```

```
./path/to/http-api-3.1//http-api-3.1/restapi/server.  
py: Строка 198: Превышена максимальная длина (>79  
СИМВОЛОВ)
```

```
| 198 -> # ImportWarning: Base category for warnings triggered during the process  
of
```

```
./path/to/http-api-3.1//http-api-3.1/restapi/server.  
py: Строка 201: Превышена максимальная длина (>79  
СИМВОЛОВ)
```

```
| 201 -> # BytesWarning: Base category for warnings related to bytes and  
bytearray.
```

./path/to/http-api-3.1//http-api-3.1/restapi/server.
py: Строка 217: Превышена максимальная длина (>79
СИМВОЛОВ)

```
| 217 -> # a lot of issues like: https://github.com/pallets/flask/issues/2468
```

./path/to/http-api-3.1//http-api-3.1/restapi/server.
py: Строка 263: Превышена максимальная длина (>79
СИМВОЛОВ)

```
| 263 -> message="SelectableGroups dict interface is deprecated. Use select.",
```

./path/to/http-api-3.1//http-api-3.1/restapi/server.
py: Строка 269: Превышена максимальная длина (>79
СИМВОЛОВ)

```
| 269 -> message="Relying on Driver's destructor to close the session is  
deprecated.",
```

./path/to/http-api-3.1//http-api-3.1/restapi/server.
py: Строка 281: Превышена максимальная длина (>79
СИМВОЛОВ)

```
| 281 -> message="datetime.datetime.utcnow() is deprecated and scheduled for  
removal in a future version", # noqa
```

./path/to/http-api-3.1//http-api-3.1/restapi/server.
py: Строка 305: Превышена максимальная длина (>79
СИМВОЛОВ)

```
| 305 -> version=get_project_configuration("project.version", default="0.0.1"),
```

./path/to/http-api-3.1//http-api-3.1/restapi/server.
py: Строка 322: Превышена максимальная длина (>79
СИМВОЛОВ)

```
| 322 -> api_key_scheme = {"type": "apiKey", "in": "header", "name":  
"Authorization"}
```

./path/to/http-api-3.1//http-api-3.1/restapi/server.
py: Строка 366: Превышена максимальная длина (>79
СИМВОЛОВ)

```
| 366 -> log.error("Cannot register {}: {}", endpoint.cls.__name__, e)
```


./path/to/http-api-3.1//http-api-3.1/restapi/server.
py: Строка 394: Превышена максимальная длина (>79
СИМВОЛОВ)

```
| 394 -> log.info("Boot completed (version {}, rapydo {})", version,  
restapi_version)
```

./path/to/http-api-3.1//http-api-3.1/restapi/server.
py: Строка 395: Превышена максимальная длина (>79
СИМВОЛОВ)

```
| 395 -> if PRODUCTION and not TESTING and name == MAIN_SERVER_NAME: # pragma: no  
cover
```

./path/to/http-api-3.1//http-api-3.1/restapi/server.
py: Переменная NORMAL: Некорректное именован
NORMAL

./path/to/http-api-3.1//http-api-3.1/restapi/server.
py: Переменная INIT: Некорректное именован
INIT

./path/to/http-api-3.1//http-api-3.1/restapi/server.
py: Переменная DESTROY: Некорректное именован
DESTROY

./path/to/http-api-3.1//http-api-3.1/restapi/server.
py: Переменная WORKER: Некорректное именован
WORKER

./path/to/http-api-3.1//http-api-3.1/restapi/server.
py: Строка 76: Слишком длинный комментарий

```
This is needed to let connectors to complete the disconnection and prevent
```

./path/to/http-api-3.1//http-api-3.1/restapi/server.
py: Строка 175: Слишком длинный комментарий

```
Restful plugin with endpoint mapping (skipped in INIT|DESTROY|WORKER modes)
```

./path/to/http-api-3.1//http-api-3.1/restapi/server.
py: Строка 190: Слишком длинный комментарий

DeprecationWarning: Base category for warnings about deprecated features when

./path/to/http-api-3.1//http-api-3.1/restapi/server.
py: Строка 192: Слишком длинный комментарий

SyntaxWarning: Base category for warnings about dubious syntactic features.

./path/to/http-api-3.1//http-api-3.1/restapi/server.
py: Строка 193: Слишком длинный комментарий

RuntimeWarning: Base category for warnings about dubious runtime features.

./path/to/http-api-3.1//http-api-3.1/restapi/server.
py: Строка 194: Слишком длинный комментарий

FutureWarning: Base category for warnings about deprecated features when those

./path/to/http-api-3.1//http-api-3.1/restapi/server.
py: Строка 196: Слишком длинный комментарий

PendingDeprecationWarning: Base category for warnings about features that will

./path/to/http-api-3.1//http-api-3.1/restapi/server.
py: Строка 198: Слишком длинный комментарий

ImportWarning: Base category for warnings triggered during the process of

./path/to/http-api-3.1//http-api-3.1/restapi/service
s/authentication.py: Строка 99: Превышена
максимальная длина (>79 символов)

| 99 -> # Produced by unpack_token. Datetimes are converted to int as specified in
rfc7519

./path/to/http-api-3.1//http-api-3.1/restapi/service
s/authentication.py: Строка 133: Превышена
максимальная длина (>79 символов)

| 133 -> #
#####

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 161: Превышена максимальная длина (>79 символов)

```
| 161 -> #  
#####
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 181: Превышена максимальная длина (>79 символов)

```
| 181 -> GRACE_PERIOD =  
timedelta(seconds=Env.get_int("AUTH_TOKEN_IP_GRACE_PERIOD", 7200))
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 243: Превышена максимальная длина (>79 символов)

```
| 243 -> BaseAuthentication.default_password = Env.get("AUTH_DEFAULT_PASSWORD",  
"")
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 264: Превышена максимальная длина (>79 символов)

```
| 264 -> mem.configuration, "variables.roles_descriptions", default=empty_dict
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 291: Превышена максимальная длина (>79 символов)

```
| 291 -> raise ServiceUnavailable("Unable to connect to auth backend") from e
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 326: Превышена максимальная длина (>79 символов)

```
| 326 -> payload, full_payload = self.fill_payload(user,  
expiration=user.expiration)
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 347: Превышена

максимальная длина (>79 символов)

```
| 347 -> hashed_password = BaseAuthentication.get_password_hash(hashed_password)
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 349: Превышена
максимальная длина (>79 символов)

```
| 349 -> hashed_password_byte_enc = hashed_password.encode("utf-8",  
errors="strict")
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 351: Превышена
максимальная длина (>79 символов)

```
| 351 -> password=password_byte_enc, hashed_password=hashed_password_byte_enc
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 369: Превышена
максимальная длина (>79 символов)

```
| 369 -> # , If a request goes through multiple proxies, the
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 370: Превышена
максимальная длина (>79 символов)

```
| 370 -> # IP addresses of each successive proxy is listed. This means, the
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 371: Превышена
максимальная длина (>79 символов)

```
| 371 -> # right-most IP address is the IP address of the most recent proxy
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 372: Превышена
максимальная длина (>79 символов)

```
| 372 -> # and the left-most IP address is the IP address of the originating
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 383: Превышена

максимальная длина (>79 символов)

```
| 383 -> if real_ip := request.headers.get(header_key): # pragma: no cover
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 386: Превышена максимальная длина (>79 символов)

```
| 386 -> if raise_warnings and PRODUCTION and not TESTING: # pragma: no cover
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 388: Превышена максимальная длина (>79 символов)

```
| 388 -> "Production mode is enabled, but {} header is missing", header_key
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 399: Превышена максимальная длина (>79 символов)

```
| 399 -> # Mocked IP to prevent tests failures when fn executed outside Flask context
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 409: Превышена максимальная длина (>79 символов)

```
| 409 -> cast(dict[str, Any], payload), cls.JWT_SECRET, algorithm=cls.JWT_ALGO
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 497: Превышена максимальная длина (>79 символов)

```
| 497 -> log.error("Invalid token type {}, required: {}", payload_type, token_type)
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 518: Превышена максимальная длина (>79 символов)

```
| 518 -> return self.unpacked_token(True, token=token, jti=payload["jti"], user=user)
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 623: Превышена максимальная длина (>79 символов)

```
| 623 -> log.critical("Unknown role authorization requirement: {}",  
required_roles)
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 631: Превышена максимальная длина (>79 символов)

```
| 631 -> userdata, extradata =  
mem.customizer.custom_user_properties_pre(userdata)
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 635: Превышена максимальная длина (>79 символов)

```
| 635 -> raise BadRequest(f"Unable to pre-customize user properties: {e}") from e
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 653: Превышена максимальная длина (>79 символов)

```
| 653 -> raise BadRequest(f"Unable to post-customize user properties: {e}") from  
e
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 661: Превышена максимальная длина (>79 символов)

```
| 661 -> def register_failed_login(self, username: str, user: Optional[User]) ->  
None:
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 672: Превышена максимальная длина (>79 символов)

```
| 672 -> "Reached the maximum number of failed login, account {} is blocked",
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 678: Превышена максимальная длина (>79 символов)

```
| 678 -> from restapi.connectors.smtp.notifications import notify_login_block
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 753: Превышена максимальная длина (>79 символов)

```
| 753 -> project_name = get_project_configuration("project.title", "No project name")
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 763: Превышена максимальная длина (>79 символов)

```
| 763 -> self, pwd: str, old_pwd: Optional[str], email: str, name: str, surname: str
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 767: Превышена максимальная длина (>79 символов)

```
| 767 -> return False, "The new password cannot match the previous password"
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 770: Превышена максимальная длина (>79 символов)

```
| 770 -> return False, "The new password cannot match the previous password"
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 774: Превышена максимальная длина (>79 символов)

```
| 774 -> return False, f"Password is too short, use at least {MIN} characters"
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 800: Превышена максимальная длина (>79 символов)

```
| 800 -> return False, "Password is too weak, can't contain your email address"
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 849: Превышена

максимальная длина (>79 символов)

```
| 849 -> # Check if something is missing in the authentication and ask additional actions
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 872: Превышена максимальная длина (>79 символов)

```
| 872 -> message["errors"].append("Your password is expired, please change it")
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 905: Превышена максимальная длина (>79 символов)

```
| 905 -> payload={"username": user.email, "motivation": "account not active"},
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 916: Превышена максимальная длина (>79 символов)

```
| 916 -> log.info("Default user can't be blocked for inactivity during tests")
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 919: Превышена максимальная длина (>79 символов)

```
| 919 -> if user.last_login + cls.DISABLE_UNUSED_CREDENTIALS_AFTER < now:
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 927: Превышена максимальная длина (>79 символов)

```
| 927 -> raise Forbidden("Sorry, this account is blocked for inactivity")
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 931: Превышена максимальная длина (>79 символов)

```
| 931 -> # tzinfo should be the same for both last_login and expiration fields
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 938: Превышена

максимальная длина (>79 символов)

```
| 938 -> payload={"username": user.email, "motivation": "account expired"},
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 968: Превышена
максимальная длина (>79 символов)

```
| 968 -> default_group = self.init_groups(force=options.get("force_group",  
False))
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 981: Превышена
максимальная длина (>79 символов)

```
| 981 -> print_and_exit("Found duplicated role names: {}",  
str(sorted(role_names)))
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 989: Превышена
максимальная длина (>79 символов)

```
| 989 -> log.info("Role {} already exists, updating description", role_name)
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 1011: Превышена
максимальная длина (>79 символов)

```
| 1011 -> # If there are some groups skip group creation in absence of a force  
flag
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 1013: Превышена
максимальная длина (>79 символов)

```
| 1013 -> # If force flag is enable, create the default group if missing or update  
it
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 1041: Превышена
максимальная длина (>79 символов)

```
| 1041 -> def init_users(self, default_group: Group, roles: list[str], force:  
bool) -> User:
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 1052: Превышена максимальная длина (>79 символов)

```
| 1052 -> # If force flag is enable, create the default user if missing or update it
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 1084: Превышена максимальная длина (>79 символов)

```
| 1084 -> default_user.password = self.get_password_hash(self.default_password)
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 1154: Превышена максимальная длина (>79 символов)

```
| 1154 -> How to retrieve the group that the user belongs to from the current auth db
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 1193: Превышена максимальная длина (>79 символов)

```
| 1193 -> self, user: User, token: str, payload: Payload, token_type: Optional[str] = None
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 1263: Превышена максимальная длина (>79 символов)

```
| 1263 -> def save_login(self, username: str, user: Optional[User], failed: bool) -> None:
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 1291: Превышена максимальная длина (>79 символов)

```
| 1291 -> raise NotImplementedError("Create User not implemented with No Authentication")
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 1297: Превышена

максимальная длина (>79 символов)

```
| 1297 -> raise NotImplementedError("Create Group not implemented with No Authentication")
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 1325: Превышена максимальная длина (>79 символов)

```
| 1325 -> raise NotImplementedError("Get Group not implemented with No Authentication")
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 1349: Превышена максимальная длина (>79 символов)

```
| 1349 -> self, user: User, token: str, payload: Payload, token_type: Optional[str] = None
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 1367: Превышена максимальная длина (>79 символов)

```
| 1367 -> def save_login(self, username: str, user: Optional[User], failed: bool) -> None:
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 1373: Превышена максимальная длина (>79 символов)

```
| 1373 -> raise NotImplementedError("Get Login not implemented with No Authentication")
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Переменная User: Некорректное именован

User

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Переменная Group: Некорректное именован

Group

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Переменная RoleObj:
Некорректное именован

RoleObj

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Переменная Login: Некорректное
ИМЕНОВАНИЕ

Login

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Переменная ALL_ROLES:
Некорректное именован

ALL_ROLES

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Переменная ANY_ROLE:
Некорректное именован

ANY_ROLE

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Переменная ROLE_DISABLED:
Некорректное именован

ROLE_DISABLED

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Переменная DEFAULT_GROUP_NAME:
Некорректное именован

DEFAULT_GROUP_NAME

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Переменная DEFAULT_GROUP_DESCR:
Некорректное именован

DEFAULT_GROUP_DESCR

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Переменная
DISABLE_UNUSED_CREDENTIALS_AFTER_MIN_TESTNIG_VALUE:
Некорректное именован

DISABLE_UNUSED_CREDENTIALS_AFTER_MIN_TESTNIG_VALUE

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Переменная
MAX_PASSWORD_VALIDITY_MIN_TESTNIG_VALUE:
Некорректное именован

MAX_PASSWORD_VALIDITY_MIN_TESTNIG_VALUE

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Переменная
MAX_LOGIN_ATTEMPTS_MIN_TESTING_VALUE: Некорректное
именован

MAX_LOGIN_ATTEMPTS_MIN_TESTING_VALUE

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Переменная
LOGIN_BAN_TIME_MAX_TESTING_VALUE: Некорректное
именован

LOGIN_BAN_TIME_MAX_TESTING_VALUE

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Переменная IP: Некорректное
именован

IP

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Переменная ADMIN: Некорректное
именован

ADMIN

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Переменная COORDINATOR:

Некорректное именованние

COORDINATOR

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Переменная STAFF: Некорректное именованние

STAFF

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Переменная USER: Некорректное именованние

USER

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Переменная JWT_SECRET: Некорректное именованние

JWT_SECRET

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Переменная JWT_ALGO: Некорректное именованние

JWT_ALGO

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Переменная DEFAULT_TOKEN_TTL: Некорректное именованние

DEFAULT_TOKEN_TTL

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Переменная GRACE_PERIOD: Некорректное именованние

GRACE_PERIOD

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Переменная

SAVE_LAST_ACCESS_EVERY: Некорректное именован

SAVE_LAST_ACCESS_EVERY

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Переменная FULL_TOKEN:

Некорректное именован

FULL_TOKEN

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Переменная PWD_RESET:

Некорректное именован

PWD_RESET

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Переменная ACTIVATE_ACCOUNT:

Некорректное именован

ACTIVATE_ACCOUNT

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Переменная UNLOCK_CREDENTIALS:

Некорректное именован

UNLOCK_CREDENTIALS

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Переменная TOTP: Некорректное

именован

TOTP

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Переменная MIN_PASSWORD_LENGTH:

Некорректное именован

MIN_PASSWORD_LENGTH

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Переменная

SECOND_FACTOR_AUTHENTICATION: Некорректное
именование

SECOND_FACTOR_AUTHENTICATION

./path/to/http-api-3.1//http-api-3.1/restapi/service
s/authentication.py: Переменная

TOTP_VALIDITY_WINDOW: Некорректное имя

TOTP_VALIDITY_WINDOW

./path/to/http-api-3.1//http-api-3.1/restapi/service
s/authentication.py: Переменная

FORCE_FIRST_PASSWORD_CHANGE: Некорректное имя

FORCE_FIRST_PASSWORD_CHANGE

./path/to/http-api-3.1//http-api-3.1/restapi/service
s/authentication.py: Переменная

MAX_PASSWORD_VALIDITY: Некорректное имя

MAX_PASSWORD_VALIDITY

./path/to/http-api-3.1//http-api-3.1/restapi/service
s/authentication.py: Переменная

DISABLE_UNUSED_CREDENTIALS_AFTER: Некорректное
именование

DISABLE_UNUSED_CREDENTIALS_AFTER

./path/to/http-api-3.1//http-api-3.1/restapi/service
s/authentication.py: Переменная

MAX_LOGIN_ATTEMPTS: Некорректное имя

MAX_LOGIN_ATTEMPTS

./path/to/http-api-3.1//http-api-3.1/restapi/service
s/authentication.py: Переменная

FAILED_LOGINS_EXPIRATION: Некорректное имя

FAILED_LOGINS_EXPIRATION

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Переменная MIN_CONTAINED_LEN: Некорректное именование

MIN_CONTAINED_LEN

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Переменная MIN: Некорректное именование

MIN

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 99: Слишком длинный комментарий

Produced by unpack_token. Datetimes are converted to int as specified in rfc7519

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 399: Слишком длинный комментарий

Mocked IP to prevent tests failures when fn executed outside Flask context

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 849: Слишком длинный комментарий

Check if something is missing in the authentication and ask additional actions

./path/to/http-api-3.1//http-api-3.1/restapi/services/authentication.py: Строка 1013: Слишком длинный комментарий

If force flag is enable, create the default group if missing or update it

./path/to/http-api-3.1//http-api-3.1/restapi/services/cache.py: Строка 52: Превышена максимальная длина (>79 символов)

| 52 -> def invalidate(func: Callable[[Any], Any], *args: Any, **kwargs: Any) -> None:

./path/to/http-api-3.1//http-api-3.1/restapi/services/download.py: Строка 50: Превышена максимальная длина (>79 символов)

