

Abschlussarbeit

Inhaltsverzeichnis

Inhaltsverzeichnis	2
2 Einführung	4
2.1 Ausgangslage	4
2.2 Aufgabenstellung / Ziele / Organisation / Hilfsmittel	4
4 Projektplanung	5
4.1 Zