Отчет. Задание № 1. Оптимизация загрузки изображений.

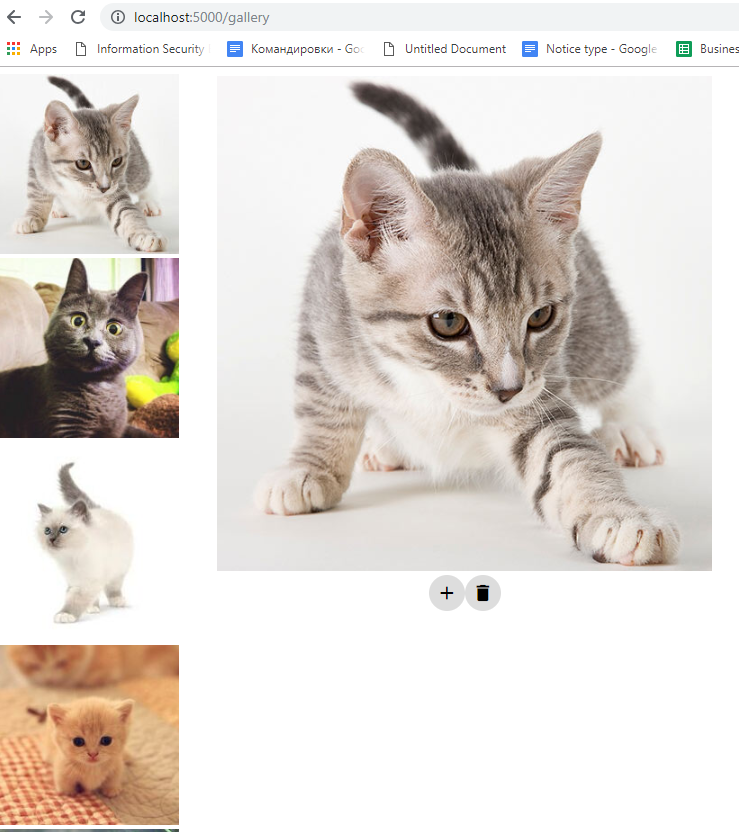
Левченко Антон, 43506/3

1. **Разработать приложение (сервис) для обмена фотографиями (на подобие instagram). Для реализации использовать любой серверный язык программирования.**

Приложение написано с использованием технологий angular 7 для клиентской части и nodejs express для сервера.

Репозиторий: **<https://github.com/antonylevchenko/ui_taks_1_image_optimization>**

Отображает список минифицированных изображений с сервера. При щелчке на изображение слева открывает его полную версию. Кнопка добавления позволяет загружать файлы. Кнопка удаления – удалять текущее изображение.



Как-то так.

Для запуска приложения необходимо произвести следующие действия:

* npm run install – установка нужных пакетов.
* npm run build – сборка angular приложения.
* node server.ts – запуск сервера nodejs.

Приложение будет доступно по адресу localhost:5000

Для разработки с использованием angular.cli и подхватом изменений на лету можно воспользоваться командой npm run start, приложение будет доступно по адресу localhost:4200

1. **При загрузке пользователем изображения реализовать механизм автоматической подготовки минифицированных версий изоражения (например 100\*100, 250\*250). Для работы можно использовать любой из инструментов (Imagemagick, Jpegtran, Jpegoptim, Optipng и тд). Можно скомбинировать несколько инструментов.**

При загрузке изображения, создаётся его минифицированная копия (с шириной в 200px и очищенная от метаданных) при помощи imagemagick.

Логика находится в server/image-service.ts в методе add класса ImageService.

1. **При отображении страницы с изображениями, на общей странице подгружать минифицированную версию, при клике по изображению открывать его полную версию.**

Для списка используются минифицированные версии изображений из директории data/images/minified. При нажатии на минифицированное изображение, загружается его полная версия из директории data/images/normal.

1. **Разработать web-worker, который по заданным правилам (по запуску адином, по таймингу и тд) будет проходить по папке с загруженными изображениями и достраивать отсутствующие минифицированные изображения.**

Логика находится в файле do-minifying.ts Минификация может быть проведена посредством запуска команды:

* node do-minifying.ts

1. **Добавить ленивую загрузку изображений (подгружаются первые N фотографий, остальные догружаются при скролинге)**

Ленивая загрузка изображений реализована при помощи расширения @trademe/ng-defer-load и используется в компоненте image-list.

В качестве альтернативы рассматривался virtual-scroll из angular-material, но заданное в нём огромное количество записей для пререндеринга не позволило улучшить производительность на имеющемся числе изображений.

Также рассматривался grid от teleric поддерживающий технологию virtual-scroll с гибкой настройкой. К сожалению, его использование требовало установки rxjs/compat для совместимости с angular 6+, что моментально приводит к падению производительности нивелирующему все преимущества.

1. **Произвести нагрузочное тестирование и оценить эффект от минификации.**

Эффект наиболее заметен на небольших экранах (на больших увеличивается количество изображений для пререндеринга) с большим количеством картинок.

1. **Выводы**

Минификация и ленивая загрузка позволяют существенно увеличить быстродействие сайта за счет уменьшения объёма и количества передаваемых данных.