```
| 50 -> # This function is mainly used for displayable files like images and video
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/download.py: Строка 84: Превышена максимальная длина (>79 символов)

```
| 84 -> log.info("Providing streamed content from {} (mime={})", filepath, mime)
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/download.py: Строка 94: Превышена максимальная длина (>79 символов)

```
| 94 -> response.headers["Content-Disposition"] = f"attachment;
filename={out_filename}"
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/download.py: Переменная DEFAULT_CHUNK_SIZE:
Некорректное именован
DEFAULT_CHUNK_SIZE

./path/to/http-api-3.1//http-api-3.1/restapi/services/uploader.py: Строка 10: Превышена максимальная длина (>79 символов)

```
| 10 -> from restapi.exceptions import BadRequest, Conflict, Forbidden,
ServiceUnavailable
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/uploader.py: Строка 28: Превышена максимальная длина (>79 символов)

```
| 28 -> "." in filename and filename.rsplit(".", 1)[1].lower() in
self.allowed_exts
```

./path/to/http-api-3.1//http-api-3.1/restapi/services/uploader.py: Строка 55: Превышена максимальная длина (>79 символов)

```
| 55 -> return {"type": out[0].strip(), "charset": out[1].split("=")[1].strip()}
```

./path/to/http-api-3.1//http-api-3.1/restapi/service
s/uploader.py: Строка 86: Превышена максимальная
длина (>79 символов)

```
| 86 -> f"File '{fname}' already exists, use force parameter to overwrite"
```

./path/to/http-api-3.1//http-api-3.1/restapi/service
s/uploader.py: Строка 154: Превышена максимальная
длина (>79 символов)

```
| 154 -> headers={"Access-Control-Expose-Headers": "Location", "Location": url},
```

./path/to/http-api-3.1//http-api-3.1/restapi/service
s/uploader.py: Строка 200: Превышена максимальная
длина (>79 символов)

```
| 200 -> # the file contents can be accessed using either request.data or  
request.stream.
```

./path/to/http-api-3.1//http-api-3.1/restapi/service
s/uploader.py: Строка 249: Превышена максимальная
длина (>79 символов)

```
| 249 -> {"filename": filename, "meta": self.get_file_metadata(file_path)},
```

./path/to/http-api-3.1//http-api-3.1/restapi/service
s/uploader.py: Переменная DEFAULT_PERMISSIONS:
Некорректное именование
DEFAULT_PERMISSIONS

./path/to/http-api-3.1//http-api-3.1/restapi/service
s/uploader.py: Строка 200: Слишком длинный
комментарий

```
the file contents can be accessed using either request.data or request.stream.
```

./path/to/http-api-3.1//http-api-3.1/restapi/tests.p
y: Строка 58: Превышена максимальная длина (>79
символов)

```
| 58 -> ["date", "ip", "user", "event", "target_type", "target_id", "url",  
"payload"],
```

./path/to/http-api-3.1//http-api-3.1/restapi/tests.p
у: Строка 96: Превышена максимальная длина (>79
СИМВОЛОВ)

```
| 96 -> raise AttributeError(f"Class variable {variable} not found") # pragma: no  
cover
```

./path/to/http-api-3.1//http-api-3.1/restapi/tests.p
у: Строка 113: Превышена максимальная длина (>79
СИМВОЛОВ)

```
| 113 -> f"{API_URI}/{endpoint}", json={"get_schema": 1}, headers=headers
```

./path/to/http-api-3.1//http-api-3.1/restapi/tests.p
у: Строка 117: Превышена максимальная длина (>79
СИМВОЛОВ)

```
| 117 -> f"{API_URI}/{endpoint}", json={"get_schema": 1}, headers=headers
```

./path/to/http-api-3.1//http-api-3.1/restapi/tests.p
у: Строка 179: Превышена максимальная длина (>79
СИМВОЛОВ)

```
| 179 -> if not Connector.check_availability("authentication"): # pragma: no  
cover
```

./path/to/http-api-3.1//http-api-3.1/restapi/tests.p
у: Строка 203: Превышена максимальная длина (>79
СИМВОЛОВ)

```
| 203 -> # This 403 is expected, return an invalid value or you can enter a loop!
```

./path/to/http-api-3.1//http-api-3.1/restapi/tests.p
у: Строка 228: Превышена максимальная длина (>79
СИМВОЛОВ)

```
| 228 -> data["password_confirm"] = cls.faker.password(strong=True)
```

./path/to/http-api-3.1//http-api-3.1/restapi/tests.p
у: Строка 240: Превышена максимальная длина (>79

СИМВОЛОВ)

```
| 240 -> # Test failure of password change if TOTP is wrong or missing
```

./path/to/http-api-3.1//http-api-3.1/restapi/tests.p

у: Строка 258: Превышена максимальная длина (>79 СИМВОЛОВ)

```
| 258 -> cls.faker.pyint(min_value=100000, max_value=999999)
```

./path/to/http-api-3.1//http-api-3.1/restapi/tests.p

у: Строка 268: Превышена максимальная длина (>79 СИМВОЛОВ)

```
| 268 -> # Change the password to silence FIRST_LOGIN and PASSWORD_EXPIRED
```

./path/to/http-api-3.1//http-api-3.1/restapi/tests.p

у: Строка 341: Превышена максимальная длина (>79 СИМВОЛОВ)

```
| 341 -> schema = cls.get_dynamic_input_schema(client, "admin/users",  
admin_headers)
```

./path/to/http-api-3.1//http-api-3.1/restapi/tests.p

у: Строка 361: Превышена максимальная длина (>79 СИМВОЛОВ)

```
| 361 -> r = client.post(f"{API_URI}/admin/users", json=user_data,  
headers=admin_headers)
```

./path/to/http-api-3.1//http-api-3.1/restapi/tests.p

у: Строка 374: Превышена максимальная длина (>79 СИМВОЛОВ)

```
| 374 -> r = client.delete(f"{API_URI}/admin/users/{uuid}",  
headers=admin_headers)
```

./path/to/http-api-3.1//http-api-3.1/restapi/tests.p

у: Строка 385: Превышена максимальная длина (>79 СИМВОЛОВ)

```
| 385 -> schema = cls.get_dynamic_input_schema(client, "admin/groups",  
admin_headers)
```

./path/to/http-api-3.1//http-api-3.1/restapi/tests.p
у: Строка 412: Превышена максимальная длина (>79
СИМВОЛОВ)

```
| 412 -> return cls.get_first_name(faker, recursion=recursion + 1) # pragma: no  
cover
```

./path/to/http-api-3.1//http-api-3.1/restapi/tests.p
у: Строка 426: Превышена максимальная длина (>79
СИМВОЛОВ)

```
| 426 -> return cls.get_last_name(faker, recursion=recursion + 1) # pragma: no  
cover
```

./path/to/http-api-3.1//http-api-3.1/restapi/tests.p
у: Строка 437: Превышена максимальная длина (>79
СИМВОЛОВ)

```
| 437 -> return cls.get_random_email(faker, name, surname, recursion=recursion +  
1)
```

./path/to/http-api-3.1//http-api-3.1/restapi/tests.p
у: Строка 441: Превышена максимальная длина (>79
СИМВОЛОВ)

```
| 441 -> return cls.get_random_email(faker, name, surname, recursion=recursion +  
1)
```

./path/to/http-api-3.1//http-api-3.1/restapi/tests.p
у: Строка 488: Превышена максимальная длина (>79
СИМВОЛОВ)

```
| 488 -> # pytest.fail(f"BuildData for {key}: invalid options (empty?)")
```

./path/to/http-api-3.1//http-api-3.1/restapi/tests.p
у: Строка 492: Превышена максимальная длина (>79
СИМВОЛОВ)

```
| 492 -> data[key] = cls.faker.pyint(min_value=min_value, max_value=max_value)
```

./path/to/http-api-3.1//http-api-3.1/restapi/tests.p
у: Строка 546: Превышена максимальная длина (>79
СИМВОЛОВ)

```
| 546 -> data[key] = cls.faker.pystr(min_chars=min_value, max_chars=max_value)
```

./path/to/http-api-3.1//http-api-3.1/restapi/tests.p
у: Строка 554: Превышена максимальная длина (>79
СИМВОЛОВ)

```
| 554 -> # the input can't be automatically set and here is simply ignored
```

./path/to/http-api-3.1//http-api-3.1/restapi/tests.p
у: Строка 555: Превышена максимальная длина (>79
СИМВОЛОВ)

```
| 555 -> log.warning("BuildData for {}: unknow type {}", key, field_type)
```

./path/to/http-api-3.1//http-api-3.1/restapi/tests.p
у: Строка 589: Превышена максимальная длина (>79
СИМВОЛОВ)

```
| 589 -> if "Content-Transfer-Encoding: base64" in data["body"]: # pragma: no  
cover
```

./path/to/http-api-3.1//http-api-3.1/restapi/tests.p
у: Строка 596: Превышена максимальная длина (>79
СИМВОЛОВ)

```
| 596 -> # b64decode gives as output bytes, decode("utf-8") needed to get a string
```

./path/to/http-api-3.1//http-api-3.1/restapi/tests.p
у: Строка 656: Превышена максимальная длина (>79
СИМВОЛОВ)

```
| 656 -> payload: dict[str, Any] = {"user_id": user_id, "jti": str(uuid.uuid4())}
```

./path/to/http-api-3.1//http-api-3.1/restapi/tests.p
у: Строка 698: Превышена максимальная длина (>79
СИМВОЛОВ)

```
| 698 -> # Not efficient read the whole file to get the last lines, to be  
improved!
```

./path/to/http-api-3.1//http-api-3.1/restapi/tests.p
у: Строка 711: Превышена максимальная длина (>79

СИМВОЛОВ)

```
| 711 -> payload = orjson.loads(" ".join(tokens[8:])) if len(tokens) >= 9 else {}
```

./path/to/http-api-3.1//http-api-3.1/restapi/tests.p

у: Строка 743: Превышена максимальная длина (>79

СИМВОЛОВ)

```
| 743 -> def send_task(app: Flask, task_name: str, *args: Any, **kwargs: Any) -> Any:
```

./path/to/http-api-3.1//http-api-3.1/restapi/tests.p

у: Строка 748: Превышена максимальная длина (>79

СИМВОЛОВ)

```
| 748 -> # Мypy complains about: error: "Callable[[], Any]" has no attribute "get"
```

./path/to/http-api-3.1//http-api-3.1/restapi/tests.p

у: Переменная SERVER_URI: Некорректное именование

SERVER_URI

./path/to/http-api-3.1//http-api-3.1/restapi/tests.p

у: Переменная API_URI: Некорректное именование

API_URI

./path/to/http-api-3.1//http-api-3.1/restapi/tests.p

у: Переменная AUTH_URI: Некорректное именование

AUTH_URI

./path/to/http-api-3.1//http-api-3.1/restapi/tests.p

у: Переменная Event: Некорректное именование

Event

./path/to/http-api-3.1//http-api-3.1/restapi/tests.p

у: Функция buildData: Некорректное именование

buildData

./path/to/http-api-3.1//http-api-3.1/restapi/tests.p
у: Переменная USER: Некорректное именован

USER

./path/to/http-api-3.1//http-api-3.1/restapi/tests.p
у: Переменная PWD: Некорректное именован

PWD

./path/to/http-api-3.1//http-api-3.1/restapi/tests_i
nitialization.py: Строка 10: Превышена максимальная
длина (>79 символов)

```
| 10 -> from restapi.services.authentication import DEFAULT_GROUP_NAME,  
BaseAuthentication
```

./path/to/http-api-3.1//http-api-3.1/restapi/types.p
у: Строка 10: Превышена максимальная длина (>79
СИМВОЛОВ)

```
| 10 -> Response = Union[FlaskResponse, WerkzeugResponse, tuple[Any, int,  
dict[str, str]]]
```

./path/to/http-api-3.1//http-api-3.1/restapi/types.p
у: Переменная Response: Некорректное именован

Response

./path/to/http-api-3.1//http-api-3.1/restapi/types.p
у: Переменная ResponseContent: Некорректное
ИМЕНОВАНИЕ

ResponseContent

./path/to/http-api-3.1//http-api-3.1/restapi/types.p
у: Переменная Props: Некорректное именован

Props

./path/to/http-api-3.1//http-api-3.1/restapi/types.p
у: Переменная FlaskRequest: Некорректное именован

FlaskRequest

./path/to/http-api-3.1//http-api-3.1/restapi/types.py: Переменная EndpointFunction: Некорректное именован

EndpointFunction

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/configuration.py: Строка 54: Превышена максимальная длина (>79 символов)

```
| 54 -> print_and_exit("Invalid repository name in extends_from, name is empty")
```

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/configuration.py: Строка 58: Превышена максимальная длина (>79 символов)

```
| 58 -> suggest = "Expected values: 'projects' or  
'submodules/${REPOSITORY_NAME}'"
```

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/configuration.py: Строка 59: Превышена максимальная длина (>79 символов)

```
| 59 -> print_and_exit("Invalid extends_from parameter: {}.\\n{}", extends_from,  
suggest)
```

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/configuration.py: Строка 70: Превышена максимальная длина (>79 символов)

```
| 70 -> def mix(base: ConfigurationType, custom: ConfigurationType) ->  
ConfigurationType:
```

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/configuration.py: Переменная PROJECTS_DEFAULTS_FILE: Некорректное именован

PROJECTS_DEFAULTS_FILE

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/configuration.py: Переменная

PROJECT_CONF_FILENAME: Некорректное именование

PROJECT_CONF_FILENAME

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/configuration.py: Переменная ConfigurationType: Некорректное именование

ConfigurationType

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/faker.py: Строка 82: Превышена максимальная длина (>79 символов)

```
| 82 -> # Password is randomly resampled, can't be sure that will be covered by tests
```

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/faker.py: Строка 87: Превышена максимальная длина (>79 символов)

```
| 87 -> "Password is not strong enough: missing lower case. Sampling again..."
```

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/faker.py: Строка 90: Превышена максимальная длина (>79 символов)

```
| 90 -> length, strong=strong, low=low, up=up, digits=digits, symbols=symbols
```

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/faker.py: Строка 92: Превышена максимальная длина (>79 символов)

```
| 92 -> # Password is randomly resampled, can't be sure that will be covered by tests
```

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/faker.py: Строка 97: Превышена максимальная длина (>79 символов)

```
| 97 -> "Password is not strong enough: missing upper case. Sampling again..."
```

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/faker.py: Строка 100: Превышена максимальная длина (>79 символов)

```
| 100 -> length, strong=strong, low=low, up=up, digits=digits, symbols=symbols
```

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/faker.py: Строка 102: Превышена максимальная длина (>79 символов)

```
| 102 -> # Password is randomly resampled, can't be sure that will be covered by tests
```

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/faker.py: Строка 103: Превышена максимальная длина (>79 символов)

