🌊Web Push Benachrichtigungen einrichten

Schritt für Schritt (PHP, ohne CMS)

Lutz Neumeier · Oktober 2025

Diese Anleitung führt von Null zur ersten Push-Benachrichtigung – inklusive iPhone-Sonderregeln, vollständigen Code-Snippets und genauen Einfügepunkten.

Inhalt

---

0) Voraussetzungen & Konzept 🧠

- **HTTPS** ist Pflicht (sonst funktionieren Service Worker & Push nicht).
- **Web Push ohne Account**: Nutzer brauchen keinen Login – sie müssen aber *einmal* zustimmen.
- **iPhone/iPad**: Web Push funktioniert ab iOS/iPadOS 16.4 *nur als installierte Web-App* (über „Zum Home-Bildschirm“), nicht im normalen Safari-Tab.

---

1) Ordnerstruktur 📁

```
/dein-ordner/
├─ index.php
├─ upload.php
├─ sw.js                  <— Service Worker (im selben Ordner
├─ subscribe.php          <— Abo-Endpunkt
├─ vapid_public.key       <— Public (Base64-URL)
├─ vapid_private.key      <— Private (geheim)
├─ vendor/                <— aus Composer (Minishlink/WebPush)
└─ subscribers.jsonl      <— wird automatisch erzeugt
```

---

2) Lokale Vorbereitung am Mac (Composer) 🧰

```
mkdir -p ~/Desktop/webpush-setup
cd ~/Desktop/webpush-setup
composer require minishlink/web-push
```

Ergebnis: Der Ordner `vendor/` entsteht lokal – den lädst du später auf den Server hoch.

---

3) VAPID-Schlüssel erzeugen 🔑

Lege lokal eine Datei `gen_keys.php` an:

```
<?php
require __DIR__.'/vendor/autoload.php';
use Minishlink\WebPush\VAPID;
$keys = VAPID::createVapidKeys();
file_put_contents(__DIR__.'/vapid_public.key',
$keys['publicKey']); file_put_contents(__DIR__.'/
vapid_private.key', $keys['privateKey']); echo "Public:
\n{$keys['publicKey']}\n\nPrivate:\n{$keys['privateKey']}\n";
```

Ausführen:

```
php gen_keys.php
```

Es entstehen `vapid_public.key` (einzeiliger Base64-URL-String) und `vapid_private.key` (geheim). *Lösche* danach `gen_keys.php`.

---

4) Upload auf den Server ⬆️

Lade in den Web-Ordner mit deiner `index.php` hoch:

- den kompletten `vendor/`-Ordner
- `vapid_public.key` und `vapid_private.key`
- `sw.js` und `subscribe.php`

---

5) Service Worker `sw.js` 🔔

```
self.addEventListener('push', (event) => {
  let data = {};
  try { data = event.data ? event.data.json() : {}; } catch(e
  const title = data.title || 'Neuer Beitrag';
  const options = {
    body:  data.body  || 'Es gibt neue Inhalte in der Galerie
    icon:  data.icon  || 'icons/icon-192.png',
    badge: data.badge || 'icons/badge-72.png',
    data: { url: data.url || './' }
  };
  event.waitUntil(self.registration.showNotification(title, o
});

self.addEventListener('notificationclick', (event) => {
  event.notification.close();
  const url = event.notification.data?.url || './';
  event.waitUntil(clients.matchAll({type:'window', includeUnc
    for (const c of list) { if ('focus' in c) { c.navigate(ur
    return clients.openWindow(url);
  }));
});
```

6) Abo-Endpunkt `subscribe.php` 📨

```
<?php
header('Content-Type: application/json; charset=utf-8');
try {
  if ($_SERVER['REQUEST_METHOD'] !== 'POST') { http_response_
  $body = file_get_contents('php://input');
  if (!$body) { http_response_code(400); echo json_encode(['o
  $sub = json_decode($body, true, 512, JSON_THROW_ON_ERROR);
  if (!is_array($sub) || empty($sub['endpoint'])) { http_resp

  $file = __DIR__ . '/subscribers.jsonl'; $entries = [];
  if (file_exists($file)) {
    $lines = file($file, FILE_IGNORE_NEW_LINES | FILE_SKIP_EM
    foreach ($lines as $ln) { $o=json_decode($ln,true); if (!
  }
  $entries[$sub['endpoint']] = json_encode($sub, JSON_UNESCAP

  $tmp=$file.'.tmp';
  $ok = file_put_contents($tmp, implode("\n",$entries)) !== f
  if (!$ok) { http_response_code(500); echo json_encode(['ok'

  echo json_encode(['ok'=>true]);
} catch (Throwable $e) {
  http_response_code(500); echo json_encode(['ok'=>false,'err
}
```

7) Frontend (index.php): Button & Abo-Logik 🔔

7.1 HTML unter dem Header

```
<div id="notify-wrap" class="notify-container" style="display
  <button id="notify-btn" class="btn" type="button">🔔 Benach
  <div id="notify-status" style="margin-top:6px; font-size:0.
</div>
```

