

Тестовое задание Python (Junior Flask)

Задача:

Разработать внутренний сервер, который позволяет пользователям управлять файлами, включая сохранение файлов различных доступных форматов, получение списка файлов по расширению, проверку на наличие дубликатов файлов при загрузке нового файла и удаление файлов.

Требования:

1. Серверная часть должна быть создана с использованием Python3.11 Flask.
2. Сервер должен иметь следующие конечные точки:
 - `/files/get/list` : GET-запрос, который возвращает список всех доступных файлов.
 - `/files/get/{extantion}` : GET-запрос, который возвращает список файлов с указанным расширением.
 - `/files/create/` : POST-запрос, который принимает загрузку файла и сохраняет его на сервере, проверяя на наличие дубликатов.
 - `/files/delete/{file_name}` : запрос на удаление, который удаляет указанный файл с сервера.
 - `/files/get/{extantion}/{file_name}` : GET-запрос, который возвращает выбранный файл.
3. Сервер должен поддерживать загрузку файлов в любом из следующих форматов:
 - Text files (.txt)
 - CSV files (.csv)
 - JSON files (.json)
 - Excel files (.xlsx)
 - Image File (.png, .jpg)



Проще сказать — он должен уметь работать с любым типом файлов. И важно что бы были ограничение в максимальном размере файлов.

4. При загрузке нового файла сервер должен проверить, есть ли уже файл с таким же содержимым в системе. Если файл с таким же содержимым уже существует, сервер не должен сохранять новый файл и вместо этого возвращать соответствующее сообщение об ошибке.
5. Весь код должен быть написан, с максимально покрывающими и соответствующими подсказками по типу и ссылками на документы.
6. Сервер должен быть протестирован с использованием соответствующих платформ тестирования, таких как Pytest или Unit test.

Примеры запросов на ссылку:

- **GET** `/files/get/list` : Возвращает список всех доступных файлов в формате JSON.
- **GET** `/files/get/csv` : Возвращает список всех CSV-файлов в формате JSON.
- **POST** `/files/create/` : Загружает новый файл на сервер. Тело запроса должно содержать содержимое файла и метаданные.
- **DELETE** `/files/delete/{file_name}` : Удаляет файл с идентификатором 123 с сервера.