MediaCMS Документация администратора

1. Добро пожаловать

Эта страница создана для администраторов MediaCMS, которые отвечают за настройку программного обеспечения, его обслуживание и внесение изменений.

2. Установка сервера

Основные зависимости: Python3, Django3, Celery, PostgreSQL, Redis, ffmpeg. Любая система, в которой могут быть установлены эти зависимости, может запустить MediaCMS. Но мы настоятельно рекомендуем устанавливать на Linux Ubuntu (проверено на версиях 20, 22).

Установка в системе Ubuntu с установленной утилитой git должна быть завершена за несколько минут с помощью следующих шагов. Убедитесь, что вы запускаете его как пользователь root, на чистой системе, так как автоматический скрипт установит и настроит следующие службы: Celery/PostgreSQL/Redis/Nginx и перезапишет любые существующие настройки.

Автоматизированный скрипт — протестирован на Ubuntu 20, Ubuntu 22 и Debian Buster

```
mkdir /home/mediacms.io && cd /home/mediacms.io/
git clone https://github.com/mediacms-io/mediacms
cd /home/mediacms.io/mediacms/ && bash ./install.sh
```

Скрипт спросит, есть ли у вас URL, где вы хотите развернуть MediaCMS, в противном случае он будет использовать localhost. Если вы предоставите URL, он будет использовать службу Let's Encrypt для установки действительного сертификата ssl.

Обновлять Если вы использовали описанный выше способ установки MediaCMS, выполните обновление следующим образом:

```
cd /home/mediacms.io/mediacms # enter mediacms directory
source  /home/mediacms.io/bin/activate # use virtualenv
git pull # update code
pip install -r requirements.txt -U # run pip install to update
python manage.py migrate # run Django migrations
sudo systemctl restart mediacms celery_long celery_short # restart services
```

Обновление с версии 2 до версии 3 Версия 3 использует Django 4 и Celery 5 и требует последней версии Python 3.x. Если вы обновляете более старую версию, убедитесь, что Python обновлен. Версия 2 может работать на Python 3.6, но для версии 3 требуется Python3.8 и выше. Синтаксис для запуска Celery также изменился, поэтому вам нужно скопировать файлы systemctl, связанные с celery, и перезапустить

```
# cp deploy/local_install/celery_long.service
```

```
/etc/systemd/system/celery_long.service
# cp deploy/local_install/celery_short.service
/etc/systemd/system/celery_short.service
# cp deploy/local_install/celery_beat.service
/etc/systemd/system/celery_beat.service
# systemctl daemon-reload
# systemctl start celery_long celery_short celery_beat
```

Конфигурация

Ознакомьтесь с разделом конфигурации здесь.

Обслуживание

Резервную копию базы данных можно создать с помощью pg_dump и media_files в /home/mediacms.io/mediacms/media_files, включая исходные файлы и закодированные/транскодированные версии.

3. Установка докера

Установка

Установите последнюю версию Docker и Docker Compose .

Для систем Ubuntu 20/22 это:

```
curl -fsSL https://get.docker.com -o get-docker.sh
sudo sh get-docker.sh
sudo curl -L
"https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$
(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
sudo chmod +x /usr/local/bin/docker-compose
```

Затем запустите как root

```
git clone https://github.com/mediacms-io/mediacms
cd mediacms
```

По умолчанию MediaCMS обслуживается на всех доступных IP-адресах сервера (включая localhost). Если вы хотите изучить больше вариантов (включая настройку https с сертификатом letsencrypt), ознакомьтесь с разделом развертывания Docker для различных настроек docker-compose, которые можно использовать.

RUN

```
docker-compose up
```

Это загрузит все связанные с MediaCMS образы Docker и запустит все контейнеры. После завершения MediaCMS будет установлен и доступен на http://localhost или http://ip

Пользователь admin был создан со случайным паролем, вы должны увидеть его в конце контейнера миграций, например

```
migrations_1     | Created admin user with password: gwg1clfkwf
```

или если вы установили переменную ADMIN_PASSWORD в файле docker-compose, который вы использовали (пример docker-compose.yaml), эта переменная будет установлена как пароль пользователя admin

Обновлять

Получите последний образ MediaCMS и остановите/запустите контейнеры

```
cd /path/to/mediacms/installation
docker pull mediacms/mediacms
docker-compose down
docker-compose up
```

Обновление с версии 2 до версии 3

Версия 3 использует Python 3.11 и PostgreSQL 15. Если вы обновляете более старую версию, которая использовала PostgreSQL 13, автоматическое обновление не будет работать, так как при запуске контейнера PostgreSQL вы получите следующее сообщение:

```
db_1             | 2023-06-27 11:07:42.959 UTC [1] FATAL:  database files
are incompatible with server
db_1             | 2023-06-27 11:07:42.959 UTC [1] DETAIL:  The data
directory was initialized by PostgreSQL version 13, which is not compatible
with this version 15.2.
```

На этом этапе есть два варианта: либо отредактировать файл Docker Compose и использовать существующий образ postgres:13, либо выполнить миграцию с postgresql 13 на версию 15. Дополнительные заметки по #749

Конфигурация

Ознакомьтесь с документацией по конфигурации здесь.

