

Вебсайт <https://semiautoqatz.pcvr-stg-api.smartex-it.com>

Это система для хранения, поиска и управления фото-, видео-, аудио контентом, презентациями и другими файлами в больших цифровых архивах

Логин	Пароль
semiautoqatz@tz.com	Pa\$\$w0rd!

Swagger

<https://semiautoqatz.pcvr-stg-api.smartex-it.com>

P.S. следует учитывать, что данная документация сгенерирована автоматически, в ней информация может быть неточная или неполная. Для решения тестового задания предлагаем авторизоваться в системе через браузер и посмотреть как UI работает с API.

Формат предоставляемого варианта решения

Вариант решения должен быть предоставлен в виде репозитория в любой предпочтительной для вас системе контроля версий.

Примечание: **все** тестовые задания будут рассмотрены, вне зависимости от количества выполненных шагов и условий. Если в процессе решения возникнут сложности или нехватка времени - всё равно отправьте свой вариант на том этапе, на котором вы закончили.

Стек, на котором будет решено задание, не имеет значения, однако просьба: в файле README.md обосновать свой выбор.

P.S. периодически стенд может быть недоступен на 5-10 минут из-за того, что он прерываемый

Задание №1 (Основное)

Реализовать автотест (API), покрывающий кейс авторизации в системе.

При реализации использовать метод

[*/api-token-auth/*](#)

Задание №2 (Основное)

Написать автотест (API), проверяющий импорт файла в систему. При реализации использовать метод

[api/v1/content_import/assets/upload](#)

Условие №1

В рамках автотеста - необходимо загрузить в систему как минимум 10 файлов (JPG/PNG/MP3/PDF)

Условие №2

Среди набора тестовых данных должны быть хотя бы один файл размером МЕНЕЕ одного мегабайта

Условие №3

Среди набора тестовых данных должен быть хотя бы один файл размером более 15 мегабайт

Условие №4

Файлы >5мб должны импортироваться в систему минимум в два запроса (chunk upload), с соответствующими значениями заголовка Content-Range

Условие №5

Убедиться что файлы импортированные в систему содержат оригинал / preview / thumbnail
Для этого можно использовать запрос (на UI можно посмотреть как он устроен, произведя поиск):

[/api/v1/mutual_integration/find_assets/search/](#)

Обратите внимание, что импорт файла производится непосредственно в ранее созданный актив (объект).

Создать актив можно используя метод

[api/v1/content_import/assets/create](#)

:

api/v1/content_import/assets/create Минимальное тело запроса для создания актива в системе <i>В качестве наименования объекта - используйте свою фамилию</i>	<pre>{ "content_type": "base", "external_id": "Уникальная_строка", "values": { "title": "Наименование объекта", "description": "Описание объекта" } }</pre>
--	---

P.S.2 Размер chunk не может быть менее 5 мегабайт

Задание №3

Реализовать интерфейс или пример класса, который будет взаимодействовать с базой данных, и проверять данные в слое DB.

Интерфейс должен содержать в себе методы:

Соединения с базой данных

Запроса данных из БД (SELECT)

Пример запроса:

```
select
    *
from
    semiautoqatz.content_asset ca;
```

Условие №1

Соединение с базой данных должно соответствовать шаблону проектирования Singleton

Задание №4

Реализовать отчёт о прохождении автотестов. Отчёт о тестах должен быть понятен сотрудникам, не имеющим отношения к отделу QA. В отчете должны содержаться:

1. Запрос
2. Заголовки запроса
3. Параметры запроса
4. Результат выполнения запроса
5. Статус шагов
6. Статус кейсов

Задание №5

Подготовить автотесты к запуску в CI. Вам предлагается оформить код так, чтобы он был максимально готов к развёртке в любом пайплайне CI/CD.