**Ссылка на оригинальный проект:**

Данные для подключения к ресурсам проекта:

**Development**

**Основная БД:**

где:

|  |  |
| --- | --- |
| Тип БД | PostgreSql |
| Пользователь |  |
| Пароль |  |
| Хост:порт |  |
| Наименование БД | screenrecorder |

**БД Oracle:**

|  |  |
| --- | --- |
| Тип БД | Oracle |
| Пользователь/схема |  |
| Пароль |  |
| Хост:порт |  |
| Наименование БД |  |

**Production**

**Основная БД:** postgresql://

где:

|  |  |
| --- | --- |
| Тип БД | PostgreSql |
| Пользователь |  |
| Пароль |  |
| Хост:порт |  |
| Наименование БД | screenrecorder |

**БД Oracle:**

*Будут предоставлены, когда дойдем до финального этапа разработки*

**Общие и для Dev и для Production окружения параметры**

**Samba** (локальная сеть)

|  |  |
| --- | --- |
| Домен |  |
| Пользователь |  |
| Пароль |  |

**Аудио-хранилище:**

**URL:** http://d.audiorecord:4490@nextstorage.nextcontact.ru/{year}/{month}/{day}/{session\_id}.ogg?noredirect=true

где: параметры в {фигурных скобках} должны быть проставлены в зависимости от даты создания записи и id-сессии звонка, по которому делается видеозапись.

**Локальная директория на ПК оператора, куда складываются записи:**

C:\tmp\screenrecorder

**Реализовать требуется следующий функционал:**

1) Регистрация видеозаписи в БД

2) Вывод данных в структурированном виде с возможностью их сортировки и фильтрации по отдельным полям

3) Скачивание и склейка видеозаписи с аудиорядом

4) Синхронизация списка проектов из Oracle в Postgress

5) Авторизация по LDAP

Далее, подробнее по каждому пункту:

**1. Регистрация видеозаписи в БД**

Здесь мы записываем информацию по видеозаписи в БД, по которой мы в дальнейшем сможем выкачать видеофайл. Это должен быть экшен, принимающий POST запрос, с телом типа form-data. (Здесь конечно было бы удобнее использовать json формат тела запроса, но тут мы ограничены функционалом powershell-скрипта, который собственно и будет дергать этот экшен и который отправляет данные в таком виде).

Экшен должен записывать полученные данные в таблицу **videos**

Пример curl-запроса на текущий сервис:

curl --location --request POST 'https://screenrecorder2.nextcontact.ru/record/add' \

--form 'name="test.mp4"' \

--form 'login="f.senin"' \

--form 'session\_id="test1234"' \

--form 'fullpath="C:\\Windows"' \

--form 'mac\_addr="ososososos"' \

--form 'ip\_addr="127.0.0.1"' \

--form 'get\_args="sdd=sds?sss=ppps"' \

--form 'project\_uuid="b49542db-06fb-4651-8173-b95082d026b0"'

В новом сервисе, подобный запрос должен отрабатывать в таком же виде.

Поля формы одноименны полям в БД, за исключением *project\_uuid*— здесь передается uuid проекта, по которому мы должны найти id (внешний ключ) проекта из таблицы **projects** и вставить уже его**.** Однако, бывают случаи, когда на скринрекордер, навешивают проекты не зарегистрированные в нашей системе, либо которые еще не синхронизированы с Oracle БД. В таких случаях, перед вставкой записи в таблицу **videos** — нужно создать проект в таблице **projects** с переданным *project\_uuid* и названием «Неизвестный проект». Полученный id, используем как *project\_id* для таблицы **videos**.

**2) Вывод данных в структурированном виде с возможностью их сортировки и фильтрации по отдельным полям**

Здесь нужно будет дать возможность фронтенду получать информацию в упорядоченном виде из таблицы **videos**. Данные возвращать в формате json.

Возвращать следующие поля из таблиц:

videos.login

videos.session\_id

videos.created\_at

videos.ip\_addr

videos.filename

projects.uuid

projects.name

В запросе на получение данных, нужно предусмотреть следующие возможности:

* Сортировка данных. Резализовать прием параметров, отвечающих за сортировку данных и тип сортировки (по-возрастанию, по-убыванию).

Сортировка должна быть предусмотрена для полей:

videos.login

videos.session\_id

videos.created\_at

videos.ip\_addr

projects.name

* Фильтрация (поиск) данных. Реализовать прием параметров, отвечающих за фильтрацию данных по полям:

videos.login

videos.session\_id

videos.created\_at

videos.ip\_addr

projects.name

* Пагинация. Т.к. данных в таблице достаточно много, выводить их потребуется порционно, по 50 записей на страницу. Поэтому нужно предусмотреть возможность указания сдвига, с которого будем брать данные. Также, понадобится отдельный экшен, который будет возвращать количество данных в таблице **videos**, чтобы можно было указать количество страниц с данными при построении пагинации на фронтенде.

**3) Скачивание и склейка видеозаписи с аудиорядом**

Для некоторых записей, мы храним видеоряд и аудиодорожки отдельно. Это происходит, т. к. для некоторых проектов, есть требование писать только голоса клиентов и оператора, не включая все системные и иные звуки, которые могут воспроизводиться на компьютере оператора. Отличить такие записи довольно просто: у них будет заполнено поле ***videos****.session\_id —* если оно присутствует, то это как раз та самая видеозапись, к которой нужно приклеить аудиодорожку. В иных случаях, нужно будет просто отдавать видеоряд на скачивание без всяких склеек.

Для склейки звука и видео, мы используем библиотеку ffmpeg. Рекомендую использовать ее вместе с оберткой на golang:

github.com/u2takey/ffmpeg-go

Сами звуки мы выкачиваем с отдельного хранилища. Ссылка на аудио-хранилище выглядит так:

http://d.audiorecord:4490@nextstorage.nextcontact.ru/{year}/{month}/{day}/{session\_id}.ogg?noredirect=true

где {year} — год в формате 2023

{month} — месяц в формате 01

{day} — день в формате 02

{session\_id} — id сессии из ***videos****.session\_id*

year, month и day также вытаскиваются из ***videos****.created\_at*

В итоге должна получиться ссылка, на подобии:

http://d.audiorecord:4490@nextstorage.nextcontact.ru/2023/01/02/100000000.666666.ogg?noredirect=true

Аудиозапись попадает в хранилище с задержкой, обычно 5-10 мин. В редких случаях, может отсутствовать вовсе. В текущей реализации — без аудиодорожки, скачать видео мы не даем, что, как показала практика, не правильно. Поэтому в новой версии, нужно будет отдавать файл на скачивание в независимости от наличия аудиодорожки, но уведомлять фронтенд о проблеме. Для этого можно ввести в теле ответа отдельное свойство, с указанием проблемы или заранее оговорить внутренние коды ошибки, например: 01 — не удалось найти аудифайл в хранилище, 02 — не удалось подключиться по локальной сети и т. д. (тут как больше понравится).

В текущей реализации, при выкачивании файла, мы оставляем его на сутки в отдельной директории на сервере с временными файлами, как кэш, чтобы если пользователь поделится с кем то ссылкой на аудиозапись, не тратилось время на повторное выкачивание и склейку. Однако, по прошествии суток его нужно удалять, чтобы эти файлы не засорили все пространство сервера. Так что нужно будет также предусмотреть экшен для крона, в котором бы удалялись файлы во временной директории, созданные более суток назад.

**4) Синхронизация списка проектов из Oracle в Postgress**

Нужно сделать экшен для крона, который бы заполнял таблицу **projects** в postgres. Оригинальные данные хранятся в oracle, в таблице **PROJECTS**. Нужно забирать из нее данные из столбцов ID и TITLE и обновлять **projects**.name у уже существующих записей, а также вставлять проекты, которых еще нет в таблице.

**5) Авторизация по LDAP**

Необходимо ограничить доступ к сервису только пользователями, занесенными в Active Directory. Какая-то строгая ролевая модель, которая разграничивала бы доступы к отдельным частям сервиса — не нужна, достаточно просто проверки наличия пользователя в Active Directory.

Если будет время и желание, можно реализовать доступы к экшенам по токену, однако токеном не нужно закрывать экшены на скачивание файлов и экшены кронов.