Обслуживание

База данных хранится в ../postgres_data/, а media_files в media_files/

4. Варианты развертывания Docker

Образ mediacms создан для использования supervisord в качестве основного процесса, который управляет одной или несколькими службами, необходимыми для запуска mediacms. Мы можем переключать службы, которые запускаются в данном контейнере, устанавливая переменные среды ниже на yes или no:

- ENABLE_UWSGI
- ENABLE_NGINX
- ENABLE_CELERY_BEAT
- ENABLE_CELERY_SHORT
- ENABLE_CELERY_LONG
- ENABLE_MIGRATIONS

По умолчанию все эти службы включены, но для создания масштабируемого развертывания некоторые из них можно отключить, разделив службу на более мелкие службы.

Также см. `Dockerfile` для других переменных среды, которые вы, возможно, захотите переопределить. Настройки приложения, например, FRONTEND_HOST также можно переопределить, обновив `deploy/docker/local_settings.py` файл.

См. примеры развертываний в разделах ниже. Эти примеры развертываний были протестированы при docker-compose version 1.27.4запуске наDocker version 19.03.13

Для запуска обновите указанные выше конфигурации, если необходимо, соберите образ, запустив docker-compose build, затем запуститеdocker-compose run

Простое развертывание, доступ к которому осуществляется по адресу http://localhost

Основной контейнер запускает миграции, mediacms_web, celery_beat, celery_workers (службы celery_short и celery_long), доступные на порту 80, поддерживаемом базой данных Redis и Postgres.

FRONTEND_HOST `deploy/docker/local_settings.py` настроен как http://localhost на хост-машине Docker.

Сервер с сертификатом ssl через службу letsencrypt, доступ к которому осуществляется по адресу https://my_domain.com

Прежде чем попробовать это, убедитесь, что IP-адрес указывает на my_domain.com.

При этом методе используется такое развертывание .

Отредактируйте этот файл и установите `VIRTUAL_HOST my_domain.com`, `LETSENCRYPT_HOSTmy_domain.com` и ваш адрес электронной почты на`LETSENCRYPT_EMAIL`

Отредактируйте `deploy/docker/local_settings.py` и установите `https://my_domain.com` как `FRONTEND_HOST`

Теперь запустите docker-compose -f docker-compose-letsencrypt.yaml up, после завершения установки вы сможете получить доступ к https://my_domain.com, используя действительный сертификат Letsencrypt!

Расширенное развертывание, доступ к которому осуществляется по адресу http://localhost:8000

Здесь мы можем запустить 1 экземпляр mediacms_web с FRONTEND_HOST, `deploy/docker/local_settings.py` настроенным как http://localhost:8000 . Он загружается одним экземпляром миграции и поддерживается одним экземпляром celery_beat и 1 или несколькими экземплярами celery_worker. Контейнеры Redis и postgres также используются для сохранения. Клиенты могут получить доступ к службе по адресу http://localhost:8000 на хост-машине docker. Это похоже на это развертывание с portопределенным в FRONTEND_HOST.

