

Обзор Web Push Notifications. Пишем свое решение.

Александр Михайленко, Jearie
19.06.2015



<http://www.devconf.ru>

Jeapie

- Занимаемся только push-уведомлениями с 2013
- Более 500 млн персонализированных сообщений
- Выпустили сервис Chrome web push для сайтов в открытую бету 10.06.2015



TUT.BY
БЕЛОРУССКИЙ ПОРТАЛ

MAXIM



Как устроены браузерные
PUSH-УВЕДОМЛЕНИЯ



КАК ОНИ РАБОТАЮТ?

Посетитель заходит
на ваш сайт



Появляется диалоговое окно и
посетитель соглашается
получать уведомления



Вы можете посылать сообщения
этому пользователю в любое
время. Даже когда браузер закрыт



ГДЕ ОНИ ПОЯВЛЯЮТСЯ?

MacOS и Linux
в правом верхнем углу



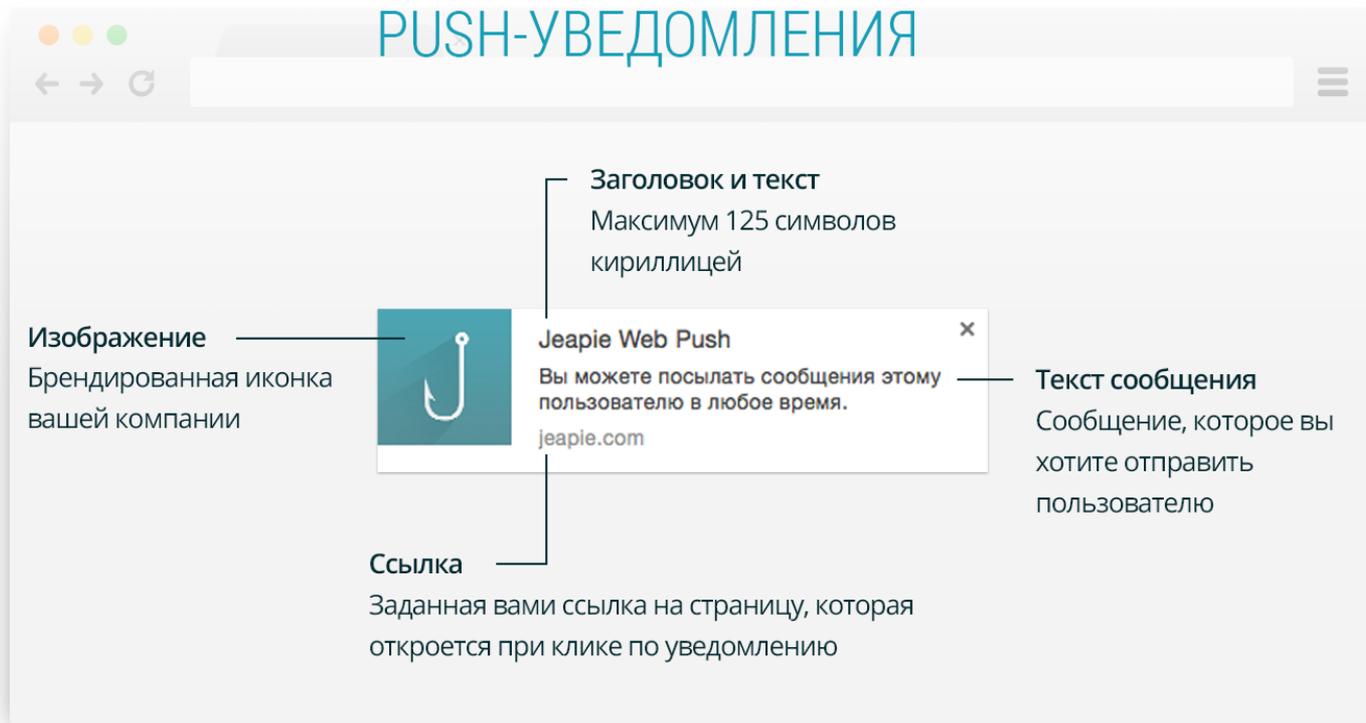
Android lockscreen
или в панели уведомлений



Windows
в правом нижнем углу



АНАТОМИЯ БРАУЗЕРНОГО PUSH-УВЕДОМЛЕНИЯ



6 ПРИЧИН ИСПОЛЬЗОВАТЬ WEB PUSH



Новый канал

Начал активно развиваться только весной 2015 года с выходом Chrome 42



Удобный для пользователя

Сообщения выстраиваются в удобную ленту, есть режим "не беспокоить"



Высокий процент подписки

На тестовых замерах показатели в 30 раз превысили Email



Актуальность

Техническое устройство гарантирует невозможность засорения канала спамом.



