

# **Progressive Web Application And Wordpress**

:::::::::::::::::::::

Oslo WordCamp  
3<sup>rd</sup>, March 2018  
Majid Hajian





# Majid Hajian

## ABOUT ME

- Passionate web developer
- Open Source lover and contributor
- Public tech speaker, Organizer
- PacktPub Instructor and trainer
- Wordpress Fan
- Oslo Vue.js meetup Organizer
- Angular and Mobile meetup co-organizer



@mhadaily



# Content Overview

---

## What is really a PWA?

What Progressive web app means?

## Manifest dot JSON

Makes your app installable

## Core Concepts

Different ways that wen can create a PWA  
From Wordpress

## Dive into service worker

What service worker life cycle,  
What is pre-cache ?

## Why PWA? And Essentials

Why you should consider a PWA and do you need?

## Advanced Cache Strategies

Dive into cache strategies and understating  
How they work

# **WHAT REALLY IS A PWA**



**native-app like experience**



# CORE CONCEPTS

.....

# Progressive



Work for every user, regardless of browser choice because they're built with progressive enhancement as a core tenet.

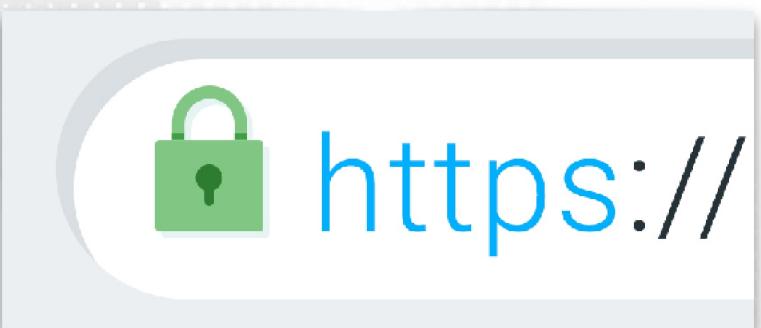


# Responsive

Fit any form factor, desktop, mobile, tablet, or whatever is next.



Served via  
HTTPS to  
prevent  
snooping and  
ensure content  
hasn't been  
tampered with.



**SECURE**



# Connectivity Independent

Enhanced with  
**service workers**  
to work offline or  
on low quality  
networks

# Installable

Allow users to “keep” apps they find most useful on their home screen without the hassle of an app store.



Use the  
app-shell model  
to provide  
app-style  
navigations and  
interactions.

# App-Like

Make  
re-engagement  
easy through  
features like  
push  
notifications.



Re-engageable



# WHY PWA?

.....



# Site Not Found

Why am I seeing this?

There are a few potential reasons:

1. You haven't deployed an app yet.
2. You may have deployed an empty directory.
3. This is a custom domain, but we haven't finished setting it up yet.

How can I deploy my first app?

Refer to our [hosting documentation](#) to get started.



A large, abstract graphic on the left side of the slide features several thick, diagonal bars. These bars are primarily blue, transitioning from dark navy at the bottom to bright cyan at the top. They are set against a white background and overlap each other in a staggered pattern, creating a sense of depth and motion.