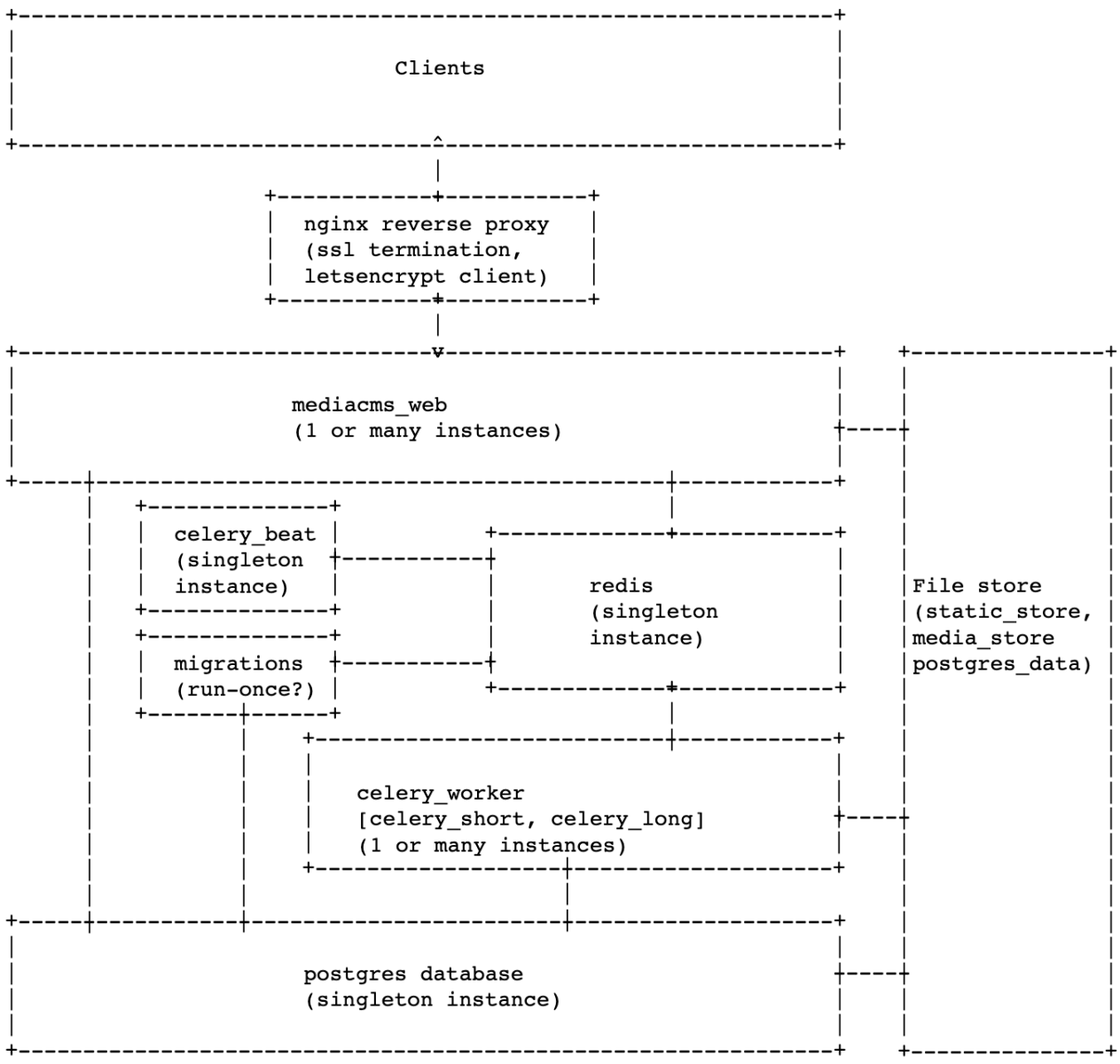
Расширенное развертывание с обратным прокси-сервером, доступ к которому осуществляется по адресу http://mediacms.io

Здесь мы можем использовать jwilder/nginx-proxyобратный прокси-сервер для одного или нескольких экземпляров mediacms_web, поддерживаемых другими службами, как упоминалось в предыдущем развертывании. FRONTEND_HOST в deploy/docker/local_settings.pyнастроен как http://mediacms.io , nginx-proxy имеет открытый порт 80. Клиенты могут получить доступ к службе по адресу http://mediacms.io (предполагая, что DNS или файл hosts настроены правильно, чтобы указывать на IP экземпляра nginx-proxy). Это похоже на это развертывание .

Расширенное развертывание с обратным прокси-сервером, доступ к которому осуществляется по адресу https://localhost Обратный прокси-сервер (jwilder/nginx-proxy) можно настроить для обеспечения SSL-терминации с использованием самоподписанных сертификатов, letsencrypt или сертификатов, подписанных CA (см.: https://hub.docker.com/r/jwilder/nginx-proxy или пример LetsEncrypt). В этом случае FRONTEND_HOST следует установить на https://mediacms.io . Это похоже на это развертывание .

Масштабируемая архитектура развертывания (Docker, Swarm, Kubernetes)

Архитектура ниже обобщает все сценарии развертывания, описанные выше, и обеспечивает концептуальный дизайн для других развертываний на основе kubernetes и docker swarm. Она обеспечивает горизонтальную масштабируемость за счет использования нескольких экземпляров mediacms_web и celery_workers. Для крупных развертываний могут быть приняты управляемые postgres, redis и storage.



5. Конфигурация

На сайте доступно несколько опций `cms/settings.py`, большинство из того, что разрешено или должно быть запрещено, описано там.

Рекомендуется переопределить любой из них, добавив его в local_settings.py.

В случае установки на один сервер добавьте в cms/local_settings.py.

В случае установки docker compose добавьте в deploy/docker/local_settings.py. Это автоматически перезапишет cms/local_settings.py.

Для вступления любых изменений в силу потребуется перезапуск MediaCMS.

Установка на одном сервере: отредактируйте cms/local_settings.py, внесите изменения и

перезапустите MediaCMS

```
#systemctl restart mediacms
```

Установка Docker Compose: отредактируйте deploy/docker/local_settings.py, внесите изменения и перезапустите контейнеры MediaCMS

```
#docker-compose restart web celery_worker celery_beat
```

5.1 Изменить логотип портала

Установите новый файл svg для белой темы (`static/images/logo_dark.svg`) или темной темы (`static/images/logo_light.svg`)

5.2 Установить глобальное название портала

набор `PORTAL_NAME`, например

```
PORTAL_NAME = 'my awesome portal'
```

5.3 Контроль над тем, кто может добавлять медиафайлы

По умолчанию `CAN_ADD_MEDIA = «all»` означает, что все зарегистрированные пользователи могут добавлять медиа. Другие допустимые варианты:

- **email_verified**, пользователь должен не только зарегистрировать учетную запись, но и подтвердить адрес электронной почты (нажав на ссылку, отправленную при регистрации). Очевидно, конфигурация электронной почты должна работать, иначе пользователи не будут получать электронные письма.
- **advancedUser**, только пользователи, отмеченные как продвинутые пользователи, могут добавлять медиа. Администраторы или менеджеры MediaCMS могут сделать пользователей продвинутыми пользователями, отредактировав их профиль и выбрав advancedUser.

