

NovaData LMS: схема переменных окружения

Область: `lms-novadata` , `SQLChecker` , `code_checker`

Документ фиксирует актуальные переменные окружения, которые используются проектом. Значения секретов в примерах являются шаблонами и не должны использоваться в production.

Связанные документы:

- `README_DOCUMENTATION.md` - индекс комплекта документации и рекомендуемый порядок чтения.
- `PROJECT_DOCUMENTATION.md` - общая техническая документация проекта.
- `DEPLOYMENT_GUIDE.md` - подробная инструкция по развертыванию.
- `API_CONTRACTS.md` - API-контракты LMS/checkers.
- `OPERATIONS_RUNBOOK.md` - регламент эксплуатации и восстановления после сбоев.
- `PROGRAMMING_TASK_AUTHOR_GUIDE.md` - руководство по созданию programming-заданий.
- `SQL_TASK_AUTHOR_GUIDE.md` - руководство по созданию SQL-заданий.

1. Расположение env-файлов

Рекомендуемые места хранения:

```
lms-novadata/lms_project/.env
SQLChecker/.env
code_checker/.env
```

Для production секреты можно хранить не в файлах, а в системном secret manager или переменных окружения systemd/Docker. Главное условие - приложение должно видеть те же имена переменных.

2. LMS: обязательные переменные

Файл:

```
lms-novadata/lms_project/.env
```

Переменная	Обязательность	Пример	Назначение
DB_USER	Да	lms_user	Пользователь MySQL для LMS
DB_PASSWORD	Да	CHANGE_ME	Пароль пользователя MySQL
DB_HOST	Да	localhost	Host MySQL
DB_PORT	Да	3306	Порт MySQL
S3_ACCESS_KEY	Да, если включено S3	CHANGE_ME	Access key S3/Selectel/MinIO
S3_SECRET_KEY	Да, если включено S3	CHANGE_ME	Secret key S3/Selectel/MinIO
S3_BUCKET_NAME	Да, если включено S3	lms	Имя bucket
S3_ENDPOINT	Да, если включено S3	s3.ru-7.storage.selcloud.ru	Host S3 endpoint без https://
VALIDATOR_PROGRAMMING_TOKEN	Да, для code_checker	CHANGE_ME	Токен доступа programming checker к LMS API
VALIDATOR_SQL_TOKEN	Опционально/резерв	CHANGE_ME	Токен для SQL validator, если используется pull-интеграция

Текущая особенность S3 в settings.py : endpoint собирается как https:// + S3_ENDPOINT . Поэтому значение S3_ENDPOINT должно быть host-частью без схемы.

3. LMS: опциональные переменные

Переменная	Пример	Назначение
FFMPEG_PATH	/usr/bin/ffmpeg	Путь к FFmpeg. На Linux в коде используется /usr/bin/ffmpeg ; на

Переменная	Пример	Назначение
		Windows может задаваться через env
FFPROBE_PATH	/usr/bin/ffprobe	Путь к FFprobe. На Windows может задаваться через env
VIDEO_ASSETS_DIR	/var/www/lms/static/intro	Каталог ассетов для обработки видео
VIDEO_WATERMARK_DIR	/var/www/lms/static/intro	Каталог watermark-файлов
VIDEO_INTRO_FILE	/var/www/lms/static/intro/intro.mp4	Файл intro-видео
VIDEO_PROCESSING_MODE	cpu	Режим обработки видео: cpu , gpu , auto
VIDEO_GPU_DEVICE	0	Номер GPU для обработки видео
JITSI_SERVER_URL	https://jitsi.example.com	URL Jitsi-сервера для вебинаров
JITSI_APP_ID	lms-app	App ID для JWT Jitsi
JITSI_APP_SECRET	CHANGE_ME	Secret для подписи JWT Jitsi
JITSI_JWT_EXPIRE_SECONDS	7200	Время жизни JWT для Jitsi
JUPYTERHUB_URL	https://jupyter.example.com	Базовый URL JupyterHub
WIDGET_FRAME_SRC	https://chat.example.com/frame	URL iframe внешнего chat-widget
WIDGET_API_BASE_URL	https://chat.example.com/api	API внешнего chat-widget
WIDGET_USE_DIRECT_CHAT_PATHS	true	Включение прямых путей chat-widget
WIDGET_SIGNING_SECRET	CHANGE_ME	Secret для подписи данных chat-widget
CHAT_SERVICE_FRAME_URL	https://chat.example.com/frame	URL frame внешнего chat-сервиса

4. LMS: текущие hardcoded-настройки

Следующие настройки сейчас не являются env-переменными, но влияют на развертывание.

Настройка	Текущее значение	Риск
SQL_CHECKER_URL	http://10.10.10.18:8000/	При смене адреса SQLChecker нужно менять код или settings
CELERY_BROKER_URL	redis://localhost:6379/0	Redis должен быть локальным, иначе нужно менять settings
CELERY_RESULT_BACKEND	redis://localhost:6379/0	Аналогично broker
AWS_S3_REGION_NAME	ru-7	Регион зафиксирован под текущего S3-провайдера
AWS_S3_CUSTOM_DOMAIN	{bucket}.s3.ru-7.storage.selcloud.ru	Домен зафиксирован под Selectel

Рекомендуемые env-переменные для будущей доработки:

```
SQL_CHECKER_URL=http://sql-checker.example.com:8000/
REDIS_URL=redis://localhost:6379/0
S3_REGION=ru-7
S3_CUSTOM_DOMAIN=lms.s3.ru-7.storage.selcloud.ru
```