Читаемость

За счет доставки прямо на рабочий стол, читаемость сообщений находится на уровне свыше 90%



Кросс-платформенность

Приходят на мобильные устройства и на стационарные компьютеры

DEMO (Chrome)

<https://jeapie.com/web>

Стандарт

<http://www.w3.org/TR/push-api/>

(последняя редакция 27 апреля 2015)

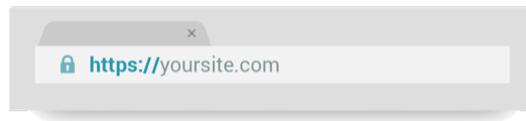
Компоненты Chrome Web Push

- `manifest.json` - лежит в корне, указывается в `<head>` содержит `ProjectNumber`, по нему генерируются токены
- `service-worker.js` - лежит в корне сайта, отвечает за прием и обработку пуш (кешируется в браузер пользователю)
- js-сниппет - в `layout` сайта, проверяет все ли ок? и вызывает окно подписки

Требования

- Google Chrome 42+ у пользователя
- SSL-сертификат у домена (https://)

Сайт должен быть защищен SSL протоколом



Токены

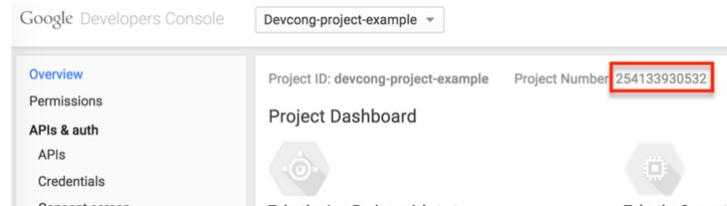
- Уникальный идентификатор **устройства**
- Уникален для данного домена, ProjectNumber и браузера
- Может “устаревать” и меняться (к примеру при отписке\подписке)

Шаг №1. Получаем ключи

<https://code.google.com/apis/console>

Key for server applications

API key	AlzaSyAXtpNT1SF6wUFJSxEY01CWavZWFGCODEk
IPs	Any IP allowed
Activation date	Jun 17, 2015, 9:19:00 AM



Google Developers Console Devcong-project-example

Project ID: devcong-project-example Project Number: 254133930532

Project Dashboard

Overview
Permissions
APIs & auth
APIs
Credentials
Concept screen

Take the App Engine tutorial
Take the Compute Engine tutorial

Шаг №2. Создаем manifest.json

```
{  
  "display": "standalone",  
  "gcm_sender_id": "625487121323",  
  "gcm_user_visible_only": true  
}
```

в <head> вставляем ссылку

Шаг №3. Пишем service-worker

В нем находится обработка событий

1. Событие получения notification
2. Событие открытия notification

Он кешируется при первом заходе, запускается при получении событий, работает в фоне некоторое время (50ms)

Особенность(!!) Chrome Push

В версиях Chrome 42-43 пуш не содержит ни текст уведомления ни идентификатор уведомления.

Поэтому приходится на сервере держать очередь пушей для каждого токена. И делать запрос на сервер при каждом приходе пуша.

Шаг №4. js-снippet

- Поддерживает ли браузер пуш
- Поддерживает ли браузер сервис воркеры
- Регистрируем сервис-воркер
- Не запретил ли пользователь прием пуш-уведомлений
- Вызываем окно подписки

Шаг №5

- Сохраняем токен на бекенде после получения согласия
- Можем отправлять пуш используя GCM API (ключ для авторизации) при помощи обычных http запросов

Шаг №6. “Окультурить”

- Инструмент alias (customer_id)
- Трекинг доставки Push
- utm-метки трекинга перехода по ссылкам
- Поддержка http сайтов через виджет
- Сегментация по тегам

Спасибо за внимание!

Александр Михайленко
CEO
Jeapie



@alexandrmikh



fb.com/alexandrmikh



alexandr.mikhaylenko@gmail.com