```
| 103 -> if digits and not any(s in randstr for s in string.digits): # pragma: no cover
```

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/faker.py: Строка 105: Превышена максимальная длина (>79 символов)

```
| 105 -> "Password is not strong enough: missing digits. Sampling again..."
```

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/faker.py: Строка 108: Превышена максимальная длина (>79 символов)

```
| 108 -> length, strong=strong, low=low, up=up, digits=digits, symbols=symbols
```

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/faker.py: Строка 110: Превышена максимальная длина (>79 символов)

```
| 110 -> # Password is randomly resampled, can't be sure that will be covered by tests
```

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/faker.py: Строка 115: Превышена максимальная длина (>79 символов)

```
| 115 -> "Password is not strong enough: missing symbols. Sampling again..."
```

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/faker.py: Строка 118: Превышена максимальная длина (>79 символов)

```
| 118 -> length, strong=strong, low=low, up=up, digits=digits, symbols=symbols
```

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/faker.py: Переменная FAKER_LOCALES: Некорректное именование

FAKER_LOCALES

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/faker.py: Строка 82: Слишком длинный комментарий

```
Password is randomly resampled, can't be sure that will be covered by tests
```

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/faker.py: Строка 92: Слишком длинный комментарий

```
Password is randomly resampled, can't be sure that will be covered by tests
```

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/faker.py: Строка 102: Слишком длинный комментарий

```
Password is randomly resampled, can't be sure that will be covered by tests
```

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/faker.py: Строка 110: Слишком длинный комментарий

```
Password is randomly resampled, can't be sure that will be covered by tests
```

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/globals.py: Класс mem: Некорректное именование

mem

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/logs.py: Строка 24: Превышена максимальная длина (>79 символов)

```
| 24 -> if Path(LOGS_PATH).exists() and not os.access(LOGS_PATH, os.W_OK): #  
pragma: no cover
```

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/logs.py: Строка 77: Превышена максимальная длина (>79 символов)

```
| 77 -> # If True the exception trace is extended upward, beyond the catching  
point
```

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/logs.py: Строка 80: Превышена максимальная длина (>79 символов)

```
| 80 -> # Display variables values in exception trace to eases the debugging.
```

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/logs.py: Строка 83: Превышена максимальная длина (>79 символов)

```
| 83 -> # Messages pass through a multiprocess-safe queue before reaching the  
sink
```

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/logs.py: Строка 84: Превышена максимальная длина (>79 символов)

```
| 84 -> # This is useful while logging to a file through multiple processes.
```

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/logs.py: Строка 88: Превышена максимальная длина (>79 символов)

```
| 88 -> # Errors occurring while sink handles logs messages are automatically  
caught
```

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/logs.py: Строка 91: Превышена максимальная длина (>79 символов)

```
| 91 -> # This is the case when pickle fails to serialize before sending to the  
queue
```

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/logs.py: Строка 111: Превышена максимальная длина (>79 символов)

```
| 111 -> # Otherwise in case of missing extra fields the event will be simply ignored
```

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/logs.py: Строка 126: Превышена максимальная длина (>79 символов)

```
| 126 -> # Set the default logger with the given log level and save the log_id as static variable
```

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/logs.py: Строка 127: Превышена максимальная длина (>79 символов)

```
| 127 -> # Further call to this function will remove the previous logger (based on saved log_id)
```

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/logs.py: Строка 138: Превышена максимальная длина (>79 символов)

```
| 138 -> # If True the exception trace is extended upward, beyond the catching point
```

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/logs.py: Строка 155: Превышена максимальная длина (>79 символов)

```
| 155 -> # Can't understand why mypy is unable to understand Env.get_int, since it is annotated
```

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/logs.py: Строка 175: Превышена максимальная длина (>79 символов)

```
| 175 -> def handle_log_output(original_parameters_string: Optional[Any]) -> dict[str, Any]:
```

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/logs.py: Строка 209: Превышена максимальная длина

(>79 СИМВОЛОВ)

```
| 209 -> parameters: dict[str, Any], urlencoded: bool = False, max_len: int = MAX_CHAR_LEN
```

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/logs.py: Строка 211: Превышена максимальная длина (>79 СИМВОЛОВ)

```
| 211 -> """Obfuscate sensitive information in a dictionary by looking at sensitive keys"""
```

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/logs.py: Строка 221: Превышена максимальная длина (>79 СИМВОЛОВ)

```
| 221 -> # urllib.parse.parse_qs converts all elements in single-elements lists...
```

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/logs.py: Переменная LOG_RETENTION: Некорректное ИМЕНОВАНИЕ

LOG_RETENTION

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/logs.py: Переменная FILE_LOGLEVEL: Некорректное ИМЕНОВАНИЕ

FILE_LOGLEVEL

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/logs.py: Переменная LOGS_FOLDER: Некорректное ИМЕНОВАНИЕ

LOGS_FOLDER

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/logs.py: Переменная LOGS_PATH: Некорректное ИМЕНОВАНИЕ

LOGS_PATH

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/logs.py: Переменная EVENTS_PATH: Некорректное именование

EVENTS_PATH

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/logs.py: Переменная MAX_CHAR_LEN: Некорректное именование

MAX_CHAR_LEN

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/logs.py: Переменная OBSCURE_VALUE: Некорректное именование

OBSCURE_VALUE

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/logs.py: Переменная OBSCURED_FIELDS: Некорректное именование

OBSCURED_FIELDS

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/logs.py: Переменная LOGS_PATH: Некорректное именование

LOGS_PATH

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/logs.py: Строка 77: Слишком длинный комментарий

If True the exception trace is extended upward, beyond the catching point

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/logs.py: Строка 88: Слишком длинный комментарий

Errors occurring while sink handles logs messages are automatically caught

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/logs.py: Строка 91: Слишком длинный комментарий

This is the case when pickle fails to serialize before sending to the queue

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/logs.py: Строка 111: Слишком длинный комментарий

Otherwise in case of missing extra fields the event will be simply ignored

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/logs.py: Строка 126: Слишком длинный комментарий

Set the default logger with the given log level and save the log_id as static variable

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/logs.py: Строка 127: Слишком длинный комментарий

Further call to this function will remove the previous logger (based on saved log_id)

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/logs.py: Строка 138: Слишком длинный комментарий

If True the exception trace is extended upward, beyond the catching point

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/logs.py: Строка 155: Слишком длинный комментарий

Can't understand why mypy is unable to understand Env.get_int, since it is annotated

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/meta.py: Строка 5: Превышена максимальная длина (>79 символов)

| 5 -> <http://python-3-patterns-idioms-test.readthedocs.org/en/latest/Metaprogramming.html>

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/meta.py: Строка 40: Превышена максимальная длина (>79 символов)

| 40 -> def get_new_classes_from_module(module: ModuleType) -> dict[str, type[Any]]:

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/meta.py: Строка 102: Превышена максимальная длина (>79 символов)

```
| 102 -> module = Meta.get_module_from_string(module_name, exit_on_fail=True)
```

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/meta.py: Строка 110: Превышена максимальная длина (>79 символов)

```
| 110 -> print_and_exit("Cannot load {} models from {}", name, module_name)
```

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/processes.py: Строка 63: Превышена максимальная длина (>79 символов)

```
| 63 -> d = datetime.fromtimestamp(process.create_time()).strftime("%Y-%m-%d
%H:%M:%S")
```

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/processes.py: Строка 65: Превышена максимальная длина (>79 символов)

```
| 65 -> "Process is running\nPID: {}\nCreated: {}\nCmd: {}", pid, d, cmdline
```

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/processes.py: Строка 100: Превышена максимальная длина (>79 символов)

```
| 100 -> f"{service_name} ({host}:{port}) unavailable after {t} seconds"
```

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/processes.py: Переменная DEFAULT_MAX_RETRIES: Некорректное именован

DEFAULT_MAX_RETRIES

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/processes.py: Переменная SLEEP_TIME: Некорректное именован

SLEEP_TIME

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/processes.py: Переменная TIMEOUT: Некорректное именован

TIMEOUT

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/time.py: Строка 11: Превышена максимальная длина (>79 символов)

```
| 11 -> "days", "seconds", "microseconds", "milliseconds", "minutes", "hours",  
"weeks"
```

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/time.py: Переменная EPOCH: Некорректное именован

EPOCH

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/time.py: Переменная AllowedTimedeltaPeriods: Некорректное именован

AllowedTimedeltaPeriods

./path/to/http-api-3.1//http-api-3.1/restapi/utilities/uuid.py: Функция getUUID: Некорректное именован
getUUID

./path/to/http-api-3.1//http-api-3.1/restapi/__commands__.py: Строка 29: Превышена максимальная длина (>79 символов)

```
| 29 -> return find_process(current_package, keywords=["init"],  
prefix="/usr/local/bin/")
```

./path/to/http-api-3.1//http-api-3.1/restapi/__commands__.py: Строка 87: Превышена максимальная длина (>79 символов)

```
| 87 -> "{} successfully authenticated on {}", service, c.variables.get("host",  
service)
```

./path/to/http-api-3.1//http-api-3.1/restapi/__commands__.py: Строка 121: Превышена максимальная длина (>79 символов)

```
| 121 -> create_app(name="Initializing services", mode=ServerModes.INIT,
options=options)
```

./path/to/http-api-3.1//http-api-3.1/restapi/__commands__.py: Строка 137: Превышена максимальная длина (>79 символов)

```
| 137 -> print_and_exit("Cannot find any variable matching {} for {}", host_var,
service)
```

./path/to/http-api-3.1//http-api-3.1/restapi/__commands__.py: Строка 141: Превышена максимальная длина (>79 символов)

```
| 141 -> print_and_exit("Cannot find any variable matching {} for {}", port_var,
service)
```

./path/to/http-api-3.1//http-api-3.1/restapi/__commands__.py: Строка 155: Превышена максимальная длина (>79 символов)

```
| 155 -> log.info("Service {} is enabled but not tested at startup time", name)
```

./path/to/http-api-3.1//http-api-3.1/restapi/__commands__.py: Строка 166: Превышена максимальная длина (>79 символов)

```
| 166 -> print_and_exit("Invalid celery broker: {}", broker) # pragma: no cover
```

./path/to/http-api-3.1//http-api-3.1/restapi/__commands__.py: Строка 169: Превышена максимальная длина (>79 символов)

```
| 169 -> host, port = get_service_address(service_vars, "host", "port", label)
```

./path/to/http-api-3.1//http-api-3.1/restapi/__commands__.py: Строка 184: Превышена максимальная длина (>79 символов)

```
| 184 -> host, port = get_service_address(service_vars, "host", "port", label)
```

./path/to/http-api-3.1//http-api-3.1/restapi/__commands__.py: Строка 224: Превышена максимальная длина (>79 символов)

```
| 224 -> @click.option("--wait/--no-wait", default=False, help="Wait for startup to finish")
```

./path/to/http-api-3.1//http-api-3.1/restapi/__commands__.py: Строка 226: Превышена максимальная длина (>79 символов)

```
| 226 -> "--core/--no-core", default=False, help="Test for core instead of vanilla code"
```

./path/to/http-api-3.1//http-api-3.1/restapi/__commands__.py: Строка 231: Превышена максимальная длина (>79 символов)

```
| 231 -> "--destroy/--no-destroy", default=False, help="Destroy database after tests"
```

./path/to/http-api-3.1//http-api-3.1/restapi/__commands__.py: Строка 234: Превышена максимальная длина (>79 символов)

```
| 234 -> wait: bool, core: bool, file: Optional[str], folder: Optional[str], destroy: bool
```

./path/to/http-api-3.1//http-api-3.1/restapi/__commands__.py: Строка 238: Превышена максимальная длина (>79 символов)

```
| 238 -> # Forced a change from DEV to TEST mode when using the restapi tests wrapper
```

./path/to/http-api-3.1//http-api-3.1/restapi/__commands__.py: Строка 258: Превышена максимальная длина (>79 символов)

```
| 258 -> "Please specify only one option between --core, --file and --folder"
```

./path/to/http-api-3.1//http-api-3.1/restapi/__commands__.py: Переменная BIND_INTERFACE: Некорректное именован

BIND_INTERFACE

./path/to/http-api-3.1//http-api-3.1/restapi/__commands__.py: Строка 238: Слишком длинный комментарий

Forced a change from DEV to TEST mode when using the restapi tests wrapper

./path/to/http-api-3.1//http-api-3.1/restapi/__init__.py: Переменная FLASK_HELP: Некорректное именован

FLASK_HELP

./path/to/http-api-3.1//http-api-3.1/tests/base/test_aaa_unused_credentials_step1.py: Строка 14: Превышена максимальная длина (>79 символов)

| 14 -> def test_create_credentials(self, client: FlaskClient, faker: Faker) -> None:

./path/to/http-api-3.1//http-api-3.1/tests/base/test_aaa_unused_credentials_step1.py: Строка 31: Превышена максимальная длина (>79 символов)

| 31 -> client, BaseTests.unused_credentials[0], BaseTests.unused_credentials[1]

./path/to/http-api-3.1//http-api-3.1/tests/base/test_cli.py: Строка 83: Превышена максимальная длина (>79 символов)

| 83 -> # make_name prevents the use of rapydo default make_name function, that is only

./path/to/http-api-3.1//http-api-3.1/tests/base/test_cli.py: Строка 84: Превышена максимальная длина (>79 символов)

| 84 -> # working on endpoints context since it is based on tokens from flask.request

./path/to/http-api-3.1//http-api-3.1/tests/base/test_cli.py: Строка 83: Слишком длинный комментарий

```
make_name prevents the use of rapydo default make_name function, that is only
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_cli.py: Строка 84: Слишком длинный комментарий

```
working on endpoints context since it is based on tokens from flask.request
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_celery.py: Строка 25: Превышена максимальная длина (>79 символов)

```
| 25 -> CONNECTOR_AVAILABLE, reason=f"This test needs {CONNECTOR} to be not available"
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_celery.py: Строка 36: Превышена максимальная длина (>79 символов)

```
| 36 -> not CONNECTOR_AVAILABLE, reason=f"This test needs {CONNECTOR} to be available"
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_celery.py: Строка 59: Превышена максимальная длина (>79 символов)

```
| 59 -> project_title = get_project_configuration("project.title", default="YourProject")
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_celery.py: Строка 69: Превышена максимальная длина (>79 символов)

```
| 69 -> # fixed-id is a mocked value set in TESTING mode by @task in Celery connector
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_celery.py: Строка 88: Превышена максимальная длина (>79 символов)

```
| 88 -> # the errors decorator re-raise the Ignore exception, without any further action
```


./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_celery.py: Строка 103: Превышена максимальная длина (>79 символов)

```
| 103 -> assert f"Subject: {project_title}: Task test_task failed (failure #1)"
in headers
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_celery.py: Строка 105: Превышена максимальная длина (>79 символов)

```
| 105 -> # fixed-id is a mocked value set in TESTING mode by @task in Celery
connector
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_celery.py: Строка 127: Превышена максимальная длина (>79 символов)

```
| 127 -> assert f"Subject: {project_title}: Task test_task failed (failure #1)"
in headers
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_celery.py: Строка 129: Превышена максимальная длина (>79 символов)

```
| 129 -> # fixed-id is a mocked value set in TESTING mode by @task in Celery
connector
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_celery.py: Строка 136: Превышена максимальная длина (>79 символов)

```
| 136 -> exc = "MyException: Force the retry of this task by using a custom
exception"
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_celery.py: Строка 144: Превышена максимальная длина (>79 символов)

```
| 144 -> "Due to limitations on RABBIT backend task results will not be tested"
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_celery.py: Строка 155: Превышена

максимальная длина (>79 символов)

```
| 155 -> pytest.fail(f"Task timeout, result={task.result}, status={task.status}")
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_celery.py: Строка 220: Превышена максимальная длина (>79 символов)

```
| 220 -> obj.create_periodic_task(name="task1", task="task.does.not.exists",  
every="60")
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_celery.py: Строка 242: Превышена максимальная длина (>79 символов)

```
| 242 -> obj.create_periodic_task(name="task1", task="task.does.not.exists",  
every="60")
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_celery.py: Строка 291: Превышена максимальная длина (>79 символов)

```
| 291 -> name="task4", task="task.does.not.exists", every=60, period="seconds"
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_celery.py: Строка 296: Превышена максимальная длина (>79 символов)

```
| 296 -> name="task5", task="task.does.not.exists", every=60, period="minutes"
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_celery.py: Строка 310: Превышена максимальная длина (>79 символов)

```
| 310 -> with pytest.raises(BadRequest, match=r"Invalid timedelta period:  
years"):
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_celery.py: Переменная CONNECTOR: Некорректное именование

CONNECTOR

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_celery.py: Переменная
CONNECTOR_AVAILABLE: Некорректное именование
CONNECTOR_AVAILABLE

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_celery.py: Строка 69: Слишком длинный комментарий

```
fixed-id is a mocked value set in TESTING mode by @task in Celery connector
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_celery.py: Строка 88: Слишком длинный комментарий

```
the errors decorator re-raise the Ignore exception, without any further action
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_celery.py: Строка 105: Слишком длинный комментарий

```
fixed-id is a mocked value set in TESTING mode by @task in Celery connector
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_celery.py: Строка 129: Слишком длинный комментарий

```
fixed-id is a mocked value set in TESTING mode by @task in Celery connector
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_ftp.py: Строка 20: Превышена максимальная длина (>79 символов)

```
| 20 -> CONNECTOR_AVAILABLE, reason=f"This test needs {CONNECTOR} to be not available"
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_ftp.py: Строка 31: Превышена максимальная длина (>79 символов)

```
| 31 -> not CONNECTOR_AVAILABLE, reason=f"This test needs {CONNECTOR} to be available"
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_ftp.py: Строка 92: Превышена максимальная длина (>79 символов)

```
| 92 -> obj.connection.storlines(f"STOR {ftp_filename}", temporary_read_file)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_ftp.py: Строка 102: Превышена максимальная длина (>79 символов)

```
| 102 -> obj.connection.retrlines(f"RETR {ftp_filename}", download_handle.write)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_ftp.py: Строка 112: Превышена максимальная длина (>79 символов)

```
| 112 -> with pytest.raises(ServiceUnavailable, match=r"Invalid retry wait value: -1"):
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_ftp.py: Переменная CONNECTOR:
Некорректное именование
CONNECTOR

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_ftp.py: Переменная CONNECTOR_AVAILABLE:
Некорректное именование
CONNECTOR_AVAILABLE

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_ftp.py: Переменная MOCKED_RETURN:
Некорректное именование
MOCKED_RETURN

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_ftp.py: Переменная WAIT: Некорректное именование
WAIT

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_neo4j.py: Строка 25: Превышена максимальная длина (>79 символов)

```
| 25 -> CONNECTOR_AVAILABLE, reason=f"This test needs {CONNECTOR} to be not available"
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_neo4j.py: Строка 36: Превышена максимальная длина (>79 символов)

```
| 36 -> not CONNECTOR_AVAILABLE, reason=f"This test needs {CONNECTOR} to be available"
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_neo4j.py: Строка 61: Превышена максимальная длина (>79 символов)

```
| 61 -> with pytest.raises(AttributeError, match=r"Model InvalidModel not found"):
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_neo4j.py: Строка 80: Превышена максимальная длина (>79 символов)

```
| 80 -> with pytest.raises(CypherSyntaxError, match=r"Failed to execute Cypher Query"):
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_neo4j.py: Строка 141: Превышена максимальная длина (>79 символов)

```
| 141 -> with pytest.raises(ServiceUnavailable, match=r"Invalid retry value: 0"):
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_neo4j.py: Строка 143: Превышена максимальная длина (>79 символов)

```
| 143 -> with pytest.raises(ServiceUnavailable, match=r"Invalid retry value: -1"):
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_neo4j.py: Строка 145: Превышена

максимальная длина (>79 символов)

```
| 145 -> with pytest.raises(ServiceUnavailable, match=r"Invalid retry wait value: -1"):
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_neo4j.py: Строка 175: Превышена максимальная длина (>79 символов)

```
| 175 -> node1 = NodeDump("TestNode1", fields=["f1:string", "f2:int", "f3:float"])
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_neo4j.py: Строка 177: Превышена максимальная длина (>79 символов)

```
| 177 -> node2 = NodeDump("TestNode2", fields=["f1:string", "f2:int", "f3:float"])
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_neo4j.py: Строка 197: Превышена максимальная длина (>79 символов)

```
| 197 -> obj.cypher("CREATE INDEX testnode_index_name1 FOR (n:TestNode1) ON (n.f1)")
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_neo4j.py: Строка 198: Превышена максимальная длина (>79 символов)

```
| 198 -> obj.cypher("CREATE INDEX testnode_index_name2 FOR (n:TestNode2) ON (n.f1)")
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_neo4j.py: Строка 278: Превышена максимальная длина (>79 символов)

```
| 278 -> DataDump.delete_relationships("TestNode1", "MY_REL", "TestNode3", limit=1)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_neo4j.py: Строка 301: Превышена максимальная длина (>79 символов)

```
| 301 -> rel = RelationDump("T1", "R1", "T2", fields=["f1", "f1"],  
ignore_indexes=True)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test
_connector_neo4j.py: Переменная CONNECTOR:

Некорректное именован

CONNECTOR

./path/to/http-api-3.1//http-api-3.1/tests/base/test
_connector_neo4j.py: Переменная CONNECTOR_AVAILABLE:

Некорректное именован

CONNECTOR_AVAILABLE

./path/to/http-api-3.1//http-api-3.1/tests/base/test
_connector_neo4j.py: Переменная MOCKED_RETURN:

Некорректное именован

MOCKED_RETURN

./path/to/http-api-3.1//http-api-3.1/tests/base/test
_connector_neo4j.py: Переменная WAIT: Некорректное

именован

WAIT

./path/to/http-api-3.1//http-api-3.1/tests/base/test
_connector_rabbit.py: Строка 18: Превышена
максимальная длина (>79 символов)

```
| 18 -> CONNECTOR_AVAILABLE, reason=f"This test needs {CONNECTOR} to be not  
available"
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test
_connector_rabbit.py: Строка 29: Превышена
максимальная длина (>79 символов)

```
| 29 -> not CONNECTOR_AVAILABLE, reason=f"This test needs {CONNECTOR} to be  
available"
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test
_connector_rabbit.py: Строка 167: Превышена

максимальная длина (>79 символов)

```
| 167 -> with pytest.raises(ServiceUnavailable, match=r"Invalid retry wait value: -1"):
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_rabbit.py: Переменная CONNECTOR: Некорректное именование

CONNECTOR

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_rabbit.py: Переменная CONNECTOR_AVAILABLE: Некорректное именование

CONNECTOR_AVAILABLE

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_rabbit.py: Переменная MOCKED_RETURN: Некорректное именование

MOCKED_RETURN

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_rabbit.py: Переменная WAIT: Некорректное именование

WAIT

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_redis.py: Строка 17: Превышена максимальная длина (>79 символов)

```
| 17 -> CONNECTOR_AVAILABLE, reason=f"This test needs {CONNECTOR} to be not available"
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_redis.py: Строка 28: Превышена максимальная длина (>79 символов)

```
| 28 -> not CONNECTOR_AVAILABLE, reason=f"This test needs {CONNECTOR} to be available"
```


./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_redis.py: Строка 81: Превышена максимальная длина (>79 символов)

```
| 81 -> with pytest.raises(ServiceUnavailable, match=r"Invalid retry wait value: -1"):
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_redis.py: Переменная CONNECTOR: Некорректное именование

CONNECTOR

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_redis.py: Переменная CONNECTOR_AVAILABLE: Некорректное именование

CONNECTOR_AVAILABLE

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_redis.py: Переменная MOCKED_RETURN: Некорректное именование

MOCKED_RETURN

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_redis.py: Переменная WAIT: Некорректное именование

WAIT

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_smtp.py: Строка 20: Превышена максимальная длина (>79 символов)

```
| 20 -> CONNECTOR_AVAILABLE, reason=f"This test needs {CONNECTOR} to be not available"
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_smtp.py: Строка 31: Превышена максимальная длина (>79 символов)

```
| 31 -> not CONNECTOR_AVAILABLE, reason=f"This test needs {CONNECTOR} to be available"
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_smtp.py: Строка 63: Превышена максимальная длина (>79 символов)

```
| 63 -> assert obj.send("body", "subject", "to_addr", "from_addr", cc="test1",  
bcc="test2")
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_smtp.py: Строка 150: Превышена максимальная длина (>79 символов)

```
| 150 -> with pytest.raises(ServiceUnavailable, match=r"Invalid retry wait value:  
-1"):
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_smtp.py: Переменная CONNECTOR: Некорректное именование

CONNECTOR

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_smtp.py: Переменная CONNECTOR_AVAILABLE: Некорректное именование

CONNECTOR_AVAILABLE

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_smtp.py: Переменная MOCKED_RETURN: Некорректное именование

MOCKED_RETURN

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_smtp.py: Переменная WAIT: Некорректное именование

WAIT

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_sqlalchemy.py: Строка 17: Превышена максимальная длина (>79 символов)

```
| 17 -> CONNECTOR_AVAILABLE, reason=f"This test needs {CONNECTOR} to be not available"
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_sqlalchemy.py: Строка 28: Превышена максимальная длина (>79 символов)

```
| 28 -> not CONNECTOR_AVAILABLE, reason=f"This test needs {CONNECTOR} to be available"
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_sqlalchemy.py: Строка 87: Превышена максимальная длина (>79 символов)

```
| 87 -> with pytest.raises(ServiceUnavailable, match=r"Invalid retry wait value: -1"):
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_sqlalchemy.py: Переменная CONNECTOR: Некорректное именование

CONNECTOR

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_sqlalchemy.py: Переменная CONNECTOR_AVAILABLE: Некорректное именование

CONNECTOR_AVAILABLE

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_sqlalchemy.py: Переменная MOCKED_RETURN: Некорректное именование

MOCKED_RETURN

./path/to/http-api-3.1//http-api-3.1/tests/base/test_connector_sqlalchemy.py: Переменная WAIT: Некорректное именование

WAIT

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_groups.py: Строка 20: Превышена

максимальная длина (>79 символов)

```
| 20 -> if not Env.get_bool("MAIN_LOGIN_ENABLE") or not  
Env.get_bool("AUTH_ENABLE"):
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_groups.py: Строка 25: Превышена максимальная длина (>79 символов)

```
| 25 -> staff_role_enabled = Role.STAFF.value in [r.name for r in  
auth.get_roles()]
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_groups.py: Строка 32: Превышена максимальная длина (>79 символов)

```
| 32 -> "Skipping tests of admin/groups endpoints, role Staff not enabled"
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_groups.py: Строка 51: Превышена максимальная длина (>79 символов)

```
| 51 -> schema = self.get_dynamic_input_schema(client, "admin/groups", headers)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_groups.py: Строка 55: Превышена максимальная длина (>79 символов)

```
| 55 -> r = client.post(f"{API_URI}/admin/groups", json=data, headers=headers)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_groups.py: Строка 143: Превышена максимальная длина (>79 символов)

```
| 143 -> r = client.put(f"{API_URI}/admin/groups/xyz", json=data,  
headers=headers)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_groups.py: Строка 149: Превышена максимальная длина (>79 символов)

```
| 149 -> # match=rf"Cannot delete this group, it is assigned to {len(members)}  
user(s)",
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_groups.py: Строка 153: Превышена максимальная длина (>79 символов)

```
| 153 -> r = client.delete(f"{API_URI}/admin/groups/{uuid}", headers=headers)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_groups.py: Строка 178: Превышена максимальная длина (>79 символов)

```
| 178 -> # Very important: admin_groups must be tested before admin_users and profile
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_groups.py: Строка 210: Превышена максимальная длина (>79 символов)

```
| 210 -> # adding coordinator to enforce the role and use it for additional tests
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_groups.py: Строка 211: Превышена максимальная длина (>79 символов)

```
| 211 -> "roles": orjson.dumps([role, "group_coordinator"]).decode("UTF8"),
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_groups.py: Строка 219: Превышена максимальная длина (>79 символов)

```
| 219 -> f"{API_URI}/admin/users/{user_uuid}", json=data, headers=headers
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_groups.py: Строка 236: Превышена максимальная длина (>79 символов)

```
| 236 -> r = client.get(f"{API_URI}/admin/users/{user_uuid}", headers=headers)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_groups.py: Строка 149: Слишком длинный комментарий

```
match=r"Cannot delete this group, it is assigned to {len(members)} user(s)",
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_groups.py: Строка 178: Слишком длинный комментарий

```
Very important: admin_groups must be tested before admin_users and profile
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_stats.py: Строка 27: Превышена максимальная длина (>79 символов)

```
| 27 -> d = datetime.strptime(stats["system"]["boot_time"],
"%Y-%m-%dT%H:%M:%S.%fZ")
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_stats.py: Строка 70: Превышена максимальная длина (>79 символов)

```
| 70 -> stats["disk"]["total_disk_space"] =
float(stats["disk"]["total_disk_space"])
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_stats.py: Строка 71: Превышена максимальная длина (>79 символов)

```
| 71 -> stats["disk"]["free_disk_space"] =
float(stats["disk"]["free_disk_space"])
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_stats.py: Строка 72: Превышена максимальная длина (>79 символов)

```
| 72 -> stats["disk"]["used_disk_space"] =
float(stats["disk"]["used_disk_space"])
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_stats.py: Строка 78: Превышена максимальная длина (>79 символов)

```
| 78 -> assert stats["disk"]["free_disk_space"] <
stats["disk"]["total_disk_space"]
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_stats.py: Строка 81: Превышена

максимальная длина (>79 символов)

```
| 81 -> assert stats["disk"]["used_disk_space"] <
stats["disk"]["total_disk_space"]
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_stats.py: Строка 89: Превышена максимальная длина (>79 символов)

```
| 89 -> p = stats["disk"]["used_disk_space"] / stats["disk"]["total_disk_space"]
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_stats.py: Строка 108: Превышена максимальная длина (>79 символов)

```
| 108 -> stats["network_latency"]["min"] = float(stats["network_latency"]["min"])
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_stats.py: Строка 109: Превышена максимальная длина (>79 символов)

```
| 109 -> stats["network_latency"]["avg"] = float(stats["network_latency"]["avg"])
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_stats.py: Строка 110: Превышена максимальная длина (>79 символов)

```
| 110 -> stats["network_latency"]["max"] = float(stats["network_latency"]["max"])
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_stats.py: Строка 113: Превышена максимальная длина (>79 символов)

```
| 113 -> assert stats["network_latency"]["min"] <=
stats["network_latency"]["avg"]
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_stats.py: Строка 114: Превышена максимальная длина (>79 символов)

```
| 114 -> assert stats["network_latency"]["min"] <=
stats["network_latency"]["max"]
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_stats.py: Строка 117: Превышена максимальная длина (>79 символов)

```
| 117 -> assert stats["network_latency"]["avg"] >=
stats["network_latency"]["min"]
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_stats.py: Строка 118: Превышена максимальная длина (>79 символов)

```
| 118 -> assert stats["network_latency"]["avg"] <=
stats["network_latency"]["max"]
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_stats.py: Строка 121: Превышена максимальная длина (>79 символов)

```
| 121 -> assert stats["network_latency"]["max"] >=
stats["network_latency"]["min"]
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_stats.py: Строка 122: Превышена максимальная длина (>79 символов)

```
| 122 -> assert stats["network_latency"]["max"] >=
stats["network_latency"]["avg"]
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_users.py: Строка 15: Превышена максимальная длина (>79 символов)

```
| 15 -> if not Env.get_bool("MAIN_LOGIN_ENABLE") or not
Env.get_bool("AUTH_ENABLE"):
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_users.py: Строка 19: Превышена максимальная длина (>79 символов)

```
| 19 -> project_tile = get_project_configuration("project.title",
default="YourProject")
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_users.py: Строка 22: Превышена

максимальная длина (>79 символов)

```
| 22 -> staff_role_enabled = Role.STAFF.value in [r.name for r in  
auth.get_roles()]
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_users.py: Строка 30: Превышена максимальная длина (>79 символов)

```
| 30 -> "Skipping tests of admin/users endpoints, role Staff not enabled"
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_users.py: Строка 48: Превышена максимальная длина (>79 символов)

```
| 48 -> schema = self.get_dynamic_input_schema(client, "admin/users", headers)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_users.py: Строка 56: Превышена максимальная длина (>79 символов)

```
| 56 -> r = client.post(f"{API_URI}/admin/users", json=data, headers=headers)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_users.py: Строка 128: Превышена максимальная длина (>79 символов)

```
| 128 -> assert users_list.get("email") == data.get("email", "MISSING").lower()
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_users.py: Строка 140: Превышена максимальная длина (>79 символов)

```
| 140 -> r = client.post(f"{API_URI}/admin/users", json=data, headers=headers)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_users.py: Строка 148: Превышена максимальная длина (>79 символов)

```
| 148 -> r = client.post(f"{API_URI}/admin/users", json=data, headers=headers)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_users.py: Строка 152: Превышена

максимальная длина (>79 символов)

```
| 152 -> == f"A User already exists with email:  
{BaseAuthentication.default_user}"
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_users.py: Строка 162: Превышена максимальная длина (>79 символов)

```
| 162 -> r = client.post(f"{API_URI}/admin/users", json=data2, headers=headers)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_users.py: Строка 224: Превышена максимальная длина (>79 символов)

