

5分でPWA + 通知機能を実装してみよう

by 市井P - @ichii731

2021年9月15日関連資料

自己紹介



市井P

@ichii731



年齢: 18才(大学1年)

所属: 明治大学IMS学部FMS学科

総合数理

先端メディアサイエンス

住処: 東京23区のどこか

趣味: オタク系全般

アニメ & ガジェット & IT特化サイト

「オタク総研 with IT」運営代表 兼 編集責任

ウェブライター

主にPHP(Laravel)を利用して個人でWeb開発



自己紹介

新宿猫で話題のビジョンにて
明治大学の錯視広告プロジェクト
の共同メンバー(企画・広報とか)

錯視広告をWeb上でVRで試せる
アプリの企画開発者です…

これ→



FMS×錯視広告体験



1

Web Pushの価値と導入する方法

Web Pushを導入する意義

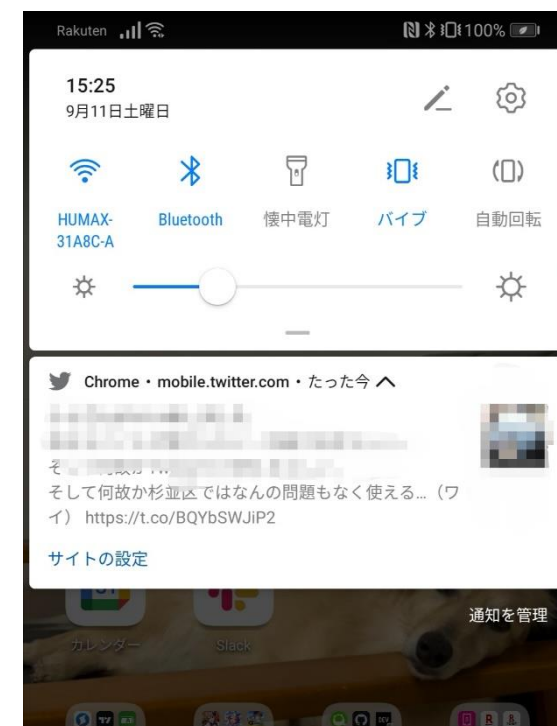
Web Pushとは…

ブラウザの種類によらず、JS APIを用いてブラウザ経由で受信できるようにする仕組み！

Use Case

- Web版Twitterの通知
- タイムセール開催予定
- クーポンの配布

→受動的な情報提供がやりやすくなる！



Web Pushを導入する意義

Web Push+PWAの成功事例

【受験のミカタ】

Push Notification経由だと

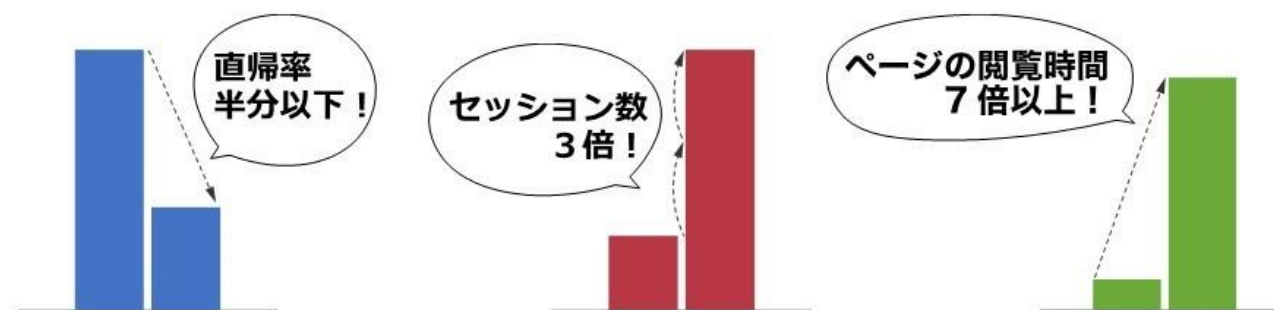
- 直帰率が**半分**以下
 - 訪問回数が**3倍**
 - サイトの閲覧時間は**7倍**以上
- という驚異的効果が。



参考リンク

Push Notificationの配信状況

「受験のミカタ」のPush Notificationによる流入でも、それぞれ平均と比べて、直帰率が半分以下になり、セッション数は3倍、サイトの閲覧時間は7倍以上になるという高い効果を上げました。



※セッション数：ユーザー1人あたりの平均サイト訪問回数

※Pantograph inc.より引用

※直帰率とは…最初のページだけを見てサイトを離脱してしまったセッションの割合のこと。

Web Pushを導入する意義

→Push通知は多くの業界にとって効果的。

※PWAではないがPush通知関連の最新の話題…

ホリエモンプロデュースのパン屋「小麦の奴隷」は

【Push通知=最強のマーケティングツール】

と銘打ってリピートされやすい体制を作っている。

(ネイティブアプリだけどWebにも言えること)

