

# Новая технология, старые грабли

CacheStorage и ServiceWorker

Денис Омельков  
omelkov@skbkontur.ru



# Что дает нам ServiceWorker?

Пуш уведомления



Офлайн доступ



Ускорение отображения страниц

Фоновый процесс

Синхронизация состояния между вкладками



ПРОФИЛЬ

+7 912  
alexki

Сегодня

8:00

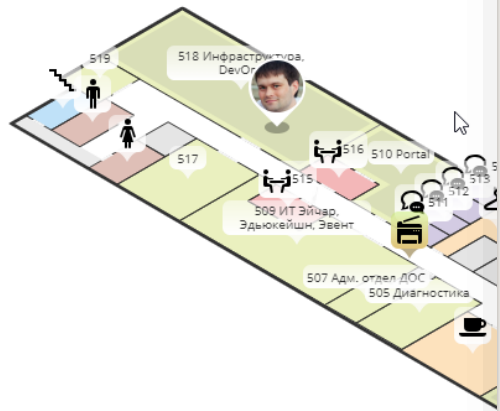
Должно

Руковод  
Отдел разработки средств эксплуатации  
Управление разработки  
Департамент программных продуктов  
АО «Производственная фирма «СКБ Контур»

Непосредственный руководитель  
Александр Голубев

## Местоположение сотрудника

Офис "Малопрудная 5", Екатеринбург, этаж 5, (518 Инфраструктура, DevOps)



Омельков Денис Иванович  
Ведущий программист

Профиль Лента

+7 982 631-98-70  
omelkov@skbkontur.ru

Народной Воли 19а, Екатеринбург,  
комната 501 ОРИР  
[Показать на плане](#)

Lync: 0272611  
Статус: неизвестно

Сегодня Завтра Выбрать день

8:00	9:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00

Что делаю  
Работаю над Стаффом, Командировками и  
Контур.Стримом. Фронтендер.

Должность

Ведущий программист  
Отдел развития информационных ресурсов  
Управление информационных технологий  
Департамент общих служб