```
| 224 -> f"{API_URI}/admin/users/{uuid2}", json=new_data, headers=headers
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_users.py: Строка 238: Превышена максимальная длина (>79 символов)

```
| 238 -> assert users_list.get("email") != data.get("email", "MISSING").lower()
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_users.py: Строка 239: Превышена максимальная длина (>79 символов)

```
| 239 -> assert users_list.get("email") == data2.get("email", "MISSING").lower()
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_users.py: Строка 250: Превышена максимальная длина (>79 символов)

```
| 250 -> r = client.delete(f"{API_URI}/admin/users/invalid", headers=headers)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_users.py: Строка 273: Превышена максимальная длина (>79 символов)

```
| 273 -> r = client.put(f"{API_URI}/admin/users/{uuid2}", json=data,  
headers=headers)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_users.py: Строка 298: Превышена максимальная длина (>79 символов)

```
| 298 -> assert data2.get("email", "MISSING").lower() in mail.get("body", "")
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_users.py: Строка 299: Превышена максимальная длина (>79 символов)

```
| 299 -> assert newpwd in mail.get("body", "") or escape(newpwd) in mail.get(
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_users.py: Строка 313: Превышена максимальная длина (>79 символов)

```
| 313 -> r = client.post(f"{API_URI}/admin/users", json=data, headers=headers2)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_users.py: Строка 323: Превышена максимальная длина (>79 символов)

```
| 323 -> r = client.delete(f"{API_URI}/admin/users/{uuid}", headers=headers2)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_users.py: Строка 335: Превышена максимальная длина (>79 символов)

```
| 335 -> r = client.delete(f"{API_URI}/admin/users/{uuid2}", headers=headers)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_users.py: Строка 347: Превышена максимальная длина (>79 символов)

```
| 347 -> # Restore the default password (changed due to  
FORCE_FIRST_PASSWORD_CHANGE)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_users.py: Строка 361: Превышена максимальная длина (>79 символов)

```
| 361 -> r = client.put(f"{API_URI}/admin/users/{uuid}", json=data,  
headers=headers)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_users.py: Строка 385: Превышена максимальная длина (>79 символов)

```
| 385 -> def test_staff_restrictions(self, client: FlaskClient, faker: Faker) ->
None:
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_users.py: Строка 386: Превышена максимальная длина (>79 символов)

```
| 386 -> if not Env.get_bool("MAIN_LOGIN_ENABLE") or not
Env.get_bool("AUTH_ENABLE"):
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_users.py: Строка 391: Превышена максимальная длина (>79 символов)

```
| 391 -> staff_role_enabled = Role.STAFF.value in [r.name for r in
auth.get_roles()]
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_users.py: Строка 395: Превышена максимальная длина (>79 символов)

```
| 395 -> "Skipping tests of admin/users restrictions, role Staff not enabled"
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_users.py: Строка 415: Превышена максимальная длина (>79 символов)

```
| 415 -> r = client.get(f"{API_URI}/admin/users/{user_uuid}",
headers=admin_headers)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_users.py: Строка 418: Превышена максимальная длина (>79 символов)

```
| 418 -> r = client.get(f"{API_URI}/admin/users/{staff_uuid}",
headers=admin_headers)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_users.py: Строка 421: Превышена

максимальная длина (>79 символов)

```
| 421 -> r = client.get(f"{API_URI}/admin/users/{admin_uuid}",  
headers=admin_headers)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_users.py: Строка 424: Превышена максимальная длина (>79 символов)

```
| 424 -> r = client.get(f"{API_URI}/admin/users/{user_uuid}",  
headers=staff_headers)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_users.py: Строка 427: Превышена максимальная длина (>79 символов)

```
| 427 -> r = client.get(f"{API_URI}/admin/users/{staff_uuid}",  
headers=staff_headers)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_users.py: Строка 430: Превышена максимальная длина (>79 символов)

```
| 430 -> r = client.get(f"{API_URI}/admin/users/{admin_uuid}",  
headers=staff_headers)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_users.py: Строка 469: Превышена максимальная длина (>79 символов)

```
| 469 -> tmp_schema = self.get_dynamic_input_schema(client, "admin/users",  
admin_headers)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_users.py: Строка 486: Превышена максимальная длина (>79 символов)

```
| 486 -> tmp_schema = self.get_dynamic_input_schema(client, "admin/users",  
staff_headers)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_users.py: Строка 515: Превышена максимальная длина (>79 символов)

```
| 515 -> schema = self.get_dynamic_input_schema(client, "admin/users",  
staff_headers)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_users.py: Строка 523: Превышена максимальная длина (>79 символов)

```
| 523 -> r = client.post(f"{API_URI}/admin/users", json=data,  
headers=staff_headers)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_users.py: Строка 546: Превышена максимальная длина (>79 символов)

```
| 546 -> r = client.delete(f"{API_URI}/admin/users/{admin_uuid}",  
headers=staff_headers)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_users.py: Строка 551: Превышена максимальная длина (>79 символов)

```
| 551 -> r = client.delete(f"{API_URI}/admin/users/{user_uuid}",  
headers=staff_headers)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_admin_users.py: Строка 347: Слишком длинный комментарий

```
Restore the default password (changed due to FORCE_FIRST_PASSWORD_CHANGE)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authentication.py: Строка 24: Превышена максимальная длина (>79 символов)

```
| 24 -> f"{API_URI}/tests/noauth", headers={"Authorization": "Bearer invalid"}
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authentication.py: Строка 73: Превышена максимальная длина (>79 символов)

```
| 73 -> f"{API_URI}/tests/authentication", query_string={"access_token": token}
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authentication.py: Строка 89: Превышена максимальная длина (>79 символов)

```
| 89 -> r = client.get(f"{API_URI}/tests/optionalauthentication",  
headers=headers)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authentication.py: Строка 97: Превышена максимальная длина (>79 символов)

```
| 97 -> # But not invalid tokens, i.e. if presented the tokens is always validated
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authentication.py: Строка 145: Превышена максимальная длина (>79 символов)

```
| 145 -> f"{API_URI}/tests/queryauthentication", query_string={"access_token":  
token}
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authentication.py: Строка 160: Превышена максимальная длина (>79 символов)

```
| 160 -> def test_optional_access_token_parameter(self, client: FlaskClient) ->  
None:
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authentication.py: Строка 172: Превышена максимальная длина (>79 символов)

```
| 172 -> r = client.get(f"{API_URI}/tests/optionalqueryauthentication",  
headers=headers)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authentication.py: Строка 180: Превышена максимальная длина (>79 символов)

```
| 180 -> # But not invalid tokens, i.e. if presented the tokens is always  
validated
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authentication.py: Строка 205: Превышена

максимальная длина (>79 символов)

```
| 205 -> def test_authentication_with_multiple_roles(self, client: FlaskClient)
-> None:
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authentication.py: Строка 250: Превышена максимальная длина (>79 символов)

```
| 250 -> f"{API_URI}/tests/unknownroleauthentication", headers=user_header
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authentication.py: Строка 256: Превышена максимальная длина (>79 символов)

```
| 256 -> def test_authentication_with_auth_callback(self, client: FlaskClient) ->
None:
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authentication.py: Строка 273: Превышена максимальная длина (>79 символов)

```
| 273 -> f"{API_URI}{VALID}", query_string={"test": True}, headers=admin_headers
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authentication.py: Строка 283: Превышена максимальная длина (>79 символов)

```
| 283 -> f"{API_URI}{INVALID}", query_string={"test": True},
headers=admin_headers
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authentication.py: Переменная VALID:
Некорректное именованние

VALID

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authentication.py: Переменная INVALID:
Некорректное именованние

INVALID

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 41: Превышена максимальная длина (>79 символов)

```
| 41 -> # Test a single endpoint, remove the path from the list and return the new list
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 93: Превышена максимальная длина (>79 символов)

```
| 93 -> # List of all paths to be tested. After each test a path will be removed.
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 98: Превышена максимальная длина (>79 символов)

```
| 98 -> headers, _ = self.do_login(client, data.get("email"), data.get("password"))
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 101: Превышена максимальная длина (>79 символов)

```
| 101 -> paths = self.check_endpoint(client, "GET", "/api/status", headers, True, paths)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 102: Превышена максимальная длина (>79 символов)

```
| 102 -> paths = self.check_endpoint(client, "GET", "/api/specs", headers, True, paths)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 103: Превышена максимальная длина (>79 символов)

```
| 103 -> paths = self.check_endpoint(client, "POST", "/auth/login", headers, True, paths)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 106: Превышена

максимальная длина (>79 символов)

```
| 106 -> client, "POST", "/auth/login/unlock/", headers, True, paths
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 116: Превышена максимальная длина (>79 символов)

```
| 116 -> client, "PUT", "/auth/profile/activate/", headers, True, paths
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 119: Превышена максимальная длина (>79 символов)

```
| 119 -> if Env.get_bool("ALLOW_PASSWORD_RESET") and  
Connector.check_availability(
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 130: Превышена максимальная длина (>79 символов)

```
| 130 -> paths = self.check_endpoint(client, "GET", "/auth/status", headers,  
True, paths)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 140: Превышена максимальная длина (>79 символов)

```
| 140 -> paths = self.check_endpoint(client, "GET", "/auth/tokens", headers,  
True, paths)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 167: Превышена максимальная длина (>79 символов)

```
| 167 -> client, "DELETE", "/api/admin/users/", headers, True, paths
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 179: Превышена максимальная длина (>79 символов)

```
| 179 -> client, "DELETE", "/api/admin/groups/", headers, True, paths
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 198: Превышена максимальная длина (>79 символов)

```
| 198 -> paths = self.check_endpoint(client, "GET", "/auth/logout", headers, True, paths)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 215: Превышена максимальная длина (>79 символов)

```
| 215 -> log.warning("Skipping authorization tests on role Staff (not enabled)")
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 218: Превышена максимальная длина (>79 символов)

```
| 218 -> # List of all paths to be tested. After each test a path will be removed.
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 223: Превышена максимальная длина (>79 символов)

```
| 223 -> headers, _ = self.do_login(client, data.get("email"), data.get("password"))
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 226: Превышена максимальная длина (>79 символов)

```
| 226 -> paths = self.check_endpoint(client, "GET", "/api/status", headers, True, paths)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 227: Превышена максимальная длина (>79 символов)

```
| 227 -> paths = self.check_endpoint(client, "GET", "/api/specs", headers, True, paths)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 228: Превышена максимальная длина (>79 символов)

```
| 228 -> paths = self.check_endpoint(client, "POST", "/auth/login", headers, True, paths)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 231: Превышена максимальная длина (>79 символов)

```
| 231 -> client, "POST", "/auth/login/unlock/", headers, True, paths
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 241: Превышена максимальная длина (>79 символов)

```
| 241 -> client, "PUT", "/auth/profile/activate/", headers, True, paths
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 244: Превышена максимальная длина (>79 символов)

```
| 244 -> if Env.get_bool("ALLOW_PASSWORD_RESET") and Connector.check_availability(
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 255: Превышена максимальная длина (>79 символов)

```
| 255 -> paths = self.check_endpoint(client, "GET", "/auth/status", headers, True, paths)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 265: Превышена максимальная длина (>79 символов)

```
| 265 -> paths = self.check_endpoint(client, "GET", "/auth/tokens", headers, True, paths)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 292: Превышена максимальная длина (>79 символов)

```
| 292 -> client, "DELETE", "/api/admin/users/", headers, False, paths
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 304: Превышена максимальная длина (>79 символов)

```
| 304 -> client, "DELETE", "/api/admin/groups/", headers, True, paths
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 313: Превышена максимальная длина (>79 символов)

```
| 313 -> client, "DELETE", "/api/admin/tokens/", headers, False, paths
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 323: Превышена максимальная длина (>79 символов)

```
| 323 -> paths = self.check_endpoint(client, "GET", "/auth/logout", headers, True, paths)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 334: Превышена максимальная длина (>79 символов)

```
| 334 -> # List of all paths to be tested. After each test a path will be removed.
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 339: Превышена максимальная длина (>79 символов)

```
| 339 -> headers, _ = self.do_login(client, data.get("email"), data.get("password"))
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 342: Превышена максимальная длина (>79 символов)

```
| 342 -> paths = self.check_endpoint(client, "GET", "/api/status", headers, True, paths)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 343: Превышена максимальная длина (>79 символов)

```
| 343 -> paths = self.check_endpoint(client, "GET", "/api/specs", headers, True, paths)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 344: Превышена максимальная длина (>79 символов)

```
| 344 -> paths = self.check_endpoint(client, "POST", "/auth/login", headers, True, paths)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 347: Превышена максимальная длина (>79 символов)

```
| 347 -> client, "POST", "/auth/login/unlock/", headers, True, paths
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 357: Превышена максимальная длина (>79 символов)

```
| 357 -> client, "PUT", "/auth/profile/activate/", headers, True, paths
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 360: Превышена максимальная длина (>79 символов)

```
| 360 -> if Env.get_bool("ALLOW_PASSWORD_RESET") and Connector.check_availability(
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 371: Превышена максимальная длина (>79 символов)

```
| 371 -> paths = self.check_endpoint(client, "GET", "/auth/status", headers, True, paths)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 381: Превышена максимальная длина (>79 символов)

```
| 381 -> paths = self.check_endpoint(client, "GET", "/auth/tokens", headers, True, paths)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 408: Превышена максимальная длина (>79 символов)

```
| 408 -> client, "DELETE", "/api/admin/users/", headers, False, paths
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 417: Превышена максимальная длина (>79 символов)

```
| 417 -> client, "PUT", "/api/admin/groups/", headers, False, paths
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 420: Превышена максимальная длина (>79 символов)

```
| 420 -> client, "DELETE", "/api/admin/groups/", headers, False, paths
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 429: Превышена максимальная длина (>79 символов)

```
| 429 -> client, "DELETE", "/api/admin/tokens/", headers, False, paths
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 439: Превышена максимальная длина (>79 символов)

```
| 439 -> paths = self.check_endpoint(client, "GET", "/auth/logout", headers, True, paths)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 450: Превышена максимальная длина (>79 символов)

```
| 450 -> # List of all paths to be tested. After each test a path will be removed.
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 455: Превышена максимальная длина (>79 символов)

```
| 455 -> headers, _ = self.do_login(client, data.get("email"), data.get("password"))
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 458: Превышена максимальная длина (>79 символов)

```
| 458 -> paths = self.check_endpoint(client, "GET", "/api/status", headers, True, paths)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 459: Превышена максимальная длина (>79 символов)

```
| 459 -> paths = self.check_endpoint(client, "GET", "/api/specs", headers, True, paths)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 460: Превышена максимальная длина (>79 символов)

```
| 460 -> paths = self.check_endpoint(client, "POST", "/auth/login", headers, True, paths)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 463: Превышена максимальная длина (>79 символов)

```
| 463 -> client, "POST", "/auth/login/unlock/", headers, True, paths
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 473: Превышена максимальная длина (>79 символов)

```
| 473 -> client, "PUT", "/auth/profile/activate/", headers, True, paths
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 476: Превышена максимальная длина (>79 символов)

```
| 476 -> if Env.get_bool("ALLOW_PASSWORD_RESET") and Connector.check_availability(
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 487: Превышена максимальная длина (>79 символов)


```
| 487 -> paths = self.check_endpoint(client, "GET", "/auth/status", headers, True, paths)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 497: Превышена максимальная длина (>79 символов)

```
| 497 -> paths = self.check_endpoint(client, "GET", "/auth/tokens", headers, True, paths)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 524: Превышена максимальная длина (>79 символов)

```
| 524 -> client, "DELETE", "/api/admin/users/", headers, False, paths
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 533: Превышена максимальная длина (>79 символов)

```
| 533 -> client, "PUT", "/api/admin/groups/", headers, False, paths
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 536: Превышена максимальная длина (>79 символов)

```
| 536 -> client, "DELETE", "/api/admin/groups/", headers, False, paths
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 545: Превышена максимальная длина (>79 символов)

```
| 545 -> client, "DELETE", "/api/admin/tokens/", headers, False, paths
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 555: Превышена максимальная длина (>79 символов)

```
| 555 -> paths = self.check_endpoint(client, "GET", "/auth/logout", headers, True, paths)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 562: Превышена

максимальная длина (>79 символов)

```
| 562 -> # List of all paths to be tested. After each test a path will be removed.
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 567: Превышена максимальная длина (>79 символов)

```
| 567 -> paths = self.check_endpoint(client, "GET", "/api/status", headers, True, paths)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 568: Превышена максимальная длина (>79 символов)

```
| 568 -> paths = self.check_endpoint(client, "GET", "/api/specs", headers, True, paths)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 576: Превышена максимальная длина (>79 символов)