結構美味しいんで食べてみてください。

Web Pushを実装する2つの手段

○自前でサーバーを立てて実装

Node.jsなどを用いて自前で実装する方法。

【利点】

- 外的な制約に縛られることなく利用可能(リクエスト制限等)

【欠点】

- 実装に時間がかかる
- Node.jsなどJavaScriptの知識が追加で必要

web-push

build failing dependencies up to date devDependencies up to date

Why

Web push requires that push messages triggered from a backend be done via the **Web Push Protocol** and if you want to send data with your push message, you must also encrypt that data according to the **Message Encryption for Web Push spec**.

This module makes it easy to send messages and will also handle legacy support for browsers relying on GCM for message sending / delivery.

Install

Installation is simple, just install via npm.

```
npm install web-push --save
```

Usage

The common use case for this library is an application server using a GCM API key and VAPID keys.

```
const webpush = require('web-push');

// VAPID keys should be generated only once.
const vapidKeys = webpush.generateVAPIDKeys();
```


Web Pushを実装する2つの手段

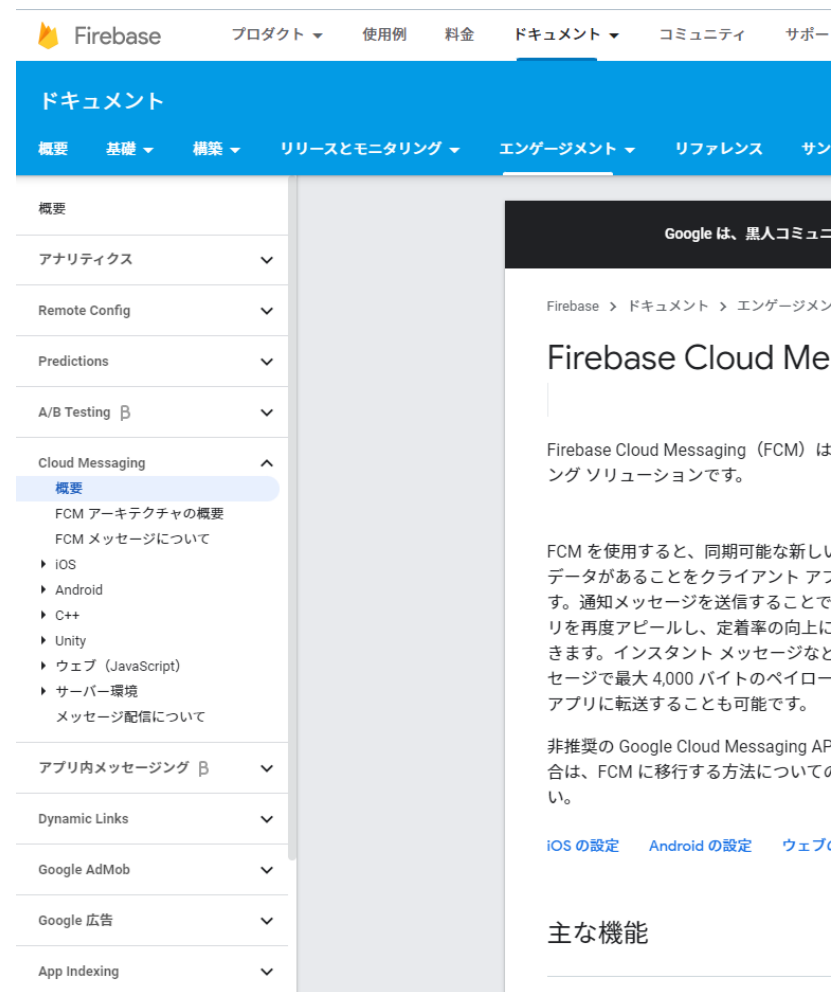
○外部サービスを利用して実装

【利点】

- Webページに埋め込むだけで実装できる
- 通知実装によるサーバーリソースの浪費が格段に下がる
- アナリティクス等の解析機能が優秀

【欠点】

- APIの利用制限等の制約がある



Web Pushを実装する2つの手段

- ✓ PWAの長所「インスタントさ」を損ないたくない
- ✓ 時間的・知識的成本を最大限低くして“爆速実装”を実現したい
- ✓ 非エンジニアにもわかりやすいGUI・簡単操作で使いたい