5.4 Каков рабочий процесс портала?

Переменная `PORTAL_WORKFLOW` определяет, что происходит с вновь загруженными медиафайлами, независимо от того, отображаются ли они в списках (на странице индекса или в результатах поиска).

- **public** — опция по умолчанию, которая означает, что медиа может отображаться в списках. Если тип медиа — видео, оно появится, как минимум, после успешного завершения задачи, которая создает закодированную версию файла. Для других типов файлов, таких как изображение/аудио, они появляются мгновенно
- **private** означает, что недавно загруженный контент является частным - его могут

видеть только пользователи или редакторы, менеджеры и администраторы MediaCMS. Они также могут установить статус public или unlisted

- **unlisted** означает, что элементы не перечислены. Однако если пользователь посещает URL-адрес не перечисленного медиа, он будет показан (в отличие от приватного)

5.5 Показать или скрыть кнопку «Войти»

для отображения кнопки:

```
LOGIN_ALLOWED = True
```

чтобы скрыть кнопку:

```
LOGIN_ALLOWED = False
```

5.6 Показать или скрыть кнопку «Регистрация»

для отображения кнопки:

```
REGISTER_ALLOWED = True
```

чтобы скрыть кнопку:

```
REGISTER_ALLOWED = False
```

5.7 Показать или скрыть кнопку загрузки медиафайлов

Показать:

```
UPLOAD_MEDIA_ALLOWED = True
```

Чтобы скрыть:

```
UPLOAD_MEDIA_ALLOWED = False
```

5.8 Показать или скрыть кнопки действий (нравится/не нравится/пожаловаться)

Внесите изменения (True/False) в любой из следующих пунктов:

```
- CAN_LIKE_MEDIA = True  # whether the like media appears
- CAN_DISLIKE_MEDIA = True  # whether the dislike media appears
- CAN_REPORT_MEDIA = True  # whether the report media appears
```

```
- CAN_SHARE_MEDIA = True  # whether the share media appears
```

5.9 Показать или скрыть опцию загрузки на носителе

Редактировать templates/config/installation/features.htmlи установить

```
download: false
```

5.10 Автоматически скрывать медиа при получении жалобы

установите небольшое число для переменной, REPORTED_TIMES_THRESHOLD например

```
REPORTED_TIMES_THRESHOLD = 2
```

как только лимит достигнут, медиафайлы переходят в режим приватности, а администраторам отправляется электронное письмо.

5.11 Установка пользовательского сообщения на странице загрузки медиафайлов

это сообщение появится под формой перетаскивания медиафайлов

```
PRE_UPLOAD_MEDIA_MESSAGE = 'custom message'
```

5.12 Установка настроек электронной почты

Установите правильные настройки для каждого провайдера

```
DEFAULT_FROM_EMAIL = 'info@mediacms.io'
EMAIL_HOST_PASSWORD = 'xyz'
EMAIL_HOST_USER = 'info@mediacms.io'
EMAIL_USE_TLS = True
SERVER_EMAIL = DEFAULT_FROM_EMAIL
EMAIL_HOST = 'mediacms.io'
EMAIL_PORT = 587
ADMIN_EMAIL_LIST = ['info@mediacms.io']
```

5.13 Запретить регистрацию пользователей из определенных доменов

Установите домены, которые недопустимы для регистрации с помощью этой переменной:

```
RESTRICTED_DOMAINS_FOR_USER_REGISTRATION = [
    'xxx.com', 'emaildomainwhatever.com']
```

В качестве альтернативы разрешите регистрацию только разрешенным доменам. Это может быть полезно, если вы используете mediacms как частную службу в организации и хотите предоставить бесплатную регистрацию для тех, кто в организации, но запретить регистрацию для всех других доменов. Установка этого параметра запрещает регистрацию всем доменам, НЕ входящим в список. По умолчанию это пустой список, который игнорируется. Чтобы отключить, установите пустой список.

```
ALLOWED_DOMAINS_FOR_USER_REGISTRATION = [
    "private.com",
    "vod.private.com",
    "my.favorite.domain",
    "test.private.com"]
```

5.14 Требовать проверки редакторами/менеджерами/администраторами MediaCMS

установить значение

```
MEDIA_IS_REVIEWED = False
```

Теперь все загруженные медиа должны быть проверены, прежде чем они появятся в списках. Редакторы/менеджеры/администраторы MediaCMS могут посещать страницу медиа и редактировать ее, где они могут увидеть опцию пометки медиа как проверенных. По умолчанию это установлено на True, поэтому все медиа не требуют проверки

5.15 Укажите максимальное количество медиафайлов для списка воспроизведения

установить другой порог для переменной MAX_MEDIA_PER_PLAYLIST

например

```
MAX_MEDIA_PER_PLAYLIST = 14
```

5.16 Укажите максимальный размер медиафайла, который можно загрузить

изменять UPLOAD_MAX_SIZE.

по умолчанию 4 ГБ

```
UPLOAD_MAX_SIZE = 800 * 1024 * 1000 * 5
```

5.17 Укажите максимальный размер комментариев

изменять MAX_CHARS_FOR_COMMENT

по умолчанию:

```
MAX_CHARS_FOR_COMMENT = 10000
```

5.18 Сколько файлов загружать параллельно

установите другой порог по UPLOAD_MAX_FILES_NUMBER умолчанию:

```
UPLOAD_MAX_FILES_NUMBER = 100
```

5.18 заставить пользователей подтвердить свой адрес электронной почты при регистрации

опция по умолчанию для подтверждения электронной почты необязательна. Установите ее как обязательную, чтобы заставить пользователей подтвердить свой адрес электронной почты перед входом в систему

```
ACCOUNT_EMAIL_VERIFICATION = 'optional'
```

5.20 Ограничение количества попыток входа в учетную запись

после достижения этого числа

```
ACCOUNT_LOGIN_ATTEMPTS_LIMIT = 20
```

устанавливает тайм-аут (в секундах)

```
ACCOUNT_LOGIN_ATTEMPTS_TIMEOUT = 5
```

5.21 Запретить регистрацию пользователя

установите следующую переменную на False

```
USERS_CAN_SELF_REGISTER = True
```

5.22 Настройка уведомлений

Реализованные глобальные уведомления контролируются следующими параметрами:

```
USERS_NOTIFICATIONS = {
    'MEDIA_ADDED': True,
}
```

Если вы хотите отключить уведомления о новых медиа, установите значение False.

Администраторы также получают уведомления о различных событиях. Чтобы отключить эти уведомления, установите для любого из следующих параметров значение False.

```
ADMINS_NOTIFICATIONS = {
    'NEW_USER': True,
    'MEDIA_ADDED': True,
    'MEDIA_REPORTED': True,
}
```

- NEW_USER: добавлен новый пользователь
- MEDIA_ADDED: добавлен медиафайл
- MEDIA_REPORTED: отчет для СМИ был поражен

5.23 Настройте доступ к медиа только для участников

Сделайте рабочий процесс портала общедоступным, но в то же время настройте его `GLOBAL_LOGIN_REQUIRED = True` так, чтобы видеть контент могли только вошедшие в систему пользователи. Вы можете либо указать `REGISTER_ALLOWED = False`, хотите ли вы добавлять участников самостоятельно, либо проверить параметры в «настройках django-allauth», которые влияют на регистрацию в `cms/settings.py`. Например, установить только приглашение на портал или сделать подтверждение по электронной почте обязательным, чтобы вы могли контролировать, кто регистрируется.

5.24 Включить карту сайта

Включать или нет генерацию файла карты сайта по адресу http://your_installation/sitemap.xml (по умолчанию: False)

```
GENERATE_SITEMAP = False
```

5.25 Управление тем, кто может добавлять комментарии

По умолчанию `CAN_COMMENT = «all»` означает, что все зарегистрированные пользователи могут добавлять комментарии. Другие допустимые варианты:

- **email_verified**, пользователь должен не только зарегистрировать учетную запись, но и подтвердить адрес электронной почты (нажав на ссылку, отправленную при регистрации). Очевидно, конфигурация электронной почты должна работать, иначе пользователи не будут получать электронные письма.

- **advancedUser**, только пользователи, отмеченные как продвинутые пользователи, могут добавлять комментарии. Администраторы или менеджеры MediaCMS могут сделать пользователей продвинутыми пользователями, отредактировав их профиль и выбрав advancedUser.

6. Управление страницами

быть написанным

7. Панель администратора Django

8. Рабочий процесс портала

Кто может публиковать контент, как контент отображается в публичных списках. Разница между статусами (частный, не включенный в список, публичный)

9. О ролях пользователей

Различия между менеджером MediaCMS, редактором MediaCMS и вошедшим в систему пользователем

10. Добавление языков для титров и субтитров

быть написанным

11. Добавить/удалить категории и теги

Через раздел администратора - http://your_installation/admin/

12. Транскодирование видео

Добавить/удалить разрешения и профили, изменив таблицу базы данных Encode profilesчерез https://your_installation/admin/files/encodeprofile/

Например, Activeсостояние любого профиля можно переключать, включая или отключая его.

13. Как добавить статическую страницу на боковую панель

1. Создайте свою html-страницу в templates/cms/

например, дублировать и переименовать about.html

```
sudo cp templates/cms/about.html templates/cms/volunteer.html
```

2. Создайте свой CSS-файл в static/css/

```
touch static/css/volunteer.css
```

3. В вашем HTML-файле обновите заголовок блока meta, чтобы отразить вашу новую страницу.

```
{% block headermeta %}
<meta property="og:title" content="Volunteer - {{PORTAL_NAME}}">
<meta property="og:type" content="website">
<meta property="og:description" content="">
<meta name="twitter:card" content="summary">
<script type="application/ld+json">
{
    "@context": "https://schema.org",
    "@type": "BreadcrumbList",
    "itemListElement": [{
        "@type": "ListItem",
        "position": 1,
        "name": "{{PORTAL_NAME}}",
        "item": {
            "@type": "WebPage",
            "@id": "{{FRONTEND_HOST}}"
        }
    },
    {
        "@type": "ListItem",
        "position": 2,
        "name": "Volunteer",
        "item": {
            "@type": "VolunteerPage",
            "@id": "{{FRONTEND_HOST}}/volunteer"
        }
    }]
}
```

```
</script>
<link href="{% static "css/volunteer.css" %}" rel="stylesheet"/>
{% endblock headermeta %}
```

4. В вашем HTML-файле обновите блок innercontent, чтобы отразить ваше фактическое содержимое.

Пишите, что хотите.

5. В вашем CSS-файле пропишите соответствующие стили для вашего HTML-файла.

Пишите, что хотите.

6. Добавьте свое представление в files/views.py

```
def volunteer(request):
    """Volunteer view"""
    context = {}
    return render(request, "cms/volunteer.html", context)
```

7. Добавьте свой шаблон URL в files/urls.py

```
urlpatterns = [
    url(r"^$", views.index),
    url(r"^about", views.about, name="about"),
    url(r"^volunteer", views.volunteer, name="volunteer"),
```

8. Добавьте свою страницу на левую боковую панель

Чтобы добавить ссылку на свою страницу в качестве пункта меню на левой боковой панели, добавьте следующий код после последней строки в _commons.js

```
/* Checks that a given selector has loaded. */
const checkElement = async selector => {
    while ( document.querySelector(selector) === null) {
      await new Promise( resolve =>  requestAnimationFrame(resolve) )
    }
    return document.querySelector(selector);
  };

/* Checks that sidebar nav menu has loaded, then adds menu item. */
checkElement('.nav-menu')
.then((element) => {
     (function(){
```

```
        var a = document.createElement('a');
        a.href = "/volunteer";
        a.title = "Volunteer";

        var s = document.createElement('span');
        s.className = "menu-item-icon";

        var icon = document.createElement('i');
        icon.className = "material-icons";
        icon.setAttribute("data-icon", "people");

        s.appendChild(icon);
        a.appendChild(s);

        var linkText = document.createTextNode("Volunteer");
        var t = document.createElement('span');

        t.appendChild(linkText);
        a.appendChild(t);

        var listItem = document.createElement('li');
        listItem.className = "link-item";
        listItem.appendChild(a);

        //if signed out use 3rd nav-menu
        var elem = document.querySelector(".nav-menu:nth-child(3) nav ul");
        var loc = elem.innerText;
        if (loc.includes("About")){
          elem.insertBefore(listItem, elem.children[2]);
        } else { //if signed in use 4th nav-menu
          elem = document.querySelector(".nav-menu:nth-child(4) nav ul");
          elem.insertBefore(listItem, elem.children[2]);
        }
    })();
});
```

9. Перезапустите веб-сервер mediacms.

В докере:

```
sudo docker stop mediacms_web_1 && sudo docker start mediacms_web_1
```

В противном случае

```
sudo systemctl restart mediacms
```

14. Добавьте Google Аналитику

Инструкции предоставлены @alberto98fx

Создайте файл:

```
touch $DIR/mediacms/templates/tracking.html
```

Добавьте скрипт Gtag/Analytics

Внутри `$DIR/mediacms/templates/root.html` вы увидите такой файл:

```
<head>
    {% block head %}

        <title>{% block headtitle %}{{PORTAL_NAME}}{% endblock headtitle
%}</title>

        {% include "common/head-meta.html" %}

        {% block headermeta %}

        <meta property="og:title" content="{{PORTAL_NAME}}">
        <meta property="og:type" content="website">

        {%endblock headermeta %}

        {% block externallinks %}{% endblock externallinks %}

        {% include "common/head-links.html" %}

        {% block topimports %}{%endblock topimports %}

        {% include "config/index.html" %}

    {% endblock head %}

</head>
```

Добавить `{% include «tracking.html» %}` в конце внутри раздела<head>

Если вы используете Docker и не смонтировали весь каталог, вам необходимо привязать новый том:

```
    web:
    image: mediacms/mediacms:latest
    restart: unless-stopped
    ports:
      - "80:80"
    deploy:
```

```
      replicas: 1
    volumes:
      - ./templates/root.html:/home/mediacms.io/mediacms/templates/root.html
      -
./templates/tracking.html://home/mediacms.io/mediacms/templates/tracking.htm
l
<code>
=====15. Устранение неполадок с электронной почтой=====
В разделе «Конфигурация» этого руководства мы увидели, как редактировать настройки электронной почты. Если мы все еще не можем получать электронную почту от MediaCMS, следующее может помочь нам отладить проблему — в большинстве случаев это проблема установки правильного имени пользователя, пароля или параметра TLS

Войдите в оболочку Django, например, если вы используете установку с одним сервером:
<code>
source  /home/mediacms.io/bin/activate
python manage.py shell
```

и внутри оболочки

```
from django.core.mail import EmailMessage
from django.conf import settings

settings.EMAIL_BACKEND = 'django.core.mail.backends.smtp.EmailBackend'

email = EmailMessage(
    'title',
    'msg',
    settings.DEFAULT_FROM_EMAIL,
    ['recipient@email.com'],
)
email.send(fail_silently=False)
```

У вас есть возможность либо получить письмо (в этом случае оно будет отправлено на адрес receiveer@email.com), либо вы увидите ошибку. Например, при указании неправильного пароля для моей учетной записи Gmail я получаю

```
SMTPAuthenticationError: (535, b'5.7.8 Username and Password not accepted.
Learn more at\n5.7.8  https://support.google.com/mail/?p=BadCredentials
d4sm12687785wrc.34 - gsmtp')
```

16. Часто задаваемые вопросы

Видео воспроизводится, но эскизы предварительного просмотра не отображаются для больших видеофайлов.

Скорее всего, файл спрайтов был создан неправильно. Вывод функции

files.tasks.produce_sprite_from_video() в этом случае будет примерно таким

```
convert-im6.q16: width or height exceeds limit `/tmp/img001.jpg' @
error/cache.c/OpenPixelCache/3912.
```

Решение: отредактируйте файл /etc/ImageMagick-6/policy.xmlи установите большие значения для строк, содержащих ширину и высоту. Например

```
  <policy domain="resource" name="height" value="16000KP"/>
  <policy domain="resource" name="width" value="16000KP"/>
```

Недавно добавленные видеофайлы теперь смогут создавать файлы спрайтов, необходимые для предпросмотра миниатюр. Чтобы повторно запустить эту задачу на существующих видео, войдите в оболочку Django

```
root@8433f923ccf5:/home/mediacms.io/mediacms# source
/home/mediacms.io/bin/activate
root@8433f923ccf5:/home/mediacms.io/mediacms# python manage.py shell
Python 3.8.14 (default, Sep 13 2022, 02:23:58)
```

и беги

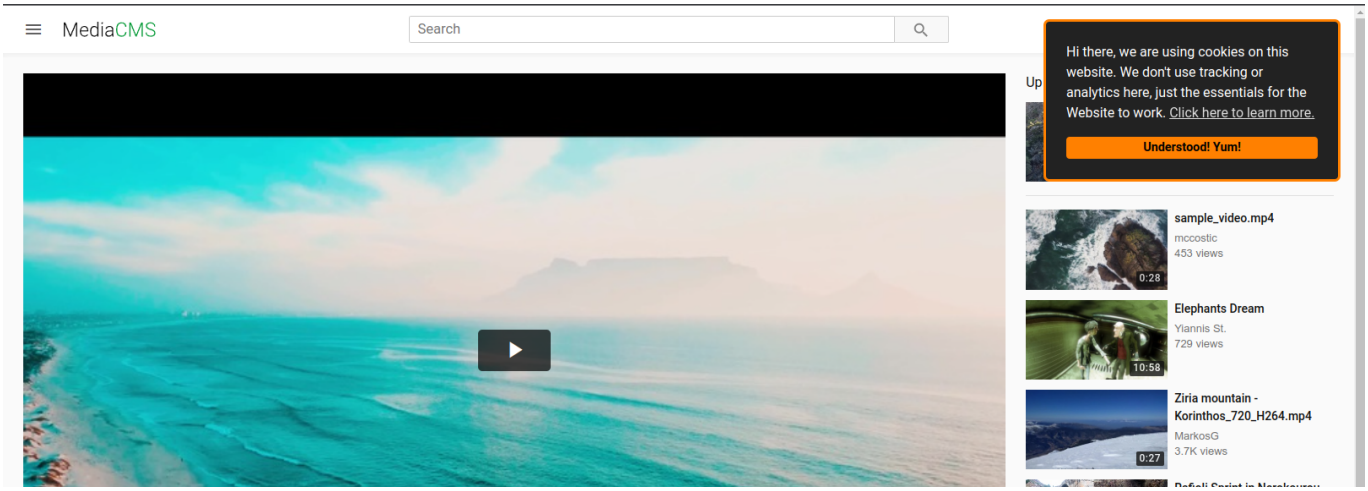
```
In [1]: from files.models import Media
In [2]: from files.tasks import produce_sprite_from_video

In [3]: for media in Media.objects.filter(media_type='video', sprites=''):
   ...:     produce_sprite_from_video(media.friendly_token)
```

это позволит заново создать спрайты для видео, которые не удалось выполнить.

17. Код согласия на использование файлов cookie

В файле `templates/components/header.html` вы можете найти простой код согласия на использование файлов cookie. Он прокомментирован, поэтому вам нужно удалить строки {% comment %}и {% endcomment %}, чтобы включить его. Или вы можете заменить эту часть собственным кодом, который обрабатывает баннеры согласия на использование файлов cookie.



18. Отключить кодирование и показывать только оригинальный файл

При загрузке видео они кодируются в несколько разрешений, эта процедура называется транскодированием. Иногда это не нужно, и вам нужно только показать исходный файл, например, когда MediaCMS работает на сервере с низкими возможностями. Чтобы добиться этого, отредактируйте settings.py и установите

```
DO_NOT_TRANSCODE_VIDEO = True
```

Это отключит процесс перекодирования и будет показан только исходный файл. Обратите внимание, что это также отключит создание файла спрайтов, поэтому у вас не будет предварительных миниатюр на видеоплеере.

19. Закругленные углы на видео

По умолчанию видеоплеер и элементы мультимедиа теперь имеют закругленные углы на больших экранах (не на мобильных устройствах). Если вам не нравится это изменение, удалите border-radiusдобавленное в следующих файлах:

```
frontend/src/static/css/_extra.css
frontend/src/static/js/components/list-item/Item.scss
frontend/src/static/js/components/media-page/MediaPage.scss
```

Теперь вам придется перезапустить сборку фронтенда, чтобы увидеть изменения (проверьте docs/dev_exp.md)

20. Переводы

20.1 Установить язык по умолчанию

По умолчанию MediaCMS доступен на нескольких языках. Чтобы установить язык по умолчанию, отредактируйте `settings.py` и установите LANGUAGE_CODE на код одного из языков.

20.2 Удалить существующие языки

Чтобы ограничить количество языков, которые отображаются как доступные, удалите их из списка ЯЗЫКИ `settings.py` или прокомментируйте их. Показывается только то, что есть.

20.3 Улучшить существующий перевод

Чтобы внести улучшения в существующий переведенный контент на языке, на который уже выполнен перевод, проверьте язык по кодовому названию `files/frontend-translations/` и отредактируйте соответствующий файл.

20.4 Добавить больше контента к существующему переводу

Не весь текст переведен, поэтому в любой момент вы можете обнаружить, что отсутствуют строки, которые необходимо добавить в перевод. Идея здесь в том, что

а) вы сделали текст переводимым в коде б) вы добавляете переведенную строку

Для а) вам нужно посмотреть, находится ли строка для перевода в каталоге frontend (приложение React) или в шаблонах Django. Для обоих вариантов есть примеры.

1. шаблоны Django, которые находятся в templates/ dir. Посмотрите templates/cms/about.htmlна пример того, как это делается
2. код фронтенда (React), посмотрите, как translateStringон используется вfrontend

После того, как строка отмечена как переводимая, добавьте ее в files/frontend-translations/en.pyfirst, а затем запустите

```
python manage.py process_translations
```

Чтобы заполнить строку на всех языках. НИКАКИЕ PR не будут приняты, если эта процедура не будет соблюдена. Вам не нужно переводить строку на все поддерживаемые языки, но команда должна быть запущена и заполнить существующие словари новыми строками для всех языков. Это гарантирует, что не будет пропущенных строк для перевода на любой язык.

После запуска этой команды переведите строку на нужный вам язык. Если переводимая строка находится в шаблонах Django, вам не нужно перестраивать frontend. Если изменение находится во frontend, вам придется перестроить, чтобы увидеть изменения. Команда `Makefile make build-frontend` может помочь в этом.

20.5 Добавить новый язык и перевести

Чтобы добавить новый язык: добавьте язык в settings.py, затем добавьте файл в files/frontend-translations/. Обязательно скопируйте начальные строки, скопировав files/frontend-translations/en.pyв него.

21. Как изменить видеокадры на видео

По умолчанию при просмотре видео вы можете навести курсор и увидеть небольшие изображения, называемые спрайтами, которые извлекаются каждые 10 секунд видео. Вы можете изменить это число на что-то меньшее, выполнив следующее:

- отредактируйте ./frontend/src/static/js/components/media-viewer/VideoViewer/index.js и измените seconds: 10 значение на желаемое, например, 2.
- отредактируйте settings.py и установите то же число для значения SPRITE_NUM_SECS
- теперь вам придется пересобрать фронтенд: самый простой способ — запустить make build-frontend, для чего требуется Docker

После этого для вновь загруженных видео будут сгенерированы спрайты с новым количеством секунд.