```
| 576 -> paths = self.check_endpoint(client, "POST", "/auth/login", headers, True, paths)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 580: Превышена максимальная длина (>79 символов)

```
| 580 -> client, "POST", "/auth/login/unlock/", headers, True, paths
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 590: Превышена максимальная длина (>79 символов)

```
| 590 -> client, "PUT", "/auth/profile/activate/", headers, True, paths
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 593: Превышена максимальная длина (>79 символов)

```
| 593 -> if Env.get_bool("ALLOW_PASSWORD_RESET") and Connector.check_availability(
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 645: Превышена максимальная длина (>79 символов)

```
| 645 -> client, "DELETE", "/api/admin/users/", headers, False, paths
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 654: Превышена максимальная длина (>79 символов)

```
| 654 -> client, "PUT", "/api/admin/groups/", headers, False, paths
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 657: Превышена максимальная длина (>79 символов)

```
| 657 -> client, "DELETE", "/api/admin/groups/", headers, False, paths
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 666: Превышена максимальная длина (>79 символов)

```
| 666 -> client, "DELETE", "/api/admin/tokens/", headers, False, paths
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_authorizations.py: Строка 41: Слишком длинный комментарий

```
Test a single endpoint, remove the path from the list and return the new list
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_autocomplete.py: Строка 11: Превышена максимальная длина (>79 символов)

```
| 11 -> # valid inputs for endpoints with inputs defined by marshmallow schemas
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_autocomplete.py: Строка 12: Превышена максимальная длина (>79 символов)

```
| 12 -> schema = self.get_dynamic_input_schema(client, "tests/autocomplete", {})
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_autocomplete.py: Строка 25: Превышена максимальная длина (>79 символов)

```
| 25 -> autocomplete_endpoint =  
f"{SERVER_URI}{schema[0]['autocomplete_endpoint']}"
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_autocomplete.py: Строка 57: Превышена максимальная длина (>79 символов)

```
| 57 -> r = client.put(f"{API_URI}/tests/autocomplete", json={"element":  
data[0]})
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_caching.py: Строка 38: Превышена максимальная длина (>79 символов)

```
| 38 -> # Since the caching is rounded to the second, few milliseconds cann make  
the
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_caching.py: Строка 41: Превышена максимальная длина (>79 символов)

```
| 41 -> # A second request at 00:00:01.002 is no longer cached, even if only 5  
millisec
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_caching.py: Строка 43: Превышена максимальная длина (>79 символов)

```
| 43 -> # Added 0.01 just to avoid to exactly start at the beginning of the second
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_caching.py: Строка 73: Превышена максимальная длина (>79 символов)

```
| 73 -> # Endpoint is unauthenticated, headers are ignored when building the  
cache key
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_caching.py: Строка 80: Превышена

максимальная длина (>79 символов)

```
| 80 -> f"{API_URI}/tests/cache/short", headers={"Authorization": "Bearer invalid"}
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_caching.py: Строка 128: Превышена максимальная длина (>79 символов)

```
| 128 -> # Endpoint is unauthenticated, headers are ignored when building the cache key
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_caching.py: Строка 135: Превышена максимальная длина (>79 символов)

```
| 135 -> f"{API_URI}/tests/cache/long", headers={"Authorization": "Bearer invalid"}
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_caching.py: Строка 159: Превышена максимальная длина (>79 символов)

```
| 159 -> f"{API_URI}/tests/cache/auth", headers={"Authorization": "Bearer invalid"}
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_caching.py: Строка 185: Превышена максимальная длина (>79 символов)

```
| 185 -> headers3, _ = self.do_login(client, data["email"], data["password"])
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_caching.py: Строка 196: Превышена максимальная длина (>79 символов)

```
| 196 -> # The counter changed, because the response is not replied from the cache
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_caching.py: Строка 206: Превышена максимальная длина (>79 символов)

```
| 206 -> # Same counter as above, because the response is replied from the cache
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_caching.py: Строка 211: Превышена максимальная длина (>79 символов)

```
| 211 -> def test_cached_semiauthenticated_endpoint(self, client: FlaskClient) -> None:
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_caching.py: Строка 234: Превышена максимальная длина (>79 символов)

```
| 234 -> # The counter changed, because the response is not replied from the cache
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_caching.py: Строка 262: Превышена максимальная длина (>79 символов)

```
| 262 -> def test_cached_authenticated_param_endpoint(self, client: FlaskClient) -> None:
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_caching.py: Строка 284: Превышена максимальная длина (>79 символов)

```
| 284 -> f"{API_URI}/tests/cache/paramauth", query_string={"access_token": token2}
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_caching.py: Строка 291: Превышена максимальная длина (>79 символов)

```
| 291 -> # but counter changed, because the response is not replied from the cache
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_caching.py: Строка 296: Превышена максимальная длина (>79 символов)

```
| 296 -> f"{API_URI}/tests/cache/paramauth", query_string={"access_token": token2}
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_caching.py: Строка 317: Превышена максимальная длина (>79 символов)

```
| 317 -> def test_cached_semiauthenticated_param_endpoint(self, client:
FlaskClient) -> None:
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_caching.py: Строка 336: Превышена максимальная длина (>79 символов)

```
| 336 -> r = client.get(f"{API_URI}/tests/cache/optionalparamauth",
headers=headers1)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_caching.py: Строка 341: Превышена максимальная длина (>79 символов)

```
| 341 -> # The counter changed, because the response is not replied from the cache
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_caching.py: Строка 344: Превышена максимальная длина (>79 символов)

```
| 344 -> r = client.get(f"{API_URI}/tests/cache/optionalparamauth",
headers=headers1)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_caching.py: Строка 353: Превышена максимальная длина (>79 символов)

```
| 353 -> # Response is reply from the cache even if the token is provided as query
param
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_caching.py: Строка 397: Превышена максимальная длина (>79 символов)

```
| 397 -> r = client.get(f"{API_URI}/tests/cache/optionalparamauth",
headers=headers2)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_caching.py: Переменная UUID: Некорректное именован

UUID

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_caching.py: Переменная COUNTER: Некорректное именование

COUNTER

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_caching.py: Переменная CONNECTOR_AVAILABLE: Некорректное именование

CONNECTOR_AVAILABLE

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_caching.py: Строка 38: Слишком длинный комментарий

Since the caching is rounded to the second, few milliseconds cann make the

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_caching.py: Строка 41: Слишком длинный комментарий

A second request at 00:00:01.002 is no longer cached, even if only 5 millisec

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_caching.py: Строка 73: Слишком длинный комментарий

Endpoint is unauthenticated, headers are ignored when building the cache key

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_caching.py: Строка 128: Слишком длинный комментарий

Endpoint is unauthenticated, headers are ignored when building the cache key

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_caching.py: Строка 353: Слишком длинный комментарий

Response is reply from the cache even if the token is provided as query param

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_db_exceptions.py: Строка 11: Превышена максимальная длина (>79 символов)

```
| 11 -> def test_database_exceptions(self, client: FlaskClient, faker: Faker) ->
None:
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_db_exceptions.py: Строка 16: Превышена максимальная длина (>79 символов)

```
| 16 -> # This is a special value. The endpoint will try to create a group without
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_db_exceptions.py: Строка 17: Превышена максимальная длина (>79 символов)

```
| 17 -> # shortname. A BadRequest is expected because the database should refuse
the
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_db_exceptions.py: Строка 22: Превышена максимальная длина (>79 символов)

```
| 22 -> assert self.get_content(r) == "Missing property shortname required by
Group"
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_db_exceptions.py: Строка 48: Превышена максимальная длина (>79 символов)

```
| 48 -> # This will try to create again a group with short/full name ==
random_name
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_db_exceptions.py: Строка 17: Слишком длинный комментарий

```
shortname. A BadRequest is expected because the database should refuse the
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_db_exceptions.py: Строка 48: Слишком длинный комментарий

```
This will try to create again a group with short/full name == random_name
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_encoding.py: Строка 56: Превышена максимальная длина (>79 символов)

```
| 56 -> # A string with len 1000 is about 1 kb, still too small to enable gzip encoding
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_encoding.py: Строка 56: Слишком длинный комментарий

```
A string with len 1000 is about 1 kb, still too small to enable gzip encoding
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_group_users.py: Строка 11: Превышена максимальная длина (>79 символов)

```
| 11 -> if not Env.get_bool("MAIN_LOGIN_ENABLE") or not Env.get_bool("AUTH_ENABLE"):
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_group_users.py: Строка 32: Превышена максимальная длина (>79 символов)

```
| 32 -> headers, _ = self.do_login(client, user1_data["email"], user1_data["password"])
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_group_users.py: Строка 46: Превышена максимальная длина (>79 символов)

```
| 46 -> r = client.delete(f"{API_URI}/group/users/{group1_uuid}", headers=headers)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_group_users.py: Строка 82: Превышена максимальная длина (>79 символов)

```
| 82 -> headers, _ = self.do_login(client, user3_data["email"], user3_data["password"])
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_group_users.py: Строка 112: Превышена максимальная длина (>79 символов)

```
| 112 -> client, roles=[Role.ADMIN, Role.COORDINATOR], data={"group":  
group1_uuid}
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_group_users.py: Строка 115: Превышена максимальная длина (>79 символов)

```
| 115 -> # Verify as Admin AND Coordinator (Expected: all members, including  
admins)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_group_users.py: Строка 116: Превышена максимальная длина (>79 символов)

```
| 116 -> headers, _ = self.do_login(client, user4_data["email"],  
user4_data["password"])
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_group_users.py: Строка 131: Превышена максимальная длина (>79 символов)

```
| 131 -> headers, _ = self.do_login(client, user1_data["email"],  
user1_data["password"])
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_group_users.py: Строка 115: Слишком длинный комментарий

```
Verify as Admin AND Coordinator (Expected: all members, including admins)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_inputs.py: Строка 12: Превышена максимальная длина (>79 символов)

```
| 12 -> # valid inputs for endpoints with inputs defined by marshmallow schemas
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_inputs.py: Строка 35: Превышена максимальная длина (>79 символов)

```
| 35 -> # This is the default case: both label and description are not explicitly set
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_inputs.py: Строка 72: Превышена максимальная длина (>79 символов)

```
| 72 -> # Here an explicit label is defined but not a description, so is == to the label
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_inputs.py: Строка 92: Превышена максимальная длина (>79 символов)

```
| 92 -> assert field["description"] == "This field accepts values in a defined range"
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_inputs.py: Строка 183: Превышена максимальная длина (>79 символов)

```
| 183 -> # The list is defined as List(CustomInt) and CustomInt is resolved as int
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_inputs.py: Строка 192: Превышена максимальная длина (>79 символов)

```
| 192 -> # The type is key[] ... should be something more explicative like FieldName[]
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_inputs.py: Строка 201: Превышена максимальная длина (>79 символов)

```
| 201 -> # mylist3 is a list of custom field, buildData can't automatically set a value
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_inputs.py: Строка 208: Превышена максимальная длина (>79 символов)

```
| 208 -> # This is to verify that access_token, if provided is excluded from parameters
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_inputs.py: Строка 228: Превышена максимальная длина (>79 символов)

```
| 228 -> schema = self.get_dynamic_input_schema(client, "tests/neo4jinputs", headers)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_inputs.py: Строка 233: Превышена максимальная длина (>79 символов)

```
| 233 -> # This is because the Neo4jChoice field is not completed for deserialization
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_inputs.py: Строка 234: Превышена максимальная длина (>79 символов)

```
| 234 -> # It is should be automatically translated into a select, with options by
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_inputs.py: Строка 239: Превышена максимальная длина (>79 символов)

```
| 239 -> f"{API_URI}/tests/neo4jinputs", json={"choice": "A"}, headers=headers
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_inputs.py: Строка 265: Превышена максимальная длина (>79 символов)

```
| 265 -> f"{API_URI}/tests/neo4jinputs", json={"choice": "B"}, headers=headers
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_inputs.py: Строка 277: Превышена максимальная длина (>79 символов)

```
| 277 -> f"{API_URI}/tests/neo4jinputs", json={"choice": "C"}, headers=headers
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_inputs.py: Строка 289: Превышена максимальная длина (>79 символов)

```
| 289 -> f"{API_URI}/tests/neo4jinputs", json={"choice": "D"}, headers=headers
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_inputs.py: Строка 35: Слишком длинный комментарий

This is the default case: both label and description are not explicitly set

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_inputs.py: Строка 72: Слишком длинный комментарий

Here an explicit label is defined but not a description, so is == to the label

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_inputs.py: Строка 192: Слишком длинный комментарий

The type is key[] ... should be something more explicative like fieldName[]

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_inputs.py: Строка 201: Слишком длинный комментарий

mylist3 is a list of custom field, buildData can't automatically set a value

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_inputs.py: Строка 208: Слишком длинный комментарий

This is to verify that access_token, if provided is excluded from parameters

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_inputs.py: Строка 233: Слишком длинный комментарий

This is because the Neo4jChoice field is not completed for deserialization

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_ban.py: Строка 30: Превышена максимальная длина (>79 символов)

| 30 -> # Login attempts are not registered, let's try to fail the login many times

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_ban.py: Строка 73: Превышена максимальная длина (>79 символов)

```
| 73 -> assert f"Subject: {project_tile}: {title}" in mail.get("headers", "")
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_ban.py: Строка 75: Превышена максимальная длина (>79 символов)

```
| 75 -> # assert "this email is to inform you that your credentials have been "
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_ban.py: Строка 77: Превышена максимальная длина (>79 символов)

```
| 77 -> # assert "inspect the list below to detect any unwanted login" in body
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_ban.py: Строка 78: Превышена максимальная длина (>79 символов)

```
| 78 -> # assert "Your credentials will be automatically unlocked in" in body
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_ban.py: Строка 148: Превышена максимальная длина (>79 символов)

```
| 148 -> self.do_login(client, data["email"], data["password"], status_code=403)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_ban.py: Строка 214: Превышена максимальная длина (>79 символов)

```
| 214 -> # Verify that unlock tokens can't be used if the user is already unlocked
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_ban.py: Строка 325: Превышена максимальная длина (>79 символов)

```
| 325 -> registration_data["password_confirm"] = registration_data["password"]
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_ban.py: Строка 327: Превышена максимальная длина (>79 символов)

```
| 327 -> # now the user is created but INACTIVE, activation endpoint is needed
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_ban.py: Строка 329: Превышена максимальная длина (>79 символов)

```
| 329 -> registration_message = "We are sending an email to your email address "
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_ban.py: Строка 330: Превышена максимальная длина (>79 символов)

```
| 330 -> registration_message += "where you will find the link to activate "
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_ban.py: Строка 334: Превышена максимальная длина (>79 символов)

```
| 334 -> # Registration endpoint send 2 mail: the first is the activation link,
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_ban.py: Строка 357: Превышена максимальная длина (>79 символов)

```
| 357 -> assert events[0].payload["username"] == registration_data["email"]
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_ban.py: Строка 364: Превышена максимальная длина (>79 символов)

```
| 364 -> # Event if non activated if password is wrong the status is 401
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_ban.py: Строка 374: Превышена максимальная длина (>79 символов)

```
| 374 -> assert events[0].payload["username"] == registration_data["email"]
```


./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_ban.py: Строка 388: Превышена максимальная длина (>79 символов)

```
| 388 -> assert events[0].payload["username"] == registration_data["email"]
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_ban.py: Строка 405: Превышена максимальная длина (>79 символов)

```
| 405 -> assert events[0].payload["username"] == registration_data["email"]
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_ban.py: Строка 425: Превышена максимальная длина (>79 символов)

```
| 425 -> # Verify that login still works (TOTP will be automatically added)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_ban.py: Строка 426: Превышена максимальная длина (>79 символов)

```
| 426 -> headers, _ = self.do_login(client, data["email"], data["password"])
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_ban.py: Строка 494: Превышена максимальная длина (>79 символов)

```
| 494 -> headers, _ = self.do_login(client, data["email"], data["password"])
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_ban.py: Строка 527: Превышена максимальная длина (>79 символов)

```
| 527 -> headers, _ = self.do_login(client, email, data["password"],  
status_code=403)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_ban.py: Переменная BAN_MESSAGE: Некорректное именование

BAN_MESSAGE

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_ban.py: Строка 30: Слишком длинный комментарий

```
Login attempts are not registered, let's try to fail the login many times
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_expiration.py: Строка 13: Превышена максимальная длина (>79 символов)

```
| 13 -> if not Env.get_bool("MAIN_LOGIN_ENABLE") or not  
Env.get_bool("AUTH_ENABLE"):
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_expiration.py: Строка 19: Превышена максимальная длина (>79 символов)

```
| 19 -> expiration = datetime.now(pytz.utc) + timedelta(seconds=expiration_time)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_expiration.py: Строка 23: Превышена максимальная длина (>79 символов)

```
| 23 -> valid_headers, _ = self.do_login(client, data["email"], data["password"])
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_expiration.py: Строка 44: Превышена максимальная длина (>79 символов)