外部サービスを活用！

FCM vs OneSignal

メッセージングソリューションの2大巨頭

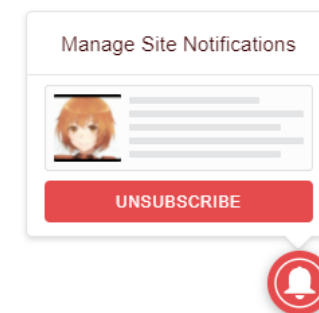
- Firebase Cloud Messaging
- OneSignal

のフリープランを比較してみる。

使いやすいかは「中・小規模事業」を基準に考慮。

FCM vs OneSignal

FCM	項目	OneSignal
-	最大送信数	10,000
△	管理のしやすさ	○
○	A/B Test	○
×	時間指定送信	○
◎	アナリティクス	○
○	通知登録のしやすさ	◎
×	画像の添付	○



結論:今回はOneSignalを使うけど好きな方どうぞ…

2

OneSignal×PWAで実際に作ってみよう

前提条件(説明するにあたって)

- ・予めnginxやApacheでWebサーバー環境を作っておく！今回はApache。
(勿論外部サービスを使うので静的ホスティングサービスでも可。)

- ・HTML・バニラのJavaScriptでコンテンツを事前に作成
→取り敢えずHTML+VanillaJSにしとけば色々活用できるから…

特にLaravelとかはファイルの位置入れ替えるだけで対応できるし。

```
</script>
</head>
<body>
  <h2>Push&SW Test</h2>
  
  <script>
```

- ・SSL環境が必須！


→頑張ればlocalhostや非SSLでも出来るらしいが無難にLet's Encryptで対処。

OneSignalを設定する

アカウントを作成

GitHub連携でとにかく時短



 OneSignal

Powerful Messaging
Free plan includes unlimited mobile, up to 10K web push subscribers, email, SMS & 1 in-app message. No credit card required.

Quick setup
Our users are up and running in under 15 minutes.

Trusted by millions of leading businesses
We power messaging for over 10%+ of all mobile apps and 4% of top websites worldwide, and we send 8 billion messages per day.

Already have an account? [Log in.](#)

Get started with OneSignal for free

Email Address
Password
8 character minimum [Show](#)Company or Organization Name
By signing up to OneSignal you agree to our [Terms of Service](#) and [Privacy Policy](#).

Create Account

or

 Google Facebook GitHub

OneSignalを設定する

Appを新規作成

1.New App/Websiteをクリック

The screenshot shows the OneSignal 'All Applications' dashboard. At the top, there's a red circular icon with a white signal symbol, followed by the text 'All Applications'. Below this, there's a blue button with a white plus icon and the text 'New App/Website', which is highlighted with a red rectangle. To the right of this button is a search bar with the placeholder text 'Search apps'. Further right is a blue button with the text 'Upgrade'.

Below the 'New App/Website' button is a table with the following data:

NAME	TOTAL USERS	TOTAL
Test	1 (∞)	-----

Below the table, there's a form for creating a new app. The first field is labeled 'Name of your app or website' and contains the text 'PWAPushuTest', which is highlighted with a red rectangle. Below this field is the text 'What platform do you wish to use for this app? You can set up more later.'.

There are four platform options, each with an icon and text:

- Apple iOS (APNs) with an Apple logo icon.
- Google Android (FCM) with an Android robot icon.
- Web with a Chrome logo icon, which is highlighted with a red rectangle.
- Email with a Gmail logo icon.

Below the platform options is a blue link that says '> More Options'.

At the bottom of the form is a blue button with the text 'Next: Configure Your Platform'.

2.Appの名前を入れてWebを選択！

OneSignalを設定する

Appを新規作成

3. サイト情報を入力

4.SDKをダウンロード

nl > Settings

Web Configuration

4. Upload Files

Next, you will need to download the OneSignal SDK files and upload them to the top-level root of your site directory.

 **DOWNLOAD ONESIGNAL SDK FILES**

Web Configuration

1. Choose Integration



Typical Site

For typical websites. Select this if you are unsure.



Wordpress Plugin or Website Builder

For sites using Wordpress, Blogger, Shopify, or other content management systems.



Custom Code

For advanced users who wish to code their web push configuration using JavaScript.

2. Site Setup

SITE NAME *

SITE URL *

AUTO RESUBSCRIBE (HTTPS ONLY) ☒


DEFAULT ICON URL

[+ Upload](#)

☐ My site is not fully HTTPS

SITE SETUP

Enter your site name, URL, and icon URL to set up Web Push on your site.

 [READ OUR DOCUMENTATION](#)
Web Push Setup

3. Advanced Push Settings (Optional)

SAFARI CERTIFICATE ☐ Upload your own .p12 certificate

ADVANCED OPTIONS

Most users will never have to use these options. We recommend only users that need these specific features use them.