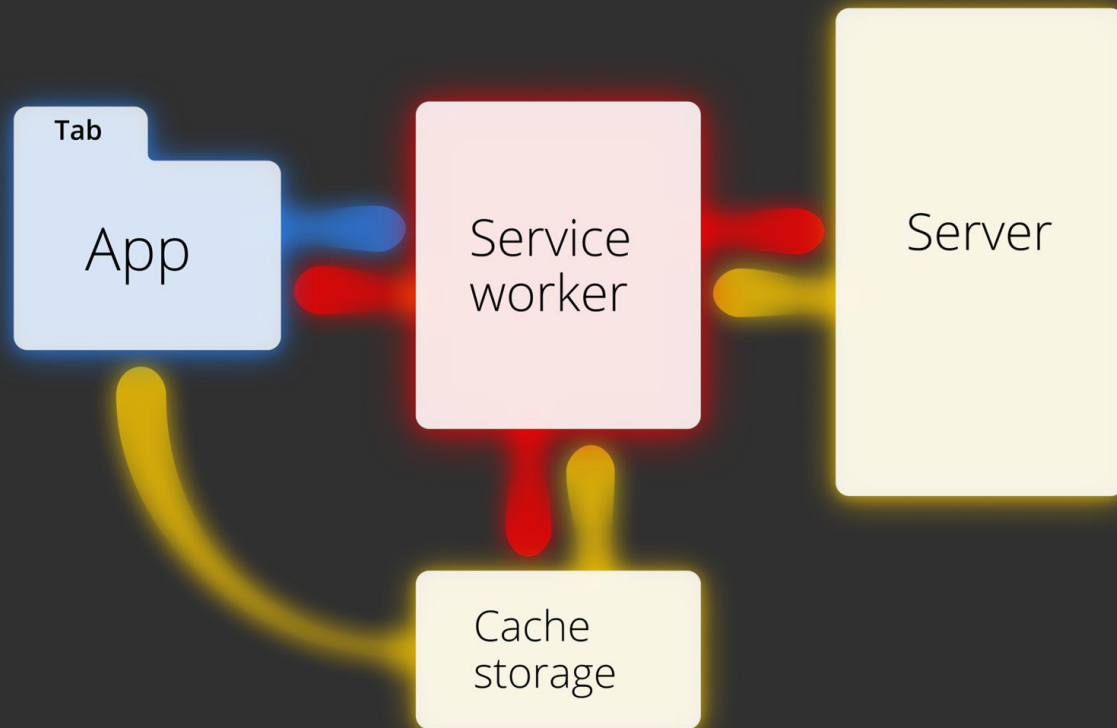
Стаж ?  
3 года и 5 месяцев

Группы в стаффе  
[Посмотреть группы](#)

Благодарности

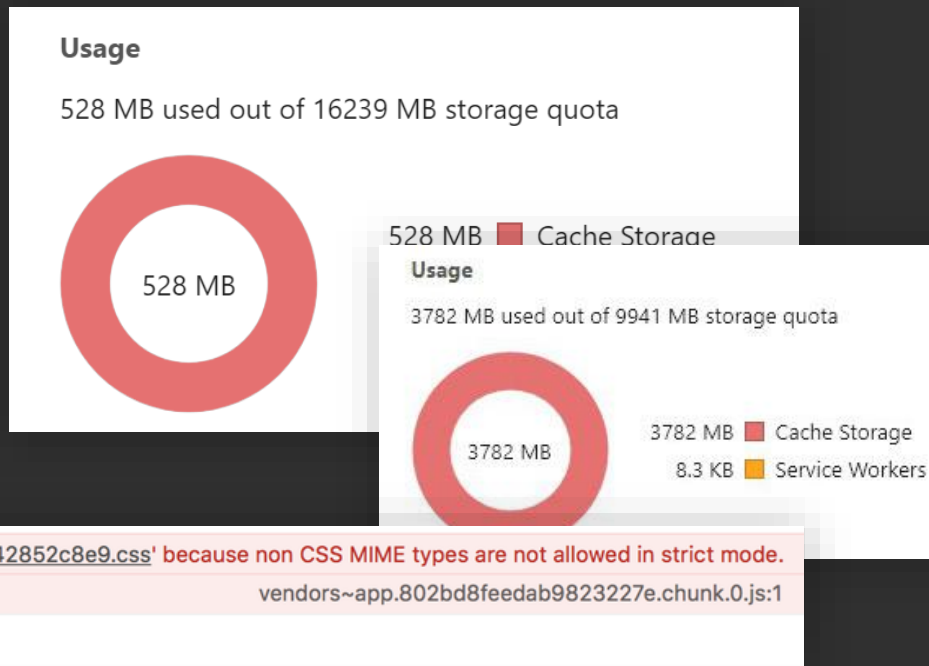


# Принцип работы кэширования



# На что напоролись?

Переполнение кэша  
Ошибки кэширования



# Как избежать переполнения?



Своевременная чистка



Избирательное кэширование



# Своевременная чистка:

по `activate`

```
self.addEventListener('activate',  
function(event) {  
    event.waitUntil(  
        // ЧИСТИМ КЭШ  
    );  
});
```



# Своевременная чистка:

по событию из приложения

```
self.addEventListener('message', event => {  
    if(event.data.action === 'cache-clean'){  
        // ЧИСТИМ КЭШ  
    } ;  
});
```



# ★ Чистка кэша:

по версии приложения

```
cache.keys().then((namespaces) => {  
  namespaces.forEach((key, i) => {  
    if (key !== "v2.1") {  
      cache.delete(namespaces[i]);  
    }  
  });  
});
```

# ★ Чистка кэша:

по количеству сохраненных запросов

```
 caches.open(namespace).then((keyList) => {
    const keysToDelete = keyList.length - 1000;
    const removePromises = [];
    keyList.forEach((key, i) => {
      if (i < keysToDelete) {
        removePromises.push(this.cache.remove(key));
      }
    });
    return Promise.all(removePromises);
  });
```

# ★ Чистка кэша:

по **устареванию**. Где взять дату?