```
| 44 -> # This token was valid before the expiration, but should be no longer  
valid
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_expiration.py: Строка 63: Превышена максимальная длина (>79 символов)

```
| 63 -> expiration = datetime.now(pytz.utc) + timedelta(seconds=expiration_time)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_expiration.py: Строка 72: Превышена максимальная длина (>79 символов)

```
| 72 -> valid_headers, _ = self.do_login(client, data["email"], data["password"])
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_expiration.py: Строка 97: Превышена максимальная длина (>79 символов)

```
| 97 -> expiration = datetime.now(pytz.utc) + timedelta(seconds=expiration_time)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_expiration.py: Строка 106: Превышена максимальная длина (>79 символов)

```
| 106 -> valid_headers, _ = self.do_login(client, data["email"], data["password"])
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_expiration.py: Строка 110: Превышена максимальная длина (>79 символов)

```
| 110 -> expiration = datetime.now(pytz.utc) - timedelta(seconds=expiration_time)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_expiration.py: Строка 44: Слишком длинный комментарий

```
This token was valid before the expiration, but should be no longer valid
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_profile_logout.py: Строка 200: Превышена максимальная длина (>79 символов)

```
| 200 -> encoded_credentials = base64.b64encode(str.encode(credentials)).decode("utf-8")
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_profile_logout.py: Строка 215: Превышена максимальная длина (>79 символов)

```
| 215 -> def test_03_change_profile(self, client: FlaskClient, faker: Faker) -> None:
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_profile_logout.py: Строка 216: Превышена максимальная длина (>79 символов)

```
| 216 -> if not Env.get_bool("MAIN_LOGIN_ENABLE") or not  
Env.get_bool("AUTH_ENABLE"):
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_profile_logout.py: Строка 324:
Превышена максимальная длина (>79 символов)

```
| 324 -> data["totp_code"] =  
BaseTests.generate_totp(BaseAuthentication.default_user)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_profile_logout.py: Строка 336:
Превышена максимальная длина (>79 символов)

```
| 336 -> # Reverse the list and skip all delete tokens to find the change password  
event
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_profile_logout.py: Строка 358:
Превышена максимальная длина (>79 символов)

```
| 358 -> data["totp_code"] =  
BaseTests.generate_totp(BaseAuthentication.default_user)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_profile_logout.py: Строка 363:
Превышена максимальная длина (>79 символов)

```
| 363 -> # Reverse the list and skip all delete tokens to find the change password  
event
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_profile_logout.py: Строка 377:
Превышена максимальная длина (>79 символов)

```
| 377 -> client, BaseAuthentication.default_user,  
BaseAuthentication.default_password
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_profile_logout.py: Строка 420:
Превышена максимальная длина (>79 символов)

```
| 420 -> def test_06_token_ip_validity(self, client: FlaskClient, faker: Faker)  
-> None:
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_profile_logout.py: Строка 444:
Превышена максимальная длина (>79 символов)

```
| 444 -> # After the failure the token is still valid if used from the correct IP
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_profile_logout.py: Строка 448:
Превышена максимальная длина (>79 символов)

```
| 448 -> # Another option to provide IP is through the header passed by nginx
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_profile_logout.py: Строка 458:
Превышена максимальная длина (>79 символов)

```
| 458 -> def test_07_totp_failures(self, client: FlaskClient, faker: Faker) ->
None:
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_profile_logout.py: Строка 495:
Превышена максимальная длина (>79 символов)

```
| 495 -> assert "You do not provided a valid verification code" in resp["errors"]
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_profile_logout.py: Строка 551:
Превышена максимальная длина (>79 символов)

```
| 551 -> assert "You do not provided a valid verification code" in resp["errors"]
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_profile_logout.py: Строка 613:
Превышена максимальная длина (>79 символов)

```
| 613 -> r = client.put(f"{AUTH_URI}/profile", json=data, headers=headers)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_profile_logout.py: Строка 625:
Превышена максимальная длина (>79 символов)

```
| 625 -> # Reverse the list and skip all delete tokens to find the change pwd
event
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_profile_logout.py: Функция test_02_GET_profile: Некорректное именован
test_02_GET_profile

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_profile_logout.py: Переменная USER: Некорректное именован
USER

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_profile_logout.py: Переменная PWD: Некорректное именован
PWD

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_profile_logout.py: Переменная USER: Некорректное именован
USER

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_profile_logout.py: Переменная PWD: Некорректное именован
PWD

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_profile_logout.py: Строка 336:
Слишком длинный комментарий

Reverse the list and skip all delete tokens to find the change password event

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_login_profile_logout.py: Строка 363:
Слишком длинный комментарий

Reverse the list and skip all delete tokens to find the change password event

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_pagination.py: Строка 16: Превышена максимальная длина (>79 символов)

```
| 16 -> r = client.get(f"{API_URI}/tests/pagination", query_string={"get_total": True})
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_pagination.py: Строка 61: Превышена максимальная длина (>79 символов)

```
| 61 -> r = client.get(f"{API_URI}/tests/pagination", query_string={"page": 20})
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_pagination.py: Строка 67: Превышена максимальная длина (>79 символов)

```
| 67 -> r = client.get(f"{API_URI}/tests/pagination", query_string={"size": 101})
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_pagination.py: Строка 70: Превышена максимальная длина (>79 символов)

```
| 70 -> r = client.get(f"{API_URI}/tests/pagination", query_string={"page": -5})
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_pagination.py: Строка 73: Превышена максимальная длина (>79 символов)

```
| 73 -> r = client.get(f"{API_URI}/tests/pagination", query_string={"size": -5})
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_pagination.py: Строка 88: Превышена максимальная длина (>79 символов)

```
| 88 -> r = client.post(f"{API_URI}/tests/pagination", json={"get_total": True})
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_pagination.py: Строка 115: Превышена максимальная длина (>79 символов)

```
| 115 -> r = client.post(f"{API_URI}/tests/pagination", json={"page": 2, "size": 10})
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_pagination.py: Строка 123: Превышена максимальная длина (>79 символов)

```
| 123 -> r = client.post(f"{API_URI}/tests/pagination", json={"page": 2, "size": 100})
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_pagination.py: Строка 146: Превышена максимальная длина (>79 символов)

```
| 146 -> r = client.post(f"{API_URI}/tests/pagination", json={"page": -5, "size": -5})
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_pagination.py: Строка 156: Превышена максимальная длина (>79 символов)

```
| 156 -> # Request get_total as body parameter but is ignored => sent a list of elements
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_pagination.py: Строка 159: Превышена максимальная длина (>79 символов)

```
| 159 -> # Request get_total as query parameter but is ignored => sent a list of elements
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_pagination.py: Строка 161: Превышена максимальная длина (>79 символов)

```
| 161 -> f"{API_URI}/tests/pagination", json={}, query_string={"get_total": True}
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_pagination.py: Функция test_GET_specs: Некорректное именование

test_GET_specs

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_pagination.py: Строка 156: Слишком длинный комментарий


```
Request get_total as body parameter but is ignored => sent a list of elements
```

`./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_pagination.py`: Строка 159: Слишком длинный комментарий

```
Request get_total as query parameter but is ignored => sent a list of elements
```

`./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_password_reset.py`: Строка 11: Превышена максимальная длина (>79 символов)

```
| 11 -> if not Env.get_bool("ALLOW_PASSWORD_RESET") or not  
Env.get_bool("AUTH_ENABLE"):
```

`./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_password_reset.py`: Строка 15: Превышена максимальная длина (>79 символов)

```
| 15 -> project_tile = get_project_configuration("project.title",  
default="YourProject")
```

`./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_password_reset.py`: Строка 57: Превышена максимальная длина (>79 символов)

```
| 57 -> assert f"Subject: {project_tile}: Password Reset" in mail.get("headers",  
"")
```

`./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_password_reset.py`: Строка 116: Превышена максимальная длина (>79 символов)

```
| 116 -> assert self.get_content(r) == "New password does not match with  
confirmation"
```

`./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_password_reset.py`: Строка 125: Превышена максимальная длина (>79 символов)

```
| 125 -> # Reverse the list and skip all delete tokens to find the change password  
event
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_password_reset.py: Строка 148: Превышена максимальная длина (>79 символов)

```
| 148 -> data["totp_code"] =  
BaseTests.generate_totp(BaseAuthentication.default_user)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_password_reset.py: Строка 157: Превышена максимальная длина (>79 символов)

```
| 157 -> # Reverse the list and skip all delete tokens to find the change password  
event
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_password_reset.py: Строка 125: Слишком длинный комментарий

```
Reverse the list and skip all delete tokens to find the change password event
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_password_reset.py: Строка 157: Слишком длинный комментарий

```
Reverse the list and skip all delete tokens to find the change password event
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_registration_and_activation.py: Строка 6: Превышена максимальная длина (>79 символов)

```
| 6 -> from restapi.tests import API_URI, AUTH_URI, BaseAuthentication,  
BaseTests, FlaskClient
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_registration_and_activation.py: Строка 12: Превышена максимальная длина (>79 символов)

```
| 12 -> if not Env.get_bool("ALLOW_REGISTRATION") or not  
Env.get_bool("AUTH_ENABLE"):
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_registration_and_activation.py: Строка 16: Превышена максимальная длина (>79 символов)

```
| 16 -> project_tile = get_project_configuration("project.title",  
default="YourProject")
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_registration_and_activation.py: Строка 69: Превышена максимальная длина (>79 символов)

```
| 69 -> assert self.get_content(r) == "Your password doesn't match the  
confirmation"
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_registration_and_activation.py: Строка 71: Превышена максимальная длина (>79 символов)

```
| 71 -> registration_data["password"] = faker.password(min_pwd_len, low=False,  
up=True)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_registration_and_activation.py: Строка 85: Превышена максимальная длина (>79 символов)

```
| 85 -> registration_data["password"] = faker.password(min_pwd_len, low=True,  
up=True)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_registration_and_activation.py: Строка 101: Превышена максимальная длина (>79 символов)

```
| 101 -> registration_data["password"] = registration_data["email"].split("@")[0]
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_registration_and_activation.py: Строка 109: Превышена максимальная длина (>79 символов)

```
| 109 -> registration_data["password"] = registration_data["name"] + "LMNopq678="
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_registration_and_activation.py: Строка 116: Превышена максимальная длина (>79 символов)

```
| 116 -> registration_data["password"] = registration_data["surname"] +  
"LMNopq678="
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_registration_and_activation.py: Строка 128: Превышена максимальная длина (>79 символов)

```
| 128 -> registration_message = "We are sending an email to your email address  
where "
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_registration_and_activation.py: Строка 129: Превышена максимальная длина (>79 символов)

```
| 129 -> registration_message += "you will find the link to activate your  
account"
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_registration_and_activation.py: Строка 157: Превышена максимальная длина (>79 символов)

```
| 157 -> assert f"Subject: {project_tile}: Account activation" in  
mail.get("headers", "")
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_registration_and_activation.py: Строка 191: Превышена максимальная длина (>79 символов)

```
| 191 -> r = client.post(f"{AUTH_URI}/profile/activate", json={"username":  
invalid})
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_registration_and_activation.py: Строка 196: Превышена максимальная длина (>79 символов)

```
| 196 -> activation_message = "We are sending an email to your email address where  
"
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_registration_and_activation.py: Строка 200: Превышена максимальная длина (>79 символов)

```
| 200 -> f"{AUTH_URI}/profile/activate", json={"username": faker.ascii_email()}
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_registration_and_activation.py: Строка

202: Превышена максимальная длина (>79 символов)

```
| 202 -> # return is 200, but no token will be generated and no mail will be sent
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_registration_and_activation.py: Строка

203: Превышена максимальная длина (>79 символов)

```
| 203 -> # but it respond with the activation msg and hides the non existence of the user
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_registration_and_activation.py: Строка

227: Превышена максимальная длина (>79 символов)

```
| 227 -> assert f"Subject: {project_tile}: Account activation" in mail.get("headers", "")
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_registration_and_activation.py: Строка

344: Превышена максимальная длина (>79 символов)

```
| 344 -> f"{API_URI}/admin/users/{uuid}", json={"is_active": True}, headers=headers
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_registration_and_activation.py: Строка

360: Превышена максимальная длина (>79 символов)

```
| 360 -> "Token not properly invalidated, still bount to user {}", t.get(id)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_registration_and_activation.py: Строка

203: Слишком длинный комментарий

```
but it respond with the activation msg and hides the non existence of the user
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_send_mail.py: Строка 12: Превышена максимальная длина (>79 символов)

```
| 12 -> not Connector.check_availability("smtp") or not Env.get_bool("AUTH_ENABLE"),
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_specs.py: Строка 65: Превышена максимальная длина (>79 символов)

```
| 65 -> # IMPORT: if authentication is enabled with content is replaced by the full spec
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_specs.py: Строка 77: Превышена максимальная длина (>79 символов)

```
| 77 -> assert endpoint["parameters"][0]["schema"]["$ref"] == "#/definitions/Input"
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_specs.py: Строка 174: Превышена максимальная длина (>79 символов)

```
| 174 -> # no property specified for custom fields... apispec is unable to convert it
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_specs.py: Функция test_GET_specs:
Некорректное именование
test_GET_specs

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_specs.py: Строка 65: Слишком длинный комментарий

```
IMPORT: if authentication is enabled with content is replaced by the full spec
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_specs.py: Строка 174: Слишком длинный комментарий

```
no property specified for custom fields... apispec is unable to convert it
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_status.py: Функция test_GET_status:
Некорректное именование
test_GET_status

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_tokens.py: Строка 52: Превышена максимальная длина (>79 символов)

```
| 52 -> r = client.get(f"{AUTH_URI}/tokens/{token_id}",  
headers=last_tokens_header)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_tokens.py: Строка 157: Превышена максимальная длина (>79 символов)

```
| 157 -> query_string={"page": 1, "size": 20, "input_filter": faker.pystr()},
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_tokens.py: Строка 213: Превышена максимальная длина (>79 символов)

```
| 213 -> # It may only be done if we were able to retrieve all tokens, in this  
case the
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_tokens.py: Строка 214: Превышена максимальная длина (>79 символов)

```
| 214 -> # first desc will be the last asc... But we cannot ensure to be able to  
always
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_tokens.py: Строка 229: Превышена максимальная длина (>79 символов)

```
| 229 -> r = client.delete(f"{API_URI}/admin/tokens/xyz",  
headers=last_tokens_header)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_tokens.py: Строка 233: Превышена максимальная длина (>79 символов)

```
| 233 -> r = client.delete(f"{AUTH_URI}/tokens/{token_id}",  
headers=last_tokens_header)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_tokens.py: Строка 245: Превышена максимальная длина (>79 символов)

```
| 245 -> r = client.delete(f"{AUTH_URI}/tokens/{token_id}",  
headers=last_tokens_header)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_tokens.py: Строка 263: Превышена максимальная длина (>79 символов)

```
| 263 -> user_header, token = self.do_login(client, data["email"],  
data["password"])
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_tokens.py: Строка 213: Слишком длинный комментарий

```
It may only be done if we were able to retrieve all tokens, in this case the
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_tokens.py: Строка 214: Слишком длинный комментарий

```
first desc will be the last asc... But we cannot ensure to be able to always
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_upload_download.py: Строка 43: Превышена максимальная длина (>79 символов)

```
| 43 -> # By setting force False only txt files will be allowed for upload
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_upload_download.py: Строка 44: Превышена максимальная длина (>79 символов)

```
| 44 -> # Strange, but it is how the endpoint is configured to improve the tests
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_upload_download.py: Строка 58: Превышена максимальная длина (>79 символов)

```
| 58 -> # By setting force False only txt files will be allowed for upload
```


./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_upload_download.py: Строка 59: Превышена максимальная длина (>79 символов)

```
| 59 -> # Strange, but it is how the endpoint is configured to improve the tests
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_upload_download.py: Строка 74: Превышена максимальная длина (>79 символов)

```
| 74 -> err = f"File '{self.fname}' already exists, use force parameter to  
overwrite"
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_upload_download.py: Строка 107: Превышена максимальная длина (>79 символов)

```
| 107 -> r = client.get(f"{API_URI}/tests/download/{upload_folder}/{self.fname}")
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_upload_download.py: Строка 122: Превышена максимальная длина (>79 символов)

```
| 122 -> r = client.get(f"{API_URI}/tests/download/{upload_folder}/{self.fname}")
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_upload_download.py: Строка 228: Превышена максимальная длина (>79 символов)

```
| 228 -> r =  
client.get(f"{API_URI}/tests/download/{upload_folder}/{uploaded_filename}")
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_upload_download.py: Строка 233: Превышена максимальная длина (>79 символов)