5. SQLChecker: переменные приложения

Файл:

```
SQLChecker/.env
```

Переменная	Обязательность	Значение по умолчанию в коде	Пример для Docker Compose	Назначение
DB_HOST	Да	localhost	postgres-db	Host PostgreSQL
DB_PORT	Да	5432	5432	Порт PostgreSQL
DB_USER	Да	user	user	Пользователь PostgreSQL

Переменная	Обязательность	Значение по умолчанию в коде	Пример для Docker Compose	Назначение
DB_PASSWORD	Да	password	password	Пароль PostgreSQL
DB_NAME	Да	mydb	mydb	Имя БД PostgreSQL
CH_HOST	Нет	localhost	clickhouse-db	Host ClickHouse
CH_PORT	Нет	8123	8123	HTTP-порт ClickHouse
CH_USER	Нет	default	default	Пользователь ClickHouse
CH_PASSWORD	Нет	пусто	пусто	Пароль ClickHouse
MAX_PARALLEL_CHECKS	Нет	10	10	Максимум параллельных проверок
QUERY_TIMEOUT_MS	Нет	30000	30000	Таймаут SQL-запроса в миллисекундах

6. SQLChecker: переменные контейнера PostgreSQL

Эти переменные нужны не Python-приложению, а контейнеру `postgres-db` из `docker-compose.yml`.

Переменная	Пример	Назначение
POSTGRES_USER	user	Пользователь, создаваемый контейнером PostgreSQL
POSTGRES_PASSWORD	password	Пароль пользователя PostgreSQL
POSTGRES_DB	mydb	База, создаваемая контейнером PostgreSQL

Важно: `POSTGRES_*` и `DB_*` должны быть согласованы. Например, `POSTGRES_USER` должен совпадать с `DB_USER`, если SQLChecker подключается к БД, созданной этим же контейнером.

7. code_checker: переменные

Файл:

```
code_checker/.env
```

Переменная	Обязательность	Пример	Назначение
LMS_BASE_URL	Да, для worker	http://10.10.10.16:8000/	Базовый URL LMS. Должен заканчиваться на /
LMS_API_TOKEN	Да, для worker	CHANGE_ME	Токен доступа к LMS validator API
PYTHONPATH	Задается в compose	/app	Путь импорта внутри контейнера

LMS_API_TOKEN должен совпадать с VALIDATOR_PROGRAMMING_TOKEN В lms-novadata/lms_project/.env .

Worker использует токен в заголовке:

```
X-API-Key: <LMS_API_TOKEN>
```

И обращается к LMS endpoints:

```
GET    /api/validator/submissions/?status=pending
GET    /api/validator/submissions/{id}/
PATCH /api/validator/submissions/{id}/
```

8. Согласование токенов валидаторов

В LMS токены читаются из .env по шаблону:

```
VALIDATOR_<SERVICE_TYPE>_TOKEN
```

Тип сервиса определяется из имени переменной:

Переменная LMS	service_type	Кто использует
VALIDATOR_PROGRAMMING_TOKEN	programming	code_checker
VALIDATOR_SQL_TOKEN	sql	SQL validator, если используется pull-модель

Пример согласованной пары:

```
# lms-novadata/lms_project/.env
VALIDATOR_PROGRAMMING_TOKEN=abc123
```

```
# code_checker/.env
LMS_API_TOKEN=abc123
```

Если значения не совпадают, LMS вернет `403 Forbidden`, а worker не сможет забрать задания.

9. Минимальные .env для локальной разработки

9.1. LMS

```
DB_USER=lms_user
DB_PASSWORD=local_password
DB_HOST=localhost
DB_PORT=3306

S3_ACCESS_KEY=local_access
S3_SECRET_KEY=local_secret
S3_BUCKET_NAME=lms-dev
S3_ENDPOINT=s3.ru-7.storage.selcloud.ru

VALIDATOR_PROGRAMMING_TOKEN=dev-programming-token
VALIDATOR_SQL_TOKEN=dev-sql-token

JITSI_SERVER_URL=
JITSI_APP_ID=lms-app
JITSI_APP_SECRET=
JITSI_JWT_EXPIRE_SECONDS=7200
JUPYTERHUB_URL=
```

9.2. SQLChecker

```
POSTGRES_USER=user
POSTGRES_PASSWORD=password
POSTGRES_DB=mydb
```

```
DB_HOST=postgres-db
DB_PORT=5432
DB_USER=user
DB_PASSWORD=password
DB_NAME=mydb
```

```
CH_HOST=clickhouse-db
CH_PORT=8123
CH_USER=default
CH_PASSWORD=
```

```
MAX_PARALLEL_CHECKS=10
QUERY_TIMEOUT_MS=30000
```

9.3. code_checker

```
LMS_BASE_URL=http://localhost:8000/
LMS_API_TOKEN=dev-programming-token
```

Если `code_checker` запущен в Docker, `localhost` внутри контейнера указывает на сам контейнер, а не на хост. В таком случае нужно использовать адрес, доступный из контейнера: IP сервера, DNS-имя или `host.docker.internal`, если он поддерживается окружением.

10. Проверка переменных окружения

10.1. LMS

Проверить, что Django видит настройки:

```
cd /var/www/lms
source venv/bin/activate
python manage.py check
```

Проверить подключение к БД:

```
python manage.py migrate --check
```

10.2. SQLChecker

```
cd /opt/nd_lms/SQLChecker
docker compose config
docker compose up -d
curl http://localhost:8000/health
```

10.3. code_checker

```
cd /opt/nd_lms/code_checker
docker compose config
docker compose up -d worker
docker compose logs -f worker
```

В логах worker не должно быть сообщения:

```
FATAL ERROR: LMS_BASE_URL or LMS_API_TOKEN is not configured.
```