

Документация СЕРВИС "РЕЕСТР-КОНТРОЛЬ"

1. Общее описание

СЕРВИС "РЕЕСТР-КОНТРОЛЬ" представляет собой веб-приложение для валидации медицинских реестров, состоящее из двух основных компонентов:

- веб-интерфейс на PHP/Laravel для взаимодействия с пользователями;
- ядро обработки данных на Python для выполнения бизнес-логики валидации.

СЕРВИС "РЕЕСТР-КОНТРОЛЬ" предназначен для автоматизированной проверки корректности медицинских реестров, обработки ошибок, формирования отчётов и интеграции с внешними системами.

2. Функциональные характеристики

1) Валидация медицинских реестров

- проверка ошибок ввода в сформированных реестрах;
- поддержка различных типов медицинских реестров (В-М, структура, санкции МЭК);
- генерация отчётов об ошибках в формате .xlsx;
- интеграция с системами НСИ (нормативно-справочная информация).

2) Управление пользователями и задачами

- авторизация пользователей по токенам;
- разграничение прав доступа (администраторы, обычные пользователи);
- управление пользователями (добавление, удаление, архивация);
- отслеживание статуса выполняемых задач;
- система новостей и уведомлений.

3) Загрузка и обработка файлов

Загрузка файлов реестров различных типов:

- В-М файлы (VM);
- подсчёты (count);
- санкции МЭК (sank-mek);
- ошибки (errors);
- структуры (structure).

4) Интеграция с внешними системами

- отчёты врачей (doctors-report);
- списки ДН (dn_list);
- госпитализации (komtek_hosp);
- скачивание обработанных файлов;
- работа с базами данных PostgreSQL и MySQL;
- интеграция с Redis для кэширования;
- поддержка Docker-контейнеризации;
- API для взаимодействия между PHP и Python компонентами.

5) Формирование отчётов

- отчёты по валидации реестров;
- статистические отчёты;
- отчёты по госпитализациям;
- отчёты по отказанным случаям.

3. Архитектура системы

PHP/Laravel компонент

- маршруты и контроллеры в routes/web.php:27-107;
- аутентификация пользователей;
- управление сессиями;
- формирование веб-интерфейса;
- оркестрация выполнения Python-скриптов.

Python компонент

Основные модули расположены в директории python/:

Основные модули обработки:

- errors.py — основной модуль валидации ошибок;
- vm.py — обработка В-М файлов;
- count.py — подсчёты реестров;
- sank_mek.py — обработка санкций МЭК;
- controls.py — функции валидации;
- hosp.py — обработка госпитализаций.

Модули отчётов:

- doctors_report_module.py — отчёты врачей;
- doctors_report_dn_school.py — отчёты по отказам в школах;
- export_komtek_hosp.py — экспорт госпитализаций.

Модули работы с данными:

- fill_d_history.py — сбор ДВН/ПВН записей;
- fill_hosp_db.py — обновление БД госпитализации;
- update_dir.py — обновление справочников НСИ;
- make_df.py — конвертация XML справочников.

Вспомогательные модули:

- config.py — конфигурация системы;
- utils.py — вспомогательные функции;
- error_utils.py — обработка ошибок;
- refs.py — работа со справочниками.

Инфраструктура:

- Dockerfile и docker-compose.yml для контейнеризации;
- requirements.txt — зависимости Python;
- system_codes.md — системные коды возврата.

Системные требования

Веб-компонент (Laravel):

- PHP 8.0+;
- Laravel 9+;

- веб-сервер (Apache/Nginx);
- база данных MySQL/PostgreSQL;
- Redis для кэширования.

Обработчик данных (Python):

- Python 3.8+;
- PostgreSQL 12+;
- MySQL 5.7+;
- Redis 6.0+.

Зависимости Python:

- pandas 1.5.2 для обработки данных;
- openpyxl 3.0.10 для работы с Excel;
- psycopg2-binary 2.9.6 для PostgreSQL;
- mysql-connector-python 8.0.33 для MySQL;
- redis 4.5.5 для кэширования.

4. Основные справочники системы

Для валидации:

- диагнозы вне ТП в поликлинике — unpaid_oms_ds_pol.csv;
- диагнозы вне ТП в стационаре — unpaid_oms_ds_stac.csv;
- соответствие профиля медпомощи и специальности врача — profprvs.csv;
- кратность услуг — usl_frequency.csv;
- услуги первичного и повторного приёма — pair_of_usl_in_one_day.csv;
- услуги в основном и диспансерном реестрах — path_to_pair_d_h_usl_codes.csv;
- список кодов МО без доступа — black_list.csv.