Заголовок `date` – только в `response`, нужно перелопатить весь кэш

Заголовок `date` может отсутствовать

При кэшировании вручную проставлять в `request` заголовок с датой (или во внешнее хранилище)

Проблемы создания объекта запроса на основе существующего

ДОРОГО

# ★ Чистка кэша:

по логике приложения

Устаревшие API и файлы

Событие из приложения

## Способ

Версионирование  
неймспейса

Ограничение  
сохранения

Очищение

**YOUR CACHE WILL NOT OVERFLOW  
IF YOU DON'T CACHE ANYTHING**

Требуется менять  
версию

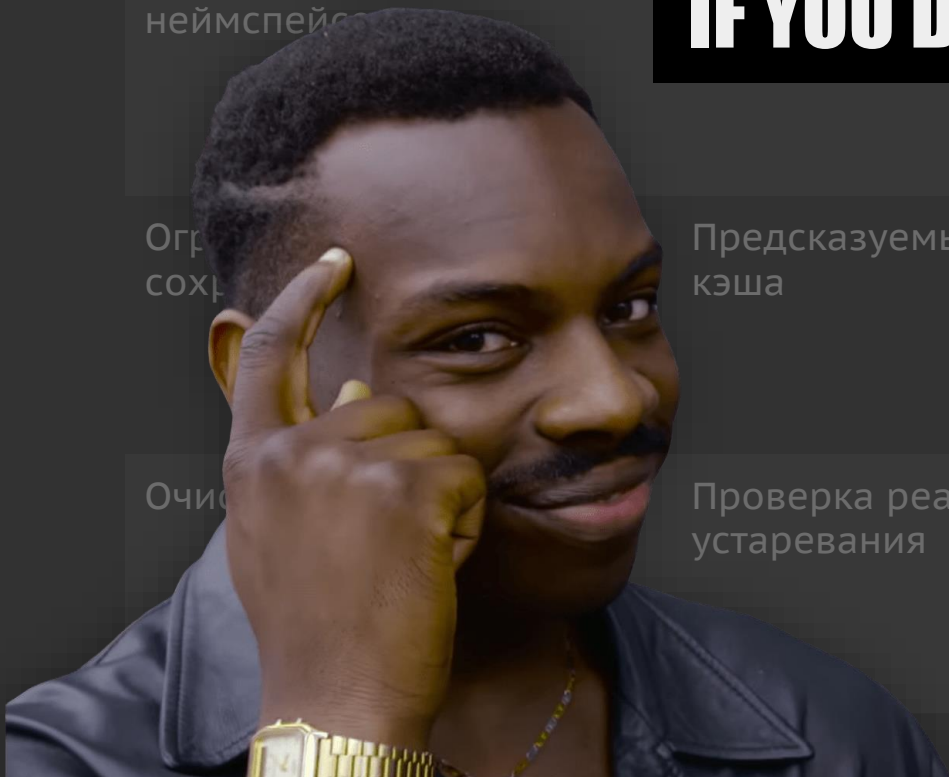
Предсказуемый размер  
кэша

Сложно управлять при  
разном размере  
ответов

Проверка реального  
устаревания

Сложно реализовать

Может медленно  
работать



# 🚫 Не кэшируем:

## большие файлы

```
fetch(request)
  .then((response) => {
    if(response.headers.get("Content-Length") < 512000){
      this.cache.set(request, response.clone());
    }
    return response;
  })
```