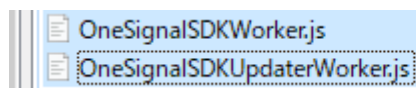
Save

Cancel

OneSignalを設定する

Appを新規作成

5.SDKを任意の位置に配置



SDKの位置はちゃんと考えて…
SW登録したら変えるの大変…

6.表示されたコードをheadタグにコピペ

これでOK！

5. Add Code to Site

If you haven't already, add this code to the `<head>` section on all pages of your site that users can subscribe to.

Because you have chosen to use **Custom Code**, you will also need to add additional code in this section to prompt users to subscribe, along with any other logic you want to use.

 COPY CODE

```
<head>
...
<script src="https://cdn.onesignal.com/sdks/OneSignalSDK.js" async=""></script>
<script>
window.OneSignal = window.OneSignal || [];
OneSignal.push(function() {
  OneSignal.init({
    [REDACTED]
  },
  {
    notifyButton: {
      enable: true,
    },
  });
});
</script>
...
</head>
```

既存サイトをPWAに対応させる

Manifest.jsonを整備

時短

PWA Manifest Generator

```
{ } manifest.json U ×  
{ } manifest.json > [ ] icons  
1 {  
2   "theme_color": "#f69435",  
3   "background_color": "#f69435",  
4   "display": "minimal-ui",  
5   "scope": "/",  
6   "start_url": "/",  
7   "short_name": "PWAPushTest",  
8   "name": "PushTest",  
9   "icons": [  
10  {
```



PWA Manifest Generator

Automatically generate a fully functional web app manifest along with size-optimized icons for your PWA

Name *	Short Name *
<input type="text"/>	<input type="text"/>
Display	Description
<input type="text" value="Browser"/>	<input type="text"/>
Application Scope	Start URL *
<input type="text" value="/"/>	<input type="text" value="/"/>
Theme Color	Background Color *
<input type="color" value="#f69435"/>	<input type="color" value="#f69435"/>
Icons *	
Please upload a 512x512 image for the icon and we'll generate the remaining sizes	
<input type="button" value="ファイルを選択"/> 選択されていません	

manifest.json Preview

```
{  
  "theme_color": "#f69435",  
  "background_color": "#f69435",  
  "display": "browser",  
  "scope": "/",  
  "start_url": "/"  
}
```

GENERATE MANIFEST

[Download sample Service Workers](#)

既存サイトをPWAに対応させる

ServiceWorkerを整備

時短

Workbox CLI



SWを手軽に扱うためのツールキット「Workbox」のCLI版。Google謹製。

workbox-sw + workbox-webpack-pluginでも良かったけど、
素のJS環境で検証するため、なるべくWebpackとかを使いたくなかったから。あと普通に時間短縮。

インストールは `npm install -g workbox-cli` で一発。

```
~# npm install workbox-cli --global
added 431 packages, and audited 432 packages in 23s

47 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```



既存サイトをPWAに対応させる

ServiceWorkerを整備

①ウィザード形式で設定ファイルを生成

workbox wizard を実行して色々聞かれるので答えるだけ。
これで設定ファイル(workbox-config.js)が作成される。

```
? Which file types would you like to precache?  
> ● png  
  ● html  
  ● json  
  ● js
```

```
? What is the root of your web app (i.e. which directory do you deploy)? Manually enter path  
? Please enter the path to the root of your web app: /var/www/html/  
? Which file types would you like to precache? png, html, json, js  
? Where would you like your service worker file to be saved? /var/www/html/sw.js  
? Where would you like to save these configuration options? workbox-config.js  
? Does your web app manifest include search parameter(s) in the 'start_url', other than 'utm_' or 'fbclid' (like '?source=pwa')? No  
To build your service worker, run
```

```
workbox generateSW workbox-config.js
```

as part of a build process. See <https://goo.gl/fdTQBf> for details.

You can further customize your service worker by making changes to workbox-config.js. See <https://goo.gl/gVo87N> for details.

既存サイトをPWAに対応させる

ServiceWorkerを整備

②ServiceWorkerのJSを生成

`workbox generateSW workbox-config.js` を実行→勝手にSWを作ってくれる！
これでsw.jsが作成されるので配置。

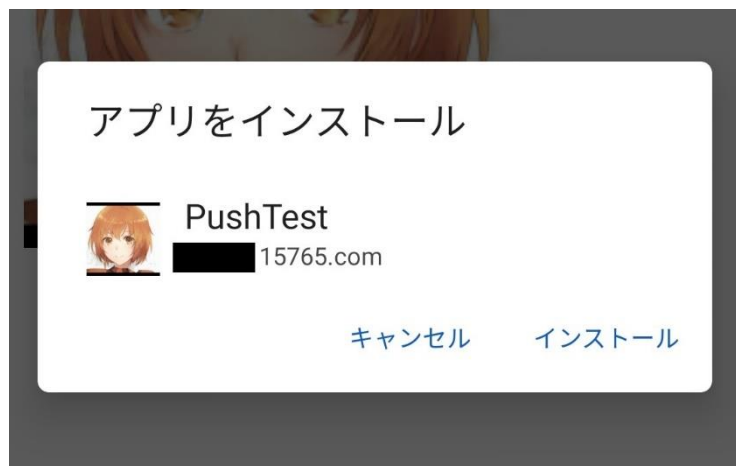
```
Using configuration from [redacted]workbox-config.js.  
The service worker files were written to:  
• [redacted]sw.js  
• [redacted]/sw.js.map  
• [redacted]/workbox-4df85334.js  
• [redacted]/workbox-4df85334.js.map  
The service worker will precache 8 URLs, totaling 792 kB.
```

あとはHTML内にServiceWorkerを登録させる処理を書くだけ！