```
| 233 -> r =  
client.get(f"{API_URI}/tests/download/{upload_folder}/{uploaded_filename}")
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_upload_download.py: Строка 290: Превышена максимальная длина (>79 символов)

```
| 290 -> assert self.get_content(r) == "Permission denied: failed to write the file"
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_upload_download.py: Переменная STR_LEN: Некорректное именование

STR_LEN

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_upload_download.py: Переменная STR_LEN: Некорректное именование

STR_LEN

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_users_custom_fields.py: Строка 13: Превышена максимальная длина (>79 символов)

```
| 13 -> reason="This test needs authentication and registration to be available",
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_users_custom_fields.py: Строка 30: Превышена максимальная длина (>79 символов)

```
| 30 -> # Verify custom output fields (if defined) included in the profile response
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_users_custom_fields.py: Строка 39: Превышена максимальная длина (>79 символов)

```
| 39 -> # Verify custom input fields (if defined) included in the profile input schema
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_users_custom_fields.py: Строка 40: Превышена максимальная длина (>79 символов)

```
| 40 -> r = client.patch(f"{AUTH_URI}/profile", json={"get_schema": 1}, headers=headers)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_users_custom_fields.py: Строка 48:
Превышена максимальная длина (>79 символов)

```
| 48 -> pytest.fail(f"Input field {field} not found in profile input schema")
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_users_custom_fields.py: Строка 50:
Превышена максимальная длина (>79 символов)

```
| 50 -> # Verify custom registration fields (if defined) included in the reg.  
schema
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_users_custom_fields.py: Строка 60:
Превышена максимальная длина (>79 символов)

```
| 60 -> f"Input field {field} not found in registration input schema"
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_users_custom_fields.py: Строка 64:
Превышена максимальная длина (>79 символов)

```
| 64 -> # Verify custom admin input fields (if defined) included in admin users  
schema
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_users_custom_fields.py: Строка 76:
Превышена максимальная длина (>79 символов)

```
| 76 -> f"Input field {field} not found in admin users input schema"
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_users_custom_fields.py: Строка 79:
Превышена максимальная длина (>79 символов)

```
| 79 -> # Verify custom admin output fields (if defined) included in admin users  
output
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_users_custom_fields.py: Строка 30:
Слишком длинный комментарий

```
Verify custom output fields (if defined) included in the profile response
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_users_custom_fields.py: Строка 39:

Слишком длинный комментарий

```
Verify custom input fields (if defined) included in the profile input schema
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_users_custom_fields.py: Строка 50:

Слишком длинный комментарий

```
Verify custom registration fields (if defined) included in the reg. schema
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_users_custom_fields.py: Строка 64:

Слишком длинный комментарий

```
Verify custom admin input fields (if defined) included in admin users schema
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_endpoints_users_custom_fields.py: Строка 79:

Слишком длинный комментарий

```
Verify custom admin output fields (if defined) included in admin users output
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_authentication.py: Строка 67: Превышена максимальная длина (>79 символов)

```
| 67 -> assert ret_text == "The new password cannot match the previous password"
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_authentication.py: Строка 106: Превышена максимальная длина (>79 символов)

```
| 106 -> pwd = faker.password(min_pwd_len, low=True, up=True, digits=True, symbols=True)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_authentication.py: Строка 122: Превышена максимальная длина (>79 символов)

```
| 122 -> # This is because with "strange characters" it is not ensured that:
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_authentication.py: Строка 127: Превышена максимальная длина (>79 символов)

```
| 127 -> # This is to prevent failures for other reasons like length of chars
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_authentication.py: Строка 130: Превышена максимальная длина (>79 символов)

```
| 130 -> pwd=pp, old_pwd=old_pwd, email=email, name=name, surname=surname
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_authentication.py: Строка 146: Превышена максимальная длина (>79 символов)

```
| 146 -> # This is because with "strange characters" it is not ensured that:
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_authentication.py: Строка 151: Превышена максимальная длина (>79 символов)

```
| 151 -> # This is to prevent failures for other reasons like length of chars
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_authentication.py: Строка 154: Превышена максимальная длина (>79 символов)

```
| 154 -> pwd=pp, old_pwd=old_pwd, email=email, name=name, surname=surname
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_authentication.py: Строка 157: Превышена максимальная длина (>79 символов)

```
| 157 -> assert txt == "Password is too weak, can't contain your email address"
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_authentication.py: Строка 161: Превышена максимальная длина (>79 символов)

```
| 161 -> pwd="Bob1234567!", old_pwd=old_pwd, email=email, name="Bob",  
surname=surname
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_authentication.py: Строка 192: Превышена максимальная длина (>79 символов)

```
| 192 -> match=rf"Password is too short, use at least {min_pwd_len} characters",
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_authentication.py: Строка 194: Превышена максимальная длина (>79 символов)

```
| 194 -> # the first password parameter is only checked for new password strenght
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_authentication.py: Строка 196: Превышена максимальная длина (>79 символов)

```
| 196 -> # pwd validity will be checked once completed checks on new password
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_authentication.py: Строка 230: Превышена максимальная длина (>79 символов)

```
| 230 -> with pytest.raises(Unauthorized, match=r"Verification code is missing"):
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_authentication.py: Строка 249: Превышена максимальная длина (>79 символов)

```
| 249 -> with pytest.raises(Unauthorized, match=r"Verification code is not valid"):
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_authentication.py: Строка 253: Превышена максимальная длина (>79 символов)

```
| 253 -> with pytest.raises(Unauthorized, match=r"Verification code is not valid"):
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_authentication.py: Строка 265: Превышена максимальная длина (>79 символов)

```
| 265 -> with pytest.raises(Unauthorized, match=r"Verification code is not valid"):
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_authentication.py: Строка 269: Превышена максимальная длина (>79 символов)

```
| 269 -> with pytest.raises(Unauthorized, match=r"Verification code is not valid"):
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_authentication.py: Строка 302: Превышена максимальная длина (>79 символов)

```
| 302 -> def test_tokens_management(self, client: FlaskClient, faker: Faker) -> None:
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_authentication.py: Строка 453: Превышена максимальная длина (>79 символов)

```
| 453 -> assert not auth.verify_roles(None, ["A", "B"], required_roles="invalid")
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_authentication.py: Строка 482: Превышена максимальная длина (>79 символов)

```
| 482 -> assert auth.get_user(username=BaseAuthentication.default_user) is None
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_authentication.py: Строка 487: Превышена максимальная длина (>79 символов)

```
| 487 -> assert auth.get_user(username=BaseAuthentication.default_user) is not None
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_authentication.py: Строка 573: Превышена максимальная длина (>79 символов)

```
| 573 -> # Verify init without force flag will not restore default user and group
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_authentication.py: Строка 585: Превышена максимальная длина (>79 символов)

```
| 585 -> # Verify init with force flag will not restore the default user and group
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_authentication.py: Строка 600: Превышена максимальная длина (>79 символов)

```
| 600 -> # https://clamytoe.github.io/articles/2020/Mar/12/testing-abcs-with-abstract-methods-with-pytest
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_authentication.py: Строка 612: Превышена максимальная длина (>79 символов)

```
| 612 -> auth.get_user(username=faker.ascii_email(), user_id=faker.pystr()) is None
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_authentication.py: Строка 619: Превышена максимальная длина (>79 символов)

```
| 619 -> assert auth.get_group(group_id=faker.pystr(), name=faker.pystr()) is None
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_authentication.py: Строка 630: Превышена максимальная длина (>79 символов)

```
| 630 -> assert auth.get_tokens(user=user, token_jti=faker.pystr(), get_all=True) is None
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_authentication.py: Строка 636: Превышена максимальная длина (>79 символов)

```
| 636 -> user=user, token=faker.pystr(), payload={}, token_type=faker.pystr()
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_authentication.py: Строка 647: Превышена максимальная длина (>79 символов)


```
| 647 -> assert auth.create_role(name=faker.pystr(), description=faker.pystr())  
is None
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_authentication.py: Строка 651: Превышена максимальная длина (>79 символов)

```
| 651 -> assert auth.create_user(userdata={}, roles=[faker.pystr()], group=group)  
is None
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_authentication.py: Строка 659: Превышена максимальная длина (>79 символов)

```
| 659 -> auth.save_login(username=faker.ascii_email(), user=user, failed=True)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_authentication.py: Строка 663: Превышена максимальная длина (>79 символов)

```
| 663 -> auth.save_login(username=faker.ascii_email(), user=None, failed=True)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_authentication.py: Строка 667: Превышена максимальная длина (>79 символов)

```
| 667 -> auth.save_login(username=faker.ascii_email(), user=user, failed=False)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_authentication.py: Строка 671: Превышена максимальная длина (>79 символов)

```
| 671 -> auth.save_login(username=faker.ascii_email(), user=None, failed=False)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_authentication.py: Переменная EXPIRATION: Некорректное именован

EXPIRATION

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_authentication.py: Строка 600: Слишком

длинный комментарий

<https://clamytoe.github.io/articles/2020/Mar/12/testing-abcs-with-abstract-methods-with-pytest>

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_miscellaneous.py: Строка 118: Превышена максимальная длина (>79 символов)

```
| 118 -> ResponseMaker.get_schema_type(f, fields.Str(metadata={"password": True}))
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_miscellaneous.py: Строка 124: Превышена максимальная длина (>79 символов)

```
| 124 -> assert ResponseMaker.get_schema_type(f, fields.DateTime()) == "datetime"
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_miscellaneous.py: Строка 125: Превышена максимальная длина (>79 символов)

```
| 125 -> assert ResponseMaker.get_schema_type(f, fields.AwareDateTime()) == "datetime"
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_miscellaneous.py: Строка 126: Превышена максимальная длина (>79 символов)

```
| 126 -> assert ResponseMaker.get_schema_type(f, fields.NaiveDateTime()) == "datetime"
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_miscellaneous.py: Строка 136: Превышена максимальная длина (>79 символов)

```
| 136 -> assert ResponseMaker.get_schema_type(f, fields.List(fields.Str())) == "string[]"
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_miscellaneous.py: Строка 137: Превышена максимальная длина (>79 символов)

```
| 137 -> assert ResponseMaker.get_schema_type(f, fields.Nested(MySchema())) == "nested"
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_miscellaneous.py: Строка 142: Превышена максимальная длина (>79 символов)

```
| 142 -> # assert ResponseMaker.get_schema_type(f, fields.Constant("x")) == "string"
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_miscellaneous.py: Строка 144: Превышена максимальная длина (>79 символов)

```
| 144 -> # assert ResponseMaker.get_schema_type(f, fields.Function()) == "string"
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_miscellaneous.py: Строка 148: Превышена максимальная длина (>79 символов)

```
| 148 -> # assert ResponseMaker.get_schema_type(f, fields.TimeDelta()) == "string"
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_miscellaneous.py: Строка 206: Превышена максимальная длина (>79 символов)

```
| 206 -> assert jsonifier(datetime(2023, 1, 21, 11, 34, 21)) == "2023-01-21T11:34:21"
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_miscellaneous.py: Строка 212: Превышена максимальная длина (>79 символов)

```
| 212 -> # This is a valid package containing other packages... but no task will be found
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_miscellaneous.py: Строка 221: Превышена максимальная длина (>79 символов)

```
| 221 -> mcls = Meta.get_classes_from_module("this-should-not-exist") # type: ignore
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_miscellaneous.py: Строка 225: Превышена максимальная длина (>79 символов)

```
| 225 -> assert Meta.get_class("this-should-not-exist", "this-should-not-exist")  
is None
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_miscellaneous.py: Строка 226: Превышена максимальная длина (>79 символов)

```
| 226 -> assert Meta.get_class("initialization", "this-should-not-exist") is None
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_miscellaneous.py: Строка 237: Превышена максимальная длина (>79 символов)

```
| 237 -> # This method is not very robust... but... let's test the current  
implementation
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_miscellaneous.py: Строка 238: Превышена максимальная длина (>79 символов)

```
| 238 -> # It basically return the first args if it is an instance of some classes
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_miscellaneous.py: Строка 243: Превышена максимальная длина (>79 символов)

```
| 243 -> models = Meta.import_models("this-should", "not-exist", mandatory=False)
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_miscellaneous.py: Строка 470: Превышена максимальная длина (>79 символов)

```
| 470 -> assert obfuscate_dict({"access_token": "y"}) == {"access_token": "*****"}
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_miscellaneous.py: Строка 473: Превышена максимальная длина (>79 символов)

```
| 473 -> assert obfuscate_dict({"new_password": "y"}) == {"new_password": "*****"}
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_miscellaneous.py: Строка 474: Превышена максимальная длина (>79 символов)

```
| 474 -> assert obfuscate_dict({"password_confirm": "y"}) == {"password_confirm": "****"}
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_miscellaneous.py: Строка 699: Превышена максимальная длина (>79 символов)

```
| 699 -> assert seconds_to_human(86402) == "1 day, 0 hours, 0 minutes, 2 seconds"
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_miscellaneous.py: Строка 704: Превышена максимальная длина (>79 символов)

```
| 704 -> assert seconds_to_human(86522) == "1 day, 0 hours, 2 minutes, 2 seconds"
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_miscellaneous.py: Строка 715: Превышена максимальная длина (>79 символов)

```
| 715 -> assert seconds_to_human(22222222) == "257 days, 4 hours, 50 minutes, 22 seconds"
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_miscellaneous.py: Строка 752: Превышена максимальная длина (>79 символов)

```
| 752 -> # Note: This is a replacement of the normal DelimitedList defined by rapydo
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_miscellaneous.py: Строка 756: Превышена максимальная длина (>79 символов)

```
| 756 -> # Note: This is a replacement of the normal List list defined by rapydo
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_miscellaneous.py: Строка 894: Превышена максимальная длина (>79 символов)

```
| 894 -> injected_parameters = inject_callback_parameters(missing_endpoint, {}, {})
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_miscellaneous.py: Строка 906: Превышена максимальная длина (>79 символов)

```
| 906 -> injected_parameters = inject_callback_parameters(wrong_endpoint, {}, {})
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_miscellaneous.py: Строка 142: Слишком длинный комментарий

```
assert ResponseMaker.get_schema_type(f, fields.Constant("x")) == "string"
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_miscellaneous.py: Строка 212: Слишком длинный комментарий

```
This is a valid package containing other packages... but no task will be found
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_miscellaneous.py: Строка 237: Слишком длинный комментарий

```
This method is not very robust... but... let's test the current implementation
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_module_miscellaneous.py: Строка 752: Слишком длинный комментарий

```
Note: This is a replacement of the normal DelimitedList defined by rapydo
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_xxx_unused_credentials_step2.py: Строка 52: Превышена максимальная длина (>79 символов)

```
| 52 -> assert events[0].payload["username"] == BaseTests.unused_credentials[0]
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_xxx_unused_credentials_step2.py: Строка 54: Превышена максимальная длина (>79 символов)

```
| 54 -> events[0].payload["motivation"] == "account blocked due to inactivity"
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test
_xxx_unused_credentials_step2.py: Строка 57:
Превышена максимальная длина (>79 символов)

```
| 57 -> assert events[1].payload["username"] == BaseTests.unused_credentials[0]
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test
_xxx_unused_credentials_step2.py: Строка 59:
Превышена максимальная длина (>79 символов)

```
| 59 -> events[1].payload["motivation"] == "account blocked due to inactivity"
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test
_zzz1_destroy.py: Строка 2: Превышена максимальная
длина (>79 символов)

```
| 2 -> This test is intended to be executed as second-last, just before the  
re-init test
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test
_zzz1_destroy.py: Строка 4: Превышена максимальная
длина (>79 символов)

```
| 4 -> Beware: if env TEST_DESTROY_MODE == 1 this test will destroy your database,  
be careful
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test
_zzz1_destroy.py: Строка 21: Превышена максимальная
длина (>79 символов)

```
| 21 -> reason="This test needs authentication and TEST_DESTROY_MODE to be  
enabled",
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test
_zzz2_re-init.py: Строка 4: Превышена максимальная
длина (>79 символов)

```
| 4 -> Beware: if env TEST_DESTROY_MODE == 1 this test will destroy your database,  
be careful
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_zzz2_re-init.py: Строка 20: Превышена максимальная длина (>79 символов)

```
| 20 -> reason="This test needs authentication and TEST_DESTROY_MODE to be enabled",
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_zzz2_re-init.py: Строка 39: Превышена максимальная длина (>79 символов)

```
| 39 -> # due to the multiple app created... should be an issue specific of this test
```

./path/to/http-api-3.1//http-api-3.1/tests/base/test_zzz2_re-init.py: Строка 39: Слишком длинный комментарий

```
due to the multiple app created... should be an issue specific of this test
```

./path/to/http-api-3.1//http-api-3.1/tests/conftest.py: Строка 14: Слишком длинный комментарий

```
Beware, this replaces the standard faker fixture provided by Faker it-self
```