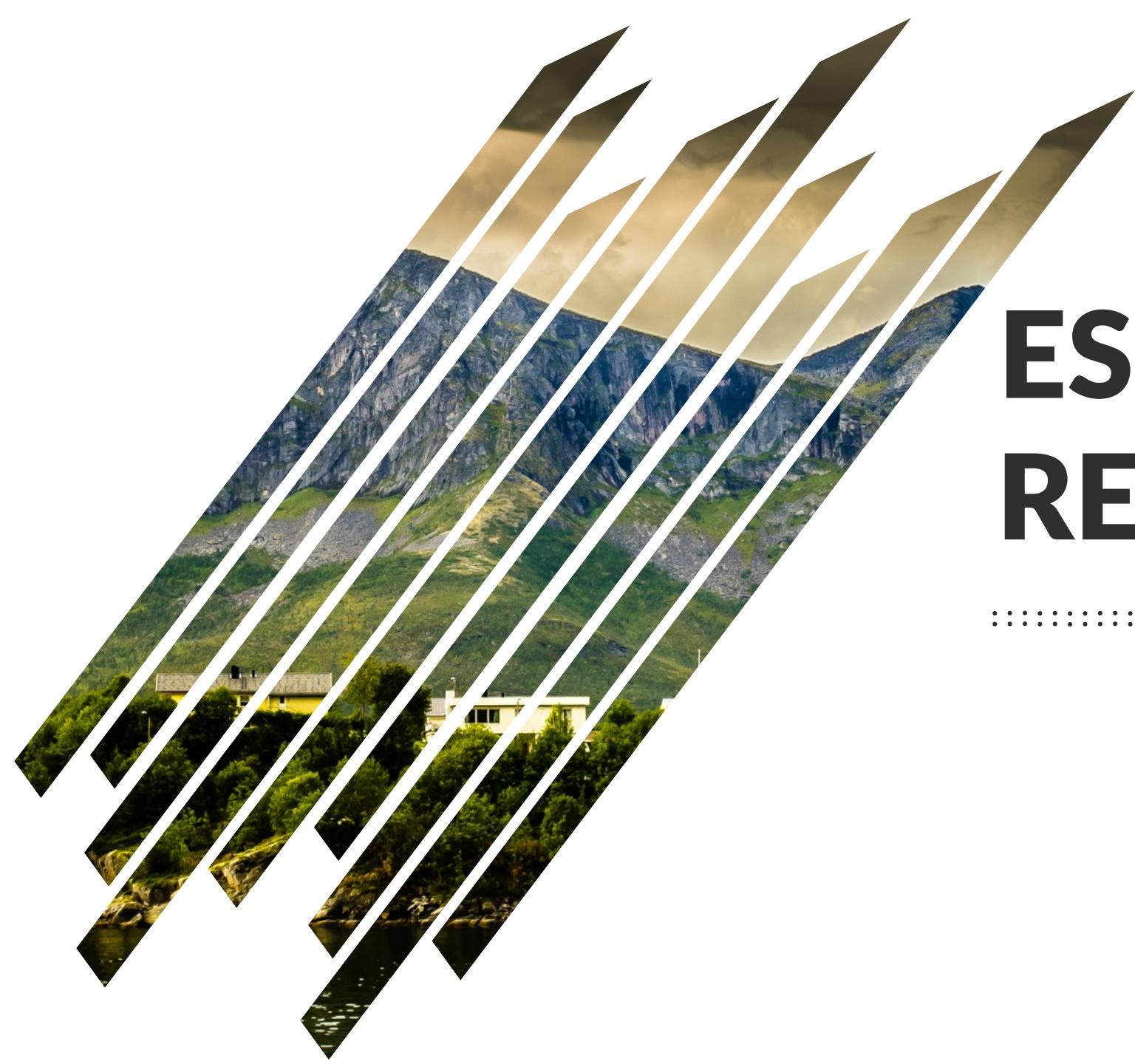
# **AWESOME WORDPRESS**

.....



**SSR**

**WP-REST-API  
SPA**



# **ESSENTIAL REQUIREMENTS**

.....