# 🚫 Не кэшируем:

## большие файлы

Заголовок `Content-Length` может отсутствовать

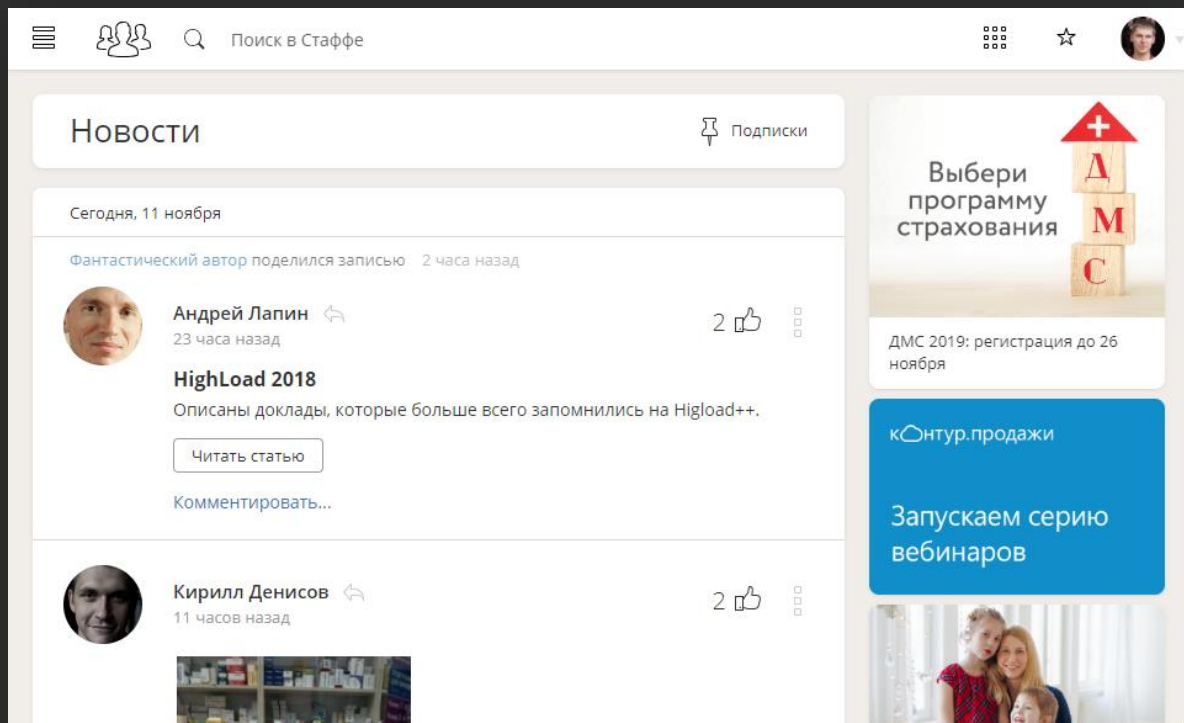
Можно оценить размер через blob

```
response.clone().blob()  
    .then(b => b.size)
```

# 🖐 Не всегда кэшируем: часто меняющиеся файлы

По блэклисту URL

- ленты
- чаты
- статусы
- и т.д.





# Несколько кэшей:

отдельные правила кэширования

	Thumbnails	Documents	API
Размер	—	< 50 Mb	< 1 Mb
Количество	1 000	100	1 000
Срок	—	10 дней	—



**Кэшируем с осторожностью:**  
**грабли**

[bit.ly/messUpSWCaching](https://bit.ly/messUpSWCaching)

index.html

Бандлы без версионирования

Внешние ресурсы



# Кэшируем с осторожностью:

## Клиентский роутинг с URL Rewrite



GET/bundle.[hash].js



index.html



Проверяйте Content-Type ответа



# Полезный заголовок:

Clear-Site-Data: storage cache



61



63

# Особая ситуация:

Logout



Есть смысл  
очистить весь кэш

# Ссылки:

## Что посмотреть и почитать

Фреймворк для работы с Service Worker и CacheStorage (англ.)

[bit.ly/SWWorkbox](https://bit.ly/SWWorkbox)

Жизненный цикл Service Worker (англ.)


[bit.ly/SWLifecycle](https://bit.ly/SWLifecycle)

Доклад Сервис-воркеры: используем накопленный опыт и смотрим в будущее

[bit.ly/HolyJS2018ServiceWorker](https://bit.ly/HolyJS2018ServiceWorker)

# Вопросы?

[bit.ly/SWExpSlides](https://bit.ly/SWExpSlides)

Денис Омельков  
[omelkov@skbkontur.ru](mailto:omelkov@skbkontur.ru)  
 [@apocalyp\\_sys](https://twitter.com/apocalyp_sys)

к<sup>cloud</sup>онтур