

Задание - Сервис безопасного обмена файлами

Разработать REST API сервис для безопасного обмена файлами через временные ссылки доступа с использованием S3-совместимого хранилища и базовой аутентификацией по API-ключу (мини Яндекс-диск)

Аутентификация

Система использует простую аутентификацию через заголовок X-API-Key:

- Каждый запрос должен содержать заголовок X-API-Key с уникальным идентификатором пользователя
- Файлы привязываются к пользователю на основе этого ключа
- Пользователь имеет доступ только к своим файлам

Пример запроса:

```
GET /api/v1/files
X-API-Key: user-12345
```

Обязательные методы

| Метод | URL | Описание | Параметры | Тело запроса |
|--------|-------------------------------|---|----------------|----------------------|
| POST | /api/v1/files/upload | Загрузка нового файла | - | Form-data с файлом |
| GET | /api/v1/files | Список файлов текущего пользователя | - | - |
| GET | /api/v1/files/{file_id} | Информация о файле (только если владелец) | file_id (path) | - |
| POST | /api/v1/files/{file_id}/share | Создание временной ссылки (только владелец) | file_id (path) | {"expires_in": 3600} |
| DELETE | /api/v1/files/{file_id} | Удаление файла (только владелец) | file_id (path) | - |
| GET | /api/v1/stats | Статистика для текущего пользователя: <ul style="list-style-type: none"> общее количество файлов пользователя суммарный размер всех файлов в байтах | - | - |

Структура хранения в S3

Рекомендуемая реализация:

- Один бакет (универсальный) для всего сервиса
- Префиксы files/{api_key}/ для организации по пользователям
- Метаданные содержат владельца и оригинальное имя файла

Пример:

```

Бакет: file-exchange-service
└── files/user-123/
    └── abc-123-456 (метаданные: owner=user-123, filename=doc.pdf)
        └── def-789-012 (метаданные: owner=user-123, filename=photo.jpg)
└── files/user-456/
    └── ghi-345-678 (метаданные: owner=user-456, filename=report.docx)
└── files/user-789/
    └── jkl-901-234 (метаданные: owner=user-789, filename=data.xlsx)

```

Стек технологий:

- Веб-фреймворк: FastAPI (Python) или Spring Boot (Java) или другие на выбор
- S3-клиент: boto3 (Python) или AWS SDK for Java 2.x или для собственного фреймворка
- S3-хранилище: MinIO как объектное хранилище или альтернатива (см. критерии)

Критерии оценки

| Критерий | Баллы | Описание |
|-------------------------------------|--------|--|
| Работа с S3 и метаданными | 3 | Корректная организация в одном бакете, использование префиксов, метаданные владельца |
| API функциональность | 3 | Все эндпоинты работают согласно спецификации, валидация данных |
| Аутентификация и авторизация | 3 | Проверка X-API-Key, разграничение прав доступа к файлам, корректная обработка ошибок 401 и 403 |
| Pre-signed URLs | 1 | Временные ссылки генерируются и корректно протухают |
| Поддержка отечественного S3 | 2 доп. | На выбор из лекции |
| Простой веб-интерфейс | 1 доп. | |

Формат сдачи

- ссылка на репозиторий:
 - код сервиса
 - README с инструкцией по запуску
- устная защита на 10-15 минут, демонстрация:
 - структуры и содержимого хранилища (например, в веб-консоли)
 - основной функциональности сервиса