**manifest.json**



**SW.js**



# MANIFEST DOT JSON

.....

```
<head>
```

```
  <link rel="manifest" href="/manifest.json">
```

OR

```
  <link rel="manifest" href="=$php echo get_manifest_path(); ?&gt;"&gt;</pre
```

```
</head>
```

```
{  
  "name": "WP PWA",  
  "short_name": "WPPWA",  
  "description": "An Awesome Offline App with Wordpress",  
  "background_color": "#fff",  
  "theme_color": "#3f51b5",  
  "start_url": "/index.html?utm_source=homescreen",  
  "scope": ".",  
  "orientation": "portrait-primary",  
  "display": "standalone",  
  "dir": "ltr",  
  "lang": "en-US",  
  "icons": [ {  
      "src": "/assets/images/icons/icon-512x512.png",  
      "type": "image/png",  
      "sizes": "512x512"  
    } ]  
}
```

```
{  
  "name": "WP PWA",  
  "short_name": "WPPWA",  
  "description": "An Awesome Offline App with Wordpress",  
  "background_color": "#fff",  
  "theme_color": "#3f51b5",  
  "start_url": "/index.html?utm_source=homescreen",  
  "scope": ".",  
  "orientation": "portrait-primary",  
  "display": "standalone",  
  "dir": "ltr",  
  "lang": "en-US",  
  "icons": [ {  
      "src": "/assets/images/icons/icon-512x512.png",  
      "type": "image/png",  
      "sizes": "512x512"  
    } ]  
}
```

```
{  
  "name": "WP PWA",  
  "short_name": "WPPWA",  
  "description": "An Awesome Offline App with Wordpress",  
  "background_color": "#fff",  
  "theme_color": "#3f51b5",  
  "start_url": "/index.html?utm_source=homescreen",  
  "scope": ".",  
  "orientation": "portrait-primary",  
  "display": "standalone",  
  "dir": "ltr",  
  "lang": "en-US",  
  "icons": [ {  
      "src": "/assets/images/icons/icon-512x512.png",  
      "type": "image/png",  
      "sizes": "512x512"  
    } ]  
}
```

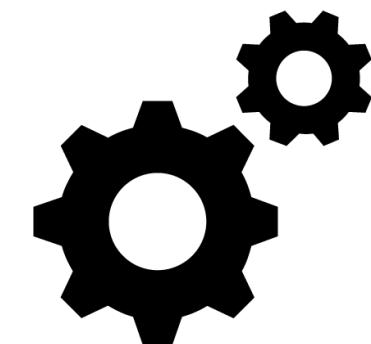
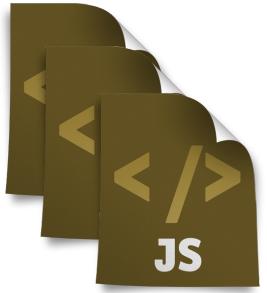