Для перекодировок:

- классификатор медицинских специальностей V021 — v021.json;
- номенклатура медицинских услуг — medical_services.json;
- V025 классификатор целей посещения — v025.json.

5. Инструкция по установке и эксплуатации

1) Установка системы

Предварительные требования — операционная система:

- Ubuntu 20.04 LTS или выше;
- CentOS 8 или выше.

Установка веб-компонента (Laravel):

- Клонирование репозитория: `git clone <репозиторий> reestr-control;`
- Установка зависимостей PHP: `composer install --no-dev --optimize-autoloader;`
- Настройка окружения: `cp .env.example .env, php artisan key:generate;`
- Настройка файла .env (APP_NAME, DB_CONNECTION, REDIS_HOST и т.д.);
- Настройка базы данных MySQL и выполнение миграций (`php artisan migrate --force`);

- Настройка веб-сервера Nginx (виртуальный хост, fastcgi для PHP);
- Настройка прав доступа для www-data к директориям storage и bootstrap/cache.

Установка Python-компонента:

- 1. Создание виртуального окружения: `python3.8 -m venv venv`;
- 2. Установка зависимостей: `pip install -r requirements.txt`;
- 3. Настройка окружения Python (файл `.env` с параметрами БД и Redis);
- 4. Настройка базы данных PostgreSQL;
- 5. Инициализация справочников в папке `resources/`.

Установка через Docker (альтернативный способ):

- 1. Сборка контейнеров: `docker-compose build`;
- 2. Настройка `docker-compose.yml` (переменные `REESTR_DIR`, `control_MODULE`, тома `input/output`);
- 3. Запуск: `docker-compose up -d`.

2) Настройка и конфигурация

Настройка крон-заданий:

- регулярное обновление справочников НСИ (ежедневно в 2:00 и 3:00);
- обновление базы госпитализаций (ежедневно в 4:00).

Настройка мониторинга:

- логирование PHP (`nginx error.log`, `laravel.log`) и Python (`python.log`);
- проверка состояния служб (`nginx`, `php-fpm`, `postgresql`, `mysql`, `redis-server`).

3) Эксплуатация системы

Запуск валидации реестров через веб-интерфейс: авторизация, загрузка реестра по типу, просмотр статуса в разделе «Задачи», скачивание результата.

Запуск валидации через командную строку: прямой запуск Python-скрипта (`python3 errors.py`) либо запуск через Docker.

Обновление справочников НСИ:

- обновление сжатых датафреймов (`update_dir.py F014/F002/F032`);
- конвертация XML справочников (`make_df.py`).

Сбор статистических данных:

- сбор ДВН/ПВН записей (`fill_d_history.py` с параметрами периода);
- генерация отчётов по врачам и госпитализациям.

4) Обслуживание системы

Резервное копирование:

- база данных: `mysqldump` для MySQL, `pg_dump` для PostgreSQL;
- файлы и логи: архивирование `storage` и конфигурационных файлов `.env`.

Мониторинг производительности:

- контроль нагрузки CPU, памяти, дискового пространства, сетевой активности;
- проверка логов на ошибки (PHP, Python, Nginx).

Обновление системы:

- обновление кода (git pull), зависимостей PHP и Python;
- выполнение миграций базы данных, очистка кэша;
- перезапуск служб (nginx, php-fpm, redis-server).

5) Устранение неисправностей

Частые проблемы и решения:

- проблема с подключением к базе данных — проверка соединения MySQL/PostgreSQL, настроек .env;
- проблемы с правами доступа — проверка и исправление прав на storage;
- проблемы с выполнением Python-скриптов — проверка версии Python, установленных пакетов, виртуального окружения;
- проблемы с Redis — проверка работы (redis-cli ping), очистка кэша;
- проблемы с загрузкой файлов — проверка ограничений upload_max_filesize и post_max_size в PHP, свободного места на диске.

6. Контакты поддержки

Для получения технической поддержки:

- системный администратор: admin@example.com;
- разработчик Python-компонента: python-dev@example.com;
- разработчик веб-компонента: web-dev@example.com.