HTTP API для работы с файловой системой

## Задача

Требуется создать HTTP API для работы внешних пользователей с файловой системой на сервере.

## Команды API

* Создать файл из данных в запросе
* Обновить содержимое файла из данных в запросе
* Получить содержимое файла
* Получить метаданные файла
* Получить список файлов

## Дополнительно

* Поддержка вложенных директорий не требуется. Достаточно реализовать операции над файлами в одной директории.

## Примеры

* AWS S3 API <http://docs.aws.amazon.com/AmazonS3/latest/API/APIRest.html>
* Dropbox Core API <https://www.dropbox.com/developers/core/docs>

## Обязательные условия

* Корректная обработка всех исключительных ситуаций и согласованный формат ошибок в ответах API
* Наличие тестов

## Дополнительными плюсами будет:

Можно реализовывать любое количество пунктов из списка ниже (даже ноль). Они разбиты по логическим блокам и друг от друга не зависят (даже внутри блока).

Если вы не успеваете сделать функционал, который хотите реализовать, можете добавить описание в readme:

* Какие подходы и технологии вы бы использовали
* Почему эту фичу важнее реализовать, чем другие

## Список дополнительных фич

* Работа с PHP экосистемой
  + Использование composer
  + Использование сторонних библиотек и фреймворков
  + Использование функционала последних версий php(5.5 - 7.0)
* Оформление
  + README.md или документация(желательно на английском)
  + Оформленный проект на Github
* Оптимизация
  + Оптимизация загрузки и отдачи файлов(в первую очередь используемой памяти)
  + Gzip сжатие ответов
  + Поддержка gzip запросов
  + Хранение файлов в сжатом виде
  + HTTP кэширование повторных вызовов получения данных
  + Работа с оооочень большими файлами
  + Разруливание конкурентного доступа к файлу
  + Отдача фрагмента файла
* Безопасность
  + Ограничение директории, доступной для записи
  + Лимиты на размер загружаемых файлов
  + Механизм авторизации в апи
  + HTTPS
  + Ограничения на количество запросов к API для клиента и/или ip адреса
  + Привязка файла к клиенту-создателю и запрет на изменение для других клиентов.
  + Квоты с местом и лимит для пользователя
  + HTTP заголовки в ответах, для безопасного вызова API со сторонней веб-страницы в браузере
* Окружение
  + Запуск из встроенного PHP сервера
  + Настроенный Vagrant/Docker/Chef/Puppet с веб-сервером, PHP и приложением
* Остальное
  + Правильный Content-Type в ответах
  + Отдельный документ с описанием архитектуры подобного API, работающего с высокой нагрузкой(больше 100 запросов в секунду, терабайты данных)