**DEMO TIME**



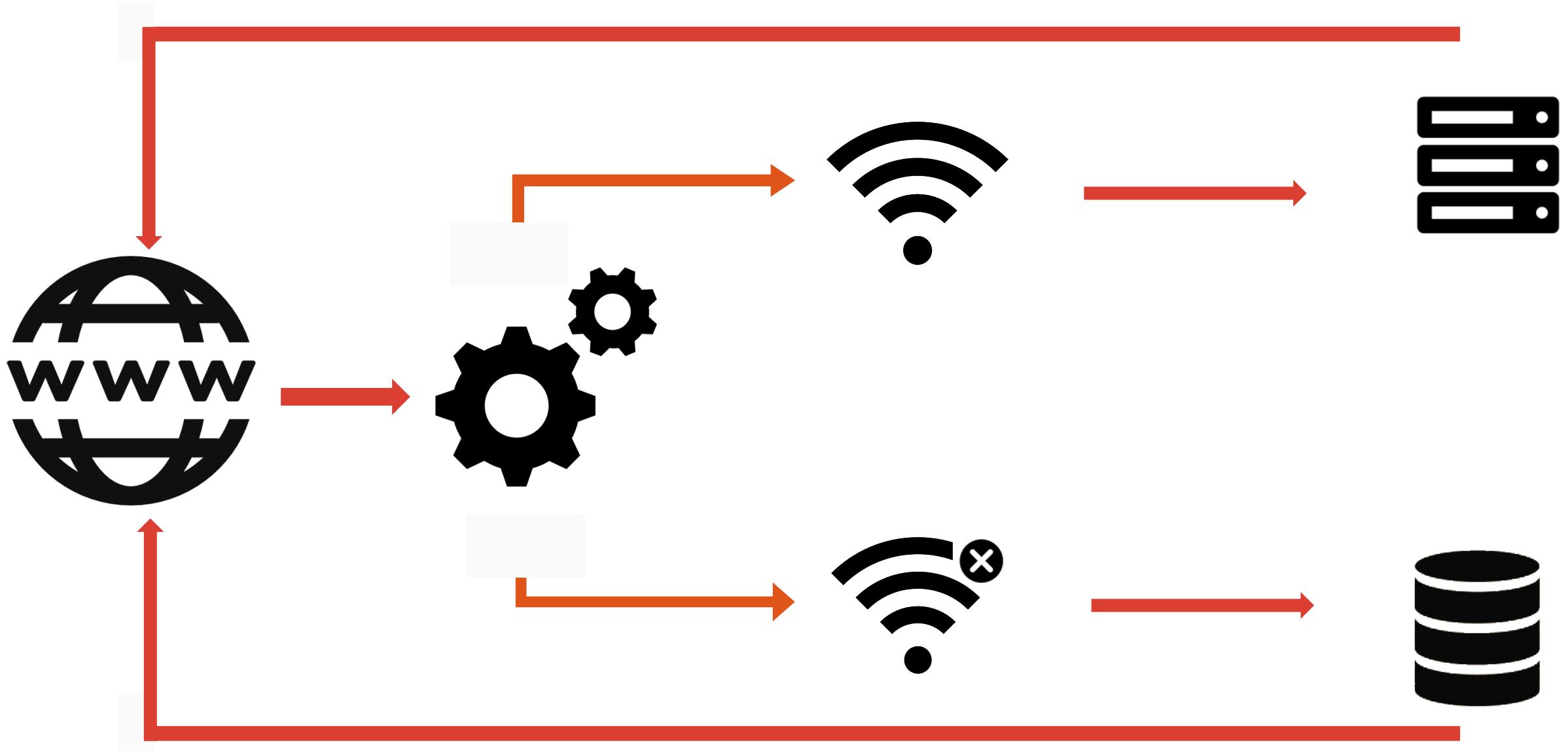
# SERVICE WORKER

.....



Service Worker





# REGISTER SERVICE WORKER

```
if ('serviceWorker' in navigator) {  
  window.addEventListener('load', () => {  
    navigator.serviceWorker.register('/sw.js')  
      .then( (reg) => console.log(reg))  
      .catch((error) => console.log('Registration failed with ' +  
        error));  
  });  
}
```

```
if ('serviceWorker' in navigator) {  
  window.addEventListener('load', () => {  
    navigator.serviceWorker.register('/sw.js', { scope: '/' })  
      .then( (reg) => console.log(reg))  
      .catch((error) => console.log('Registration failed with ' +  
        error));  
  });  
}
```

# SCOPE

# sw.php

```
<?
require(' ../../../../../../wp-load.php');
header('Service-Worker-Allowed: /');
header('Content-Type: application/javascript');
?>

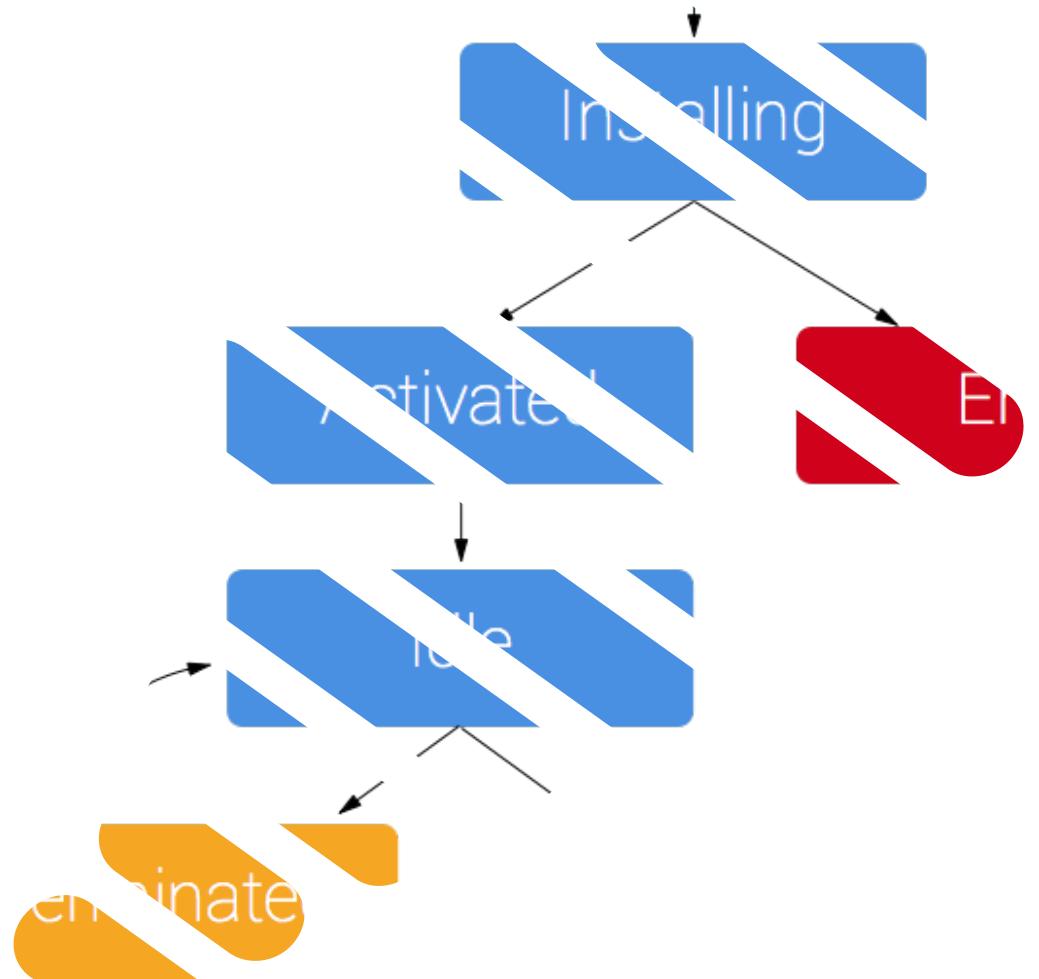
__myconfig={themeUrl:`<?=get_bloginfo('template_url');?>`};

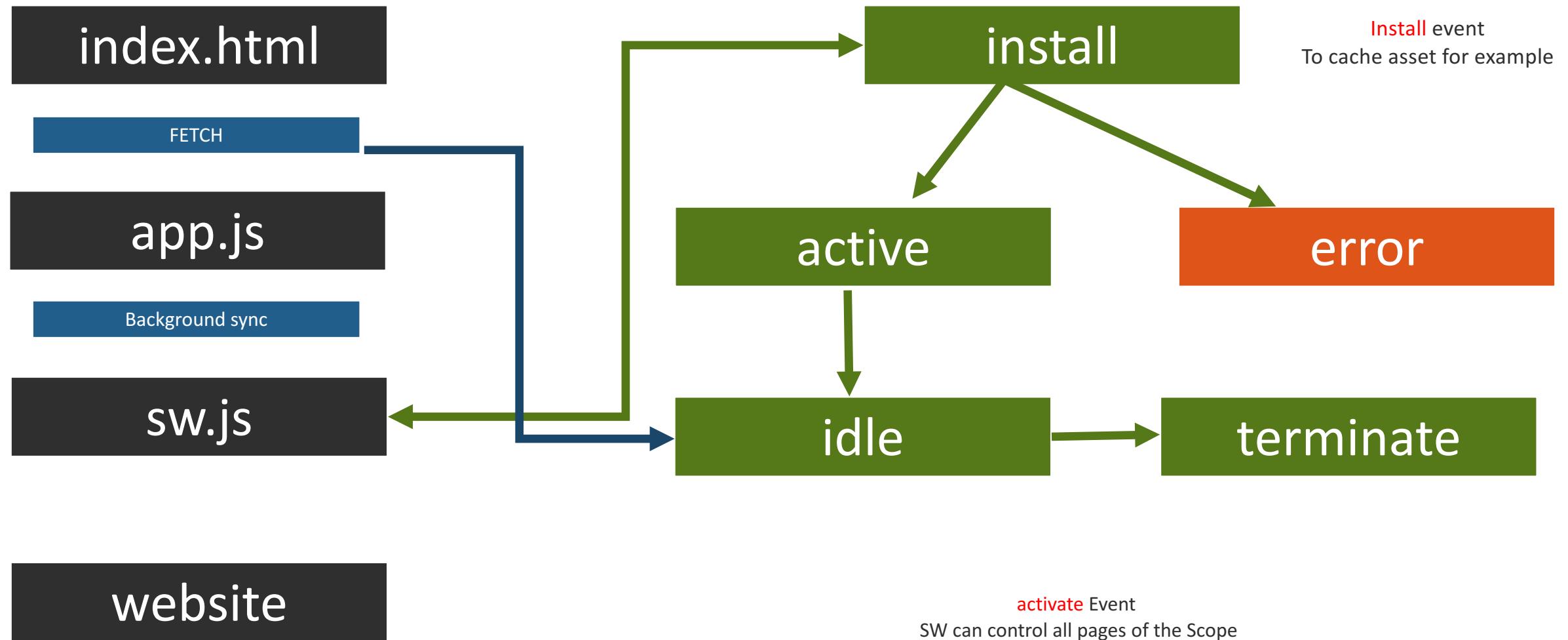
<
  readfile('sw.js');
?>
```

footer.php

```
if ('serviceWorker' in navigator) {  
    window.addEventListener('load', () => {  
        navigator  
            .serviceWorker  
            .register(`$__myconfig.themeUrl}/scripts/sw.php`);  
    });  
}
```

# Service worker Life Cycle







**DEMO TIME**



# CACH STRATEGIES

.....

# FETCH EVENT

---

```
self.addEventListener('fetch',
(event) => {
  console.log('[SW] Fetch ....');
  const request = event.request;
  event.respondWith()
}
```

# FETCH EVENT

---

```
self.addEventListener('fetch',
(event) => {
  console.log('[SW] Fetch ...');
  const request = event.request;
  event.respondWith()
}
```

# Cache only

---

```
event.respondWith(  
  caches.match(request)  
);
```

# Network only

---

```
event.respondWith(  
  fetch(event.request)  
);
```

# Cache first, falling back to network

---

```
event.respondWith(  
  caches.match(request).then( (res) => {  
    return res || fetch(request).then(  
      (newRes) => {  
        caches.open(DYNAMIC_CACHE_VERSION)  
          .then( cache => cache.put(request, newRes) );  
        return newRes.clone();  
      });  
  })  
>);
```

# Cache first, falling back to network

---

```
event.respondWith(  
  caches.match(request).then( (res) => {  
    return res || fetch(request).then(  
      (newRes) => {  
        caches.open(DYNAMIC_CACHE_VERSION)  
          .then( cache => cache.put(request, newRes) );  
        return newRes.clone();  
      });  
  })  
>);
```

# Network first, falling back to cache

---

```
event.respondWith(  
  fetch(request)  
  .then((res) => {  
    caches.open(DYNAMIC_CACHE_VERSION)  
    .then(cache => cache.put(request, res));  
    return res.clone();  
  }) // Fallback to cache  
  .catch(err => caches.match(request))  
);
```

# Network first, falling back to cache

---

```
event.respondWith(  
  fetch(request)  
  .then((res) => {  
    caches.open(DYNAMIC_CACHE_VERSION)  
    .then(cache => cache.put(request, res));  
    return res.clone();  
  }) // Fallback to cache  
  .catch(err => caches.match(request))  
);
```

# Cache with Network Update

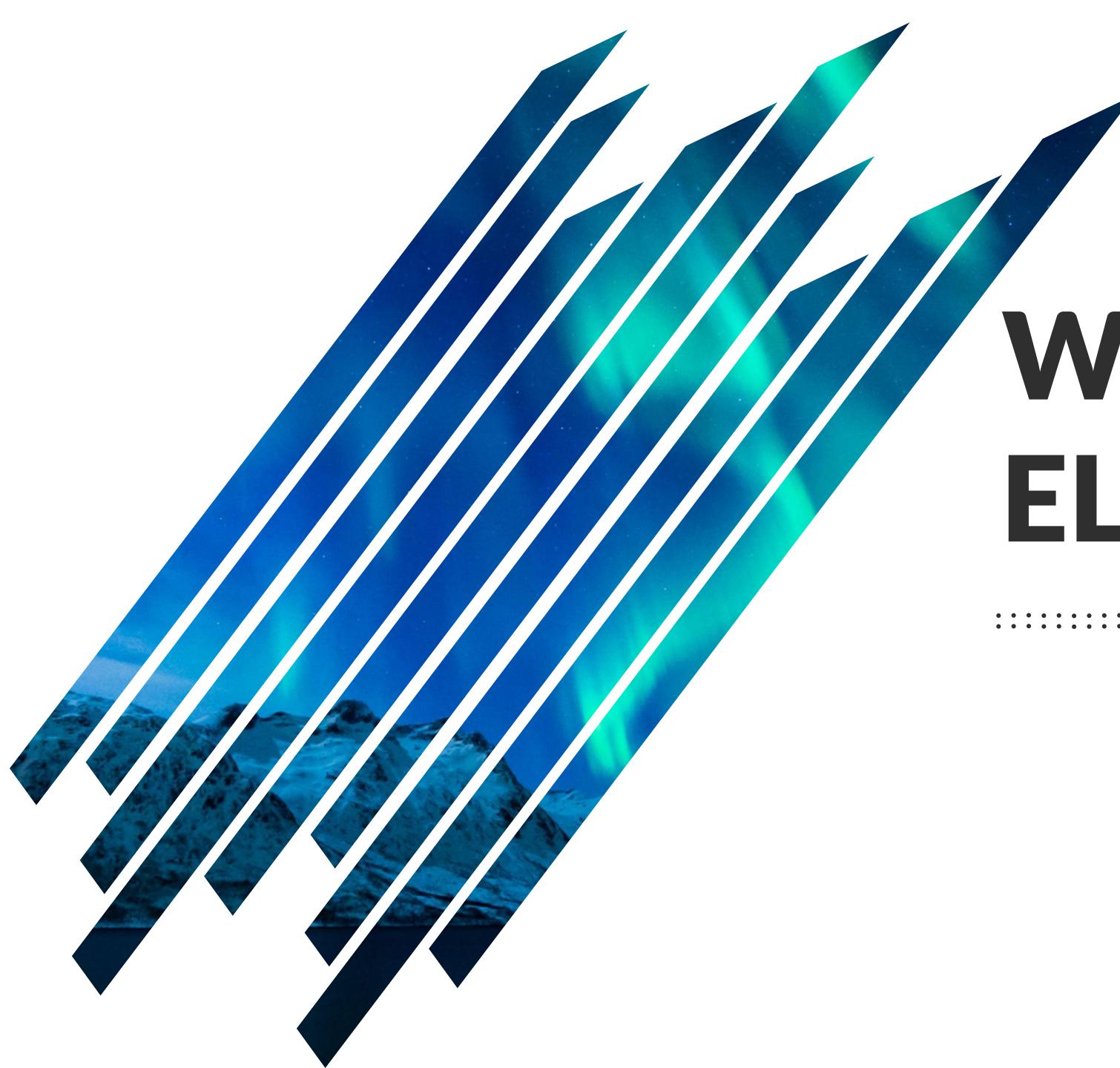
---

```
event.respondWith(  
  caches  
    .match(request).then((res) => {  
      const updatedResponse =  
        fetch(request).then((newRes) => {  
          cache.put(request, newRes.clone());  
          return newRes;  
        });  
      return res || updatedResponse;  
    })  
);
```

# Cache with Network Update

---

```
event.respondWith(  
  caches  
    .match(request).then((res) => {  
      const updatedResponse =  
        fetch(request).then((newRes) => {  
          cache.put(request, newRes.clone());  
          return newRes;  
        });  
      return res || updatedResponse;  
    })  
>;
```



**WHAT  
ELSE...**

.....