```
30  <script>  
31      if ('serviceWorker' in navigator) {  
32          window.addEventListener('load', () => {  
33              navigator.serviceWorker.register('/sw.js')  
34          })  
35      }  
36  </script>
```

PWAが使えるか確認&Push通知を許可

インストールしてみる



A screenshot of the Chrome DevTools Service Workers panel. The left sidebar shows the "Application" tab with "Service Workers" selected. The main panel shows the details for a service worker from "https://115765.com/".

Service Workers

- ☐ Offline ☐ Update on reload ☐ Bypass for network

Source: [sw.js](#)
Received 2021/9/11 4:20:39

Status: ● #129 activated and is running [stop](#)

Clients: [https://115765.com/](#) [focus](#)

Push: [Push](#)

Sync: [Sync](#)

Periodic Sync: [Periodic Sync](#)

Update Cycle

Version	Update Activity	Timeline
#129	Install	<div style="width: 100%; height: 10px; background-color: #008000;"></div>
#129	Wait	<div style="width: 0%; height: 10px; background-color: #FF0000;"></div>
#129	Activate	<div style="width: 0%; height: 10px; background-color: #FFA500;"></div>

Service workers from other origins

[See all registrations](#)



PWAが使えるか確認&Push通知を許可 通知の受信を許可する

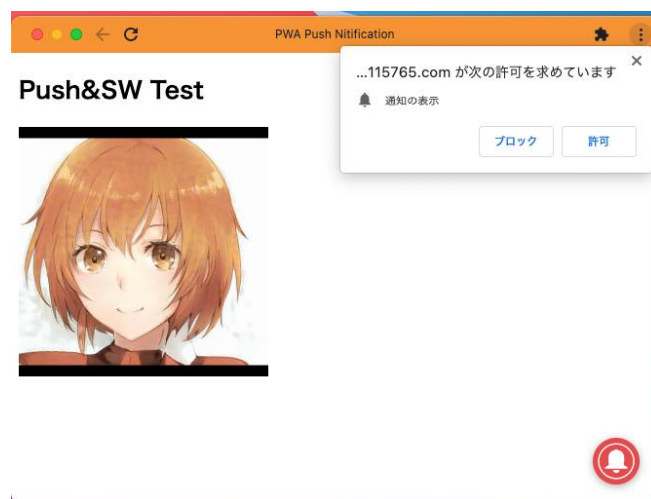
Windows



Android



Mac OS



OneSignalで通知を作成

2. Message

[Add Languages](#)

Title

プッシュ通知テスト

Message *

PWA経由でのプッシュ通知が作動しているかのテストです！

Image

<https://www.filepicker.io/api/file/WTpU00eTbCrEukyIMmW7?filename> [Upload](#)

Launch URL

<https://app.0115765.com>

[Platform Settings](#)

[Advanced Settings](#)

[A/B Test](#)

[Send to Test Device](#)



[Web: macOS](#)

3

動作確認・利用状況

動作テスト

タイムゾーンごとの時間指定・送信したい端末OSを指定 もできちゃいます！

3. Delivery Schedule

When should this message start sending? ?

☐ Immediately

☒ Specific Time

September 11, 2021 5:03 AM

Per user optimization? ?

☒ Send to everyone at the same time

☐ Intelligent delivery (recommended) ?

☐ Custom time per user timezone ?

Launch URL
https://app.0115765.com

> Platform Settings
> Advanced Settings

3. Delivery Schedule

When should this message start sending? ?

☒ Immediately
☐ Specific Time

Per user optimization? ?

☒ Send to everyone at the same time
☐ Intelligent delivery (recommended) ?
☐ Custom time per user timezone ?

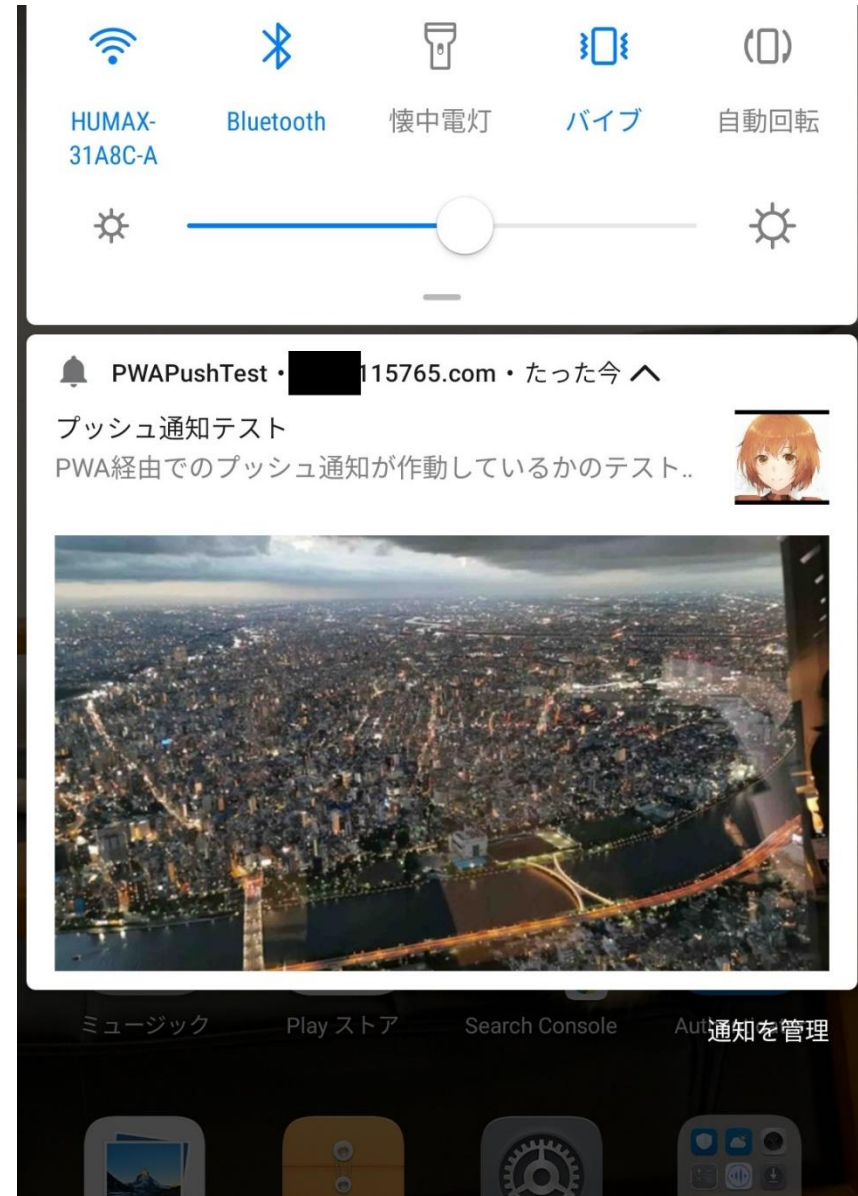
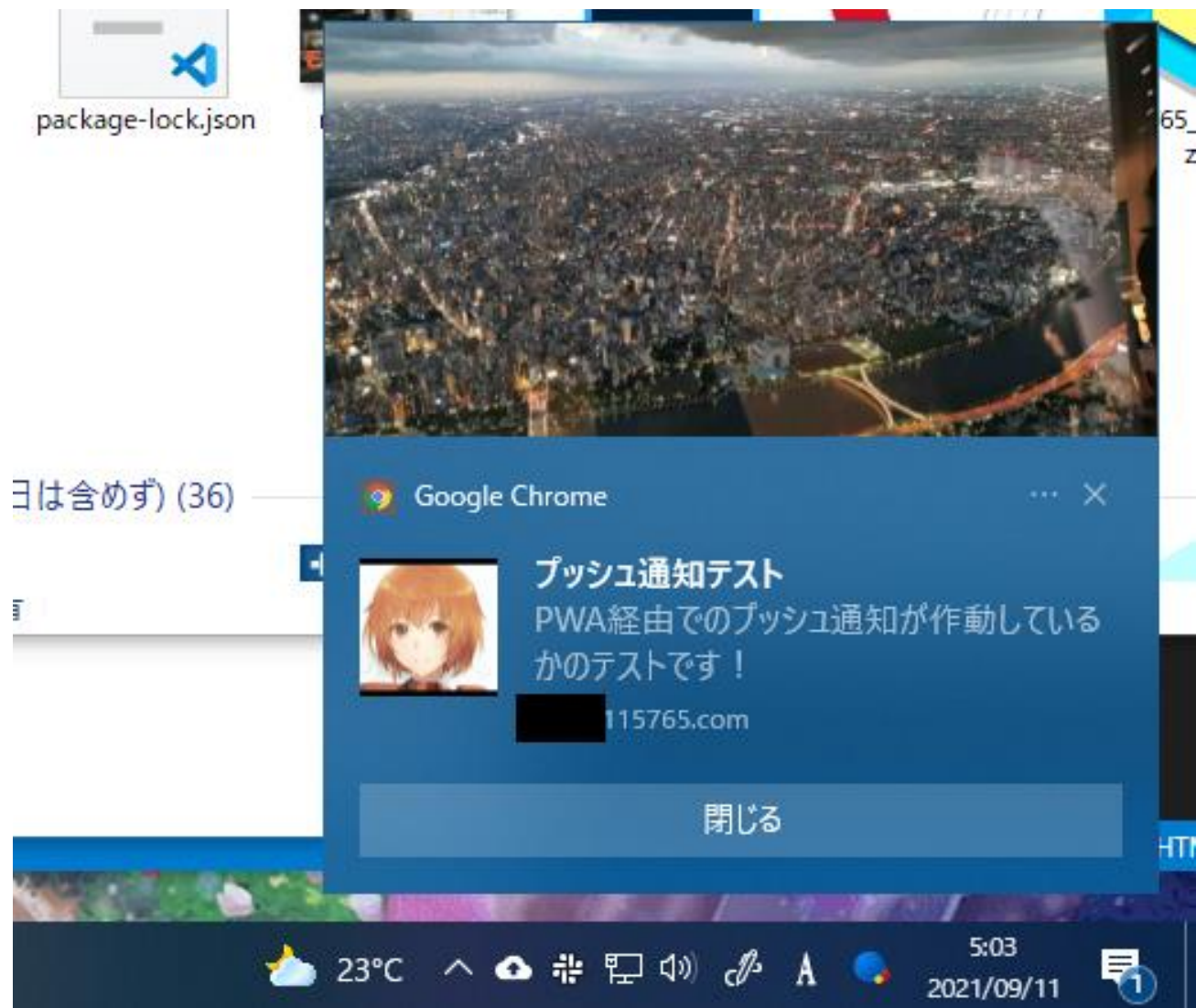
Review and Send

Review Your Message

Audience	
App	PWAPushuTest
Estimated recipients	3
Included segments	Subscribed Users
Message	
Platforms	
Title	プッシュ通知テスト
Message	PWA経由でのプッシュ通知が作動しているかのテストです！
Launch URL	115765.com
Schedule	
Start sending	Send immediately
Per user optimization	Deliver immediately

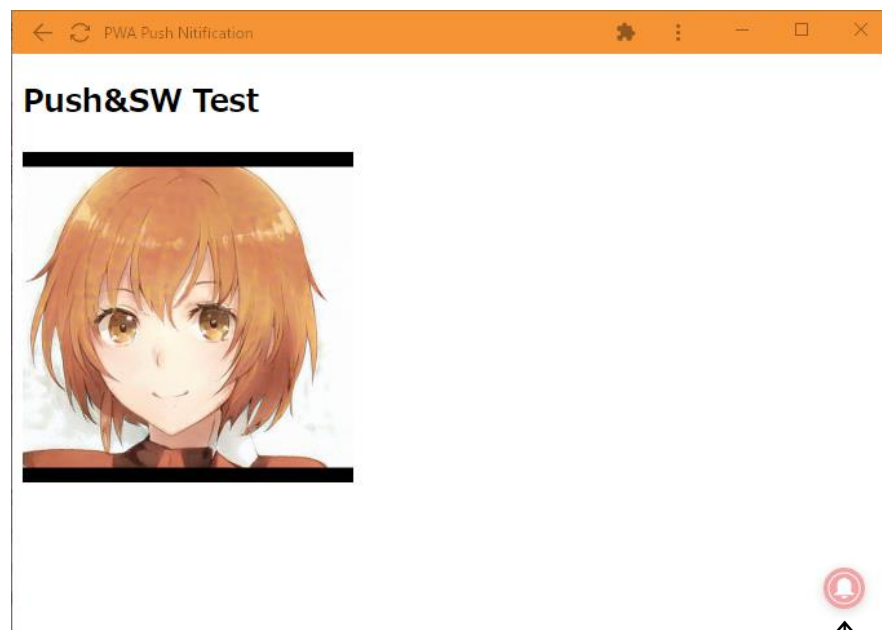
Make Changes Send Message

動作テスト

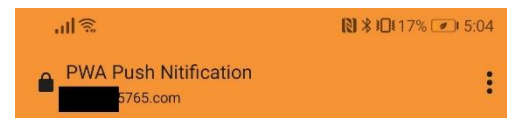


動作テスト

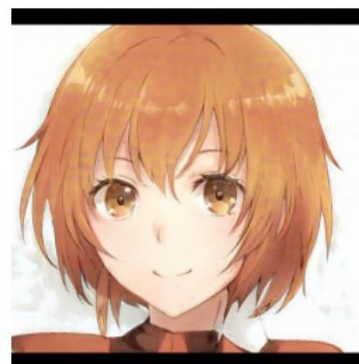
通知タップ・押下時の動作



登録中の場合グレースアウトする



Push&SW Test



通知アクティビティが優秀！

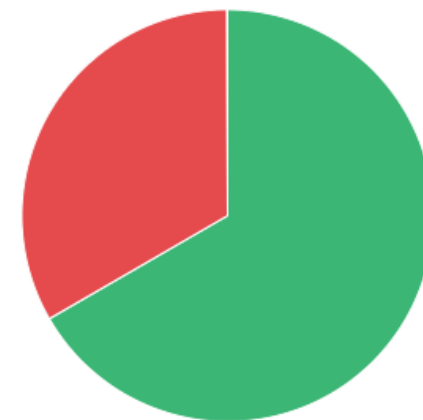
各通知のアナリティクス

通知ごとに

- 到達完了・到達失敗の可否
- 送信数
- 通知の反応率・時間

Delivery Statistics

Total Sent	3
● Delivered ?	2
● Failed (Unsubscribed) ?	1
● Failed (Errored) ?	0
● Remaining ?	0
● Capped ?	0



PWAPushuTest > Delivery

PWA経由でのプッシュ通知が作動しているかのテストです！

Delivered

Actions ▼

Retarget 🔒

Notification created from dashboard. Started sending at: September 11th 2021, 5:03:38 am UTC +09:00 0.46 seconds

Delivered

2

Click-Through Rate

100.00% (2 unique)

通知アクティビティが優秀！

アプリ全体のアナリティクス

- 通知を許可している数
- ユーザー数の推移
- デバイス割合
- MAU

Subscriber Trends

Subscribed Users

3

Monthly Active Users

9

Total Users

9

📅 Last 30 days ▾

📱 All Platforms ▾

Total Users

9 100%

10

5

0

Aug 14

Aug 16

Aug 18

Aug 20

Aug 22

Aug 24

Aug 26

Aug 28

Aug 30

Sep 1

Sep 3

Sep 5

Sep 7

Sep 9

Sep 11

4

ハマったところ・まとめ

ハマったポイント

OneSignalをPWAで使うとき…

OneSignalのSDKファイルにServiceWorkerのファイルを一緒に読み込む必要があった。

JS OneSignalSDKWorker.js U ×

JS OneSignalSDKWorker.js

```
1  importScripts('https://cdn.onesignal.com/sdks/OneSignalSDKWorker.js');
2  importScripts('sw.js');
3  |
```

JS OneSignalSDKUpdaterWorker.js U ×

JS OneSignalSDKUpdaterWorker.js

```
1  importScripts('https://cdn.onesignal.com/sdks/OneSignalSDKWorker.js');
2  importScripts(['sw.js']);
```



ハマったポイント

WorkboxCLIで楽しようとした結果やらかした…

Workboxで迂闊にconfigを見ずに生成してしまったから面倒くさいことになった。

キャッシュ戦略(workbox-strategies)を**handler**で指定してあげないといけなかった。

CacheOnly	基本的にキャッシュのみを参照
NetworkFirst	毎回サーバーにリクエストを飛ばす リクエストが失敗した場合にのみキャッシュされているリソースを参照



```
11
12   runtimeCaching: [
13     {
14       handler: 'NetworkFirst'
15     }
16   ]
```

まとめ

Web Pushは効果的！

自前で作るのもいいけど外部に頼ったが早い＆簡単。

頑張れば**5分**でPWA対応＆Push通知実装できる！！

これでまた一つWebアプリがネイティブアプリに
近づいてきたのではないのでしょうか…

ご清聴ありがとうございました。

以下ページに当スライドを掲載させていただきます！

あ、[Twitter@ichii731](#)も序によろしくお願いします…



ic731.net/docs/0915_PWA.html