7.2 JS am Ende der Seite (vor </body>)

```
<script>
function relUrl(name){ return new URL(name, location.href).to

(async () => {
  const wrap=document.getElementById('notify-wrap'), btn=docu
  const https=location.protocol==='https:', swOK='serviceWork
  const ua=navigator.userAgent||''; const isApple=/iPad|iPhon
  const isStandalone=matchMedia('(display-mode: standalone)')
  let supported=https && swOK && pushOK && notif; if (isApple
  if (!supported) return;

  let reg; try{
    reg = await (navigator.serviceWorker.getRegistration(loca
    if (!reg) reg = await navigator.serviceWorker.register(re
    await navigator.serviceWorker.ready;
  }catch(_){ return; }

  if (Notification.permission==='denied') return;
  try{ if (await reg.pushManager.getSubscription()) return; }

  wrap.style.display='block';
  btn.addEventListener('click', enablePushSafe);

  async function enablePushSafe(){
    try{
      const perm=await Notification.requestPermission(); if(p
      const regReady=await navigator.serviceWorker.ready;
      const vapidKey=(await (await fetch(relUrl('vapid_public
      const appServerKey=urlBase64ToUint8Array(vapidKey); if(
      const sub=await regReady.pushManager.subscribe({userVis
      const res=await fetch(relUrl('subscribe.php'),{method:'
      const data=await res.json().catch(()=>({})); if(!res.ok
      status.textContent='✅ Benachrichtigungen aktiviert'; v
    }catch(e){ console.error(e); status.textContent='⚠️ Fehle
  }
```

```
  function urlBase64ToUint8Array(s){const p='='.repeat((4-s.l
})();
</script>
```

---

8) Push beim Upload automatisch versenden (upload.php) 🚀

8.1 Autoload & Helfer ganz oben

```
<?php
require __DIR__ . '/vendor/autoload.php';
use Minishlink\WebPush\WebPush;
use Minishlink\WebPush\Subscription;

function item_id(array $it): string {
  $key = ($it['file'] ?? '') . '|' . ($it['time'] ?? 0) . '|'
  return substr(sha1($key), 0, 12);
}

function send_push_new_post(array $it): void {
  try {
    $pub=@trim((string)file_get_contents(__DIR__.'/vapid_publ
    $priv=@trim((string)file_get_contents(__DIR__.'/vapid_pri
    if(!$pub||!$priv) return;
    $subsFile=__DIR__.'/subscribers.jsonl';
    if(!file_exists($subsFile)) return;
    $lines=file($subsFile, FILE_IGNORE_NEW_LINES|FILE_SKIP_EM

    $wp=new WebPush(['VAPID'=>['subject'=>'mailto:info@deine-
    $caption=trim((string)($it['caption']??'')); $title=$capt
    $id=item_id($it);
    $scheme=(!empty($_SERVER['HTTPS']) && $_SERVER['HTTPS']!=
    $host=$_SERVER['HTTP_HOST']??'localhost';
    $base=rtrim(dirname($_SERVER['SCRIPT_NAME']??'/'),'/\\');
    $url=$scheme.'://'.$host.$base.'/index.php#'.$id;
    $payload=json_encode(['title'=>$title,'body'=>'Neuer Post
```

```
    foreach($lines as $ln){ $sub=json_decode($ln,true); if(!i
    foreach($wp->flush() as $r){/* ignore */}
  } catch(Throwable $e){ /* stumm */ }
}
```

8.2 Exakte Einfügepunkte (vor jedem Redirect) 📌

In deiner `upload.php` gibt es drei „Erfolgszweige“. **Füge vor jedem** `header('Location: ...'); exit;` **diesen Block ein**:

```
// Push auslösen
$last = $items[array_key_last($items)];
send_push_new_post($last);

header('Location: '.$_SERVER['PHP_SELF'].'?ok=1');
exit;
```

Variante A – Bild mit Crop:

```
if (@file_put_contents($jsonFile, json_encode($items, ...)
  // Fehler
} else {
  // Push auslösen
  $last = $items[array_key_last($items)];
  send_push_new_post($last);

  header('Location: '.$_SERVER['PHP_SELF'].'?ok=1');
  exit;
}
```

Variante B – Video (mit/ohne Poster):

```
if (@file_put_contents($jsonFile, json_encode($items, ...)
  // Fehler
} else {
```

```
  // Push auslösen
  $last = $items[array_key_last($items)];
  send_push_new_post($last);

  header('Location: '.$_SERVER['PHP_SELF'].'?ok=1');
  exit;
}
```

Variante C – Bild ohne Crop (Fallback):

```
if (@file_put_contents($jsonFile, json_encode($items, ...)
  // Fehler
} else {
  // Push auslösen
  $last = $items[array_key_last($items)];
  send_push_new_post($last);

  header('Location: '.$_SERVER['PHP_SELF'].'?ok=1');
  exit;
}
```

---

9) iPhone/iPad Besonderheiten 🍏

- Seite in Safari öffnen → `Teilen` → `Zum Home-Bildschirm` → **über das Icon** starten.
- iOS/iPadOS ≥ 16.4 erforderlich.
- Mitteilungen erlauben: Einstellungen → Mitteilungen → <App-Name> → erlauben.
- Nach größeren Änderungen Web-App neu installieren (Icon löschen → neu hinzufügen).

---

10) Fehlerbehebung & Checkliste ✅

- **404 auf subscribe.php**: Pfad relativ verwenden (z. B. `relUrl('subscribe.php')`). Direkt im Browser aufrufen → erwartetes

Ergebnis ist *405 Method not allowed*.

- **VAPID Key Fehler**: `vapid_public.key` muss ein einzeiliger Base64-URL-String sein (nur A-Z a-z 0-9 – _), kein PEM-Header.
- **Service Worker**: `sw.js` muss im selben Ordner wie `index.php` liegen und als solcher registriert werden (Scope!).
- **HTTPS** aktiv?
- **iPhone**: läuft die Seite als Web-App (Homescreen)?
- **Schreibrechte**: Ordner muss `subscribers.jsonl` anlegen dürfen.

Anhang: Komplettcodes 📚

sw.js

```
// (siehe Kapitel 5)
```

subscribe.php

```
// (siehe Kapitel 6)
```

Enable-Button & Abo-Logik

```
// (siehe Kapitel 7.1 und 7.2)
```

upload.php – Autoload & Push

```
// (siehe Kapitel 8.1 und 8.2)
```