# Web Share API

the `Navigator.share()` method invokes the native sharing mechanism of the device as part of the Web Share API.

# Battery Status

The **Battery Status API** allows Web applications to get the information about the device's power source, battery charge level, expected time of charging or discharging.

# Push Notification

Push Messaging is the well-known re-engagement mechanism from the mobile platforms

# Background Sync

Reigster a task and sync back your data when Connectivity restablished

# Payment Request API

allows Web applications to delegate the payment checkout process to the operating system, allowing it to use whatever methods and payment providers are natively available for the platform and configured for the user.

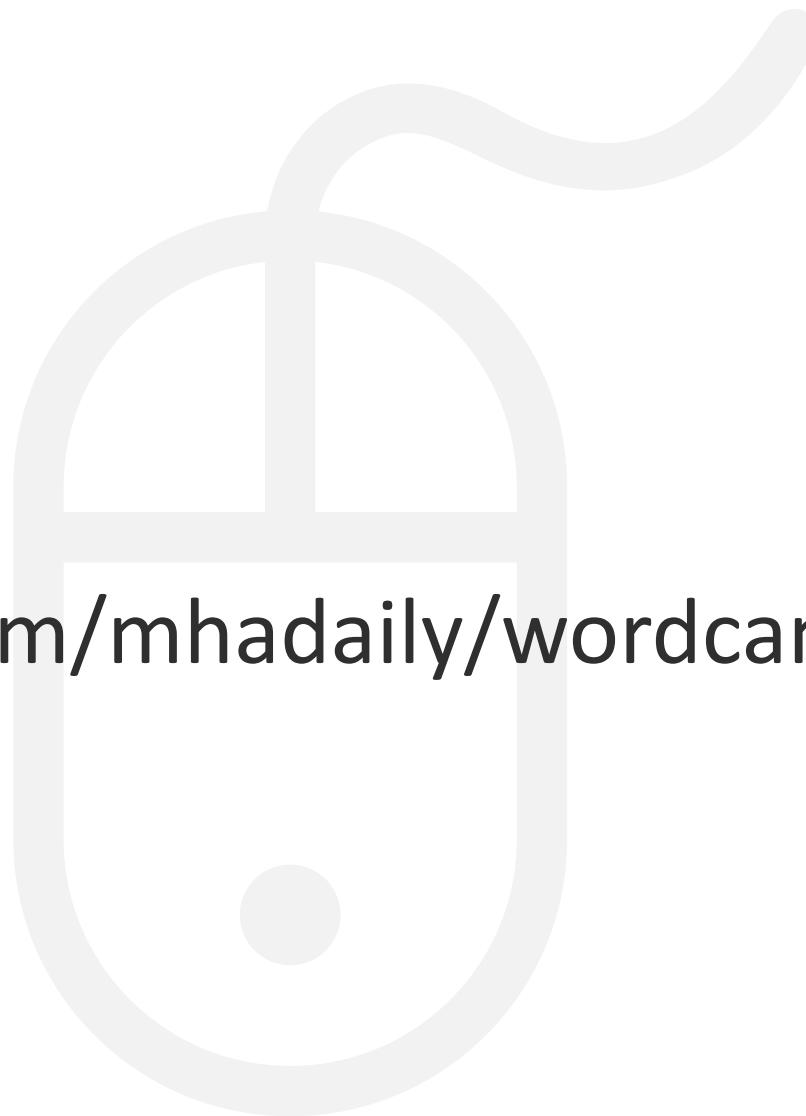
# Vibration API

Give your user the best experience with progressive enhancement

# Credential Management

The **Credential Management API** allows authorized Web applications to store and request user credentials (like login and password or federated login data) programmatically on behalf of the user. The API offers a replacement for browser built-in or 3rd-party password stores

**AND EVEN MORE...**



<https://github.com/mhadaily/wordcamp2018Oslo>

---

# THANK YOU

See You Next Time

---

MAJID HAJIAN

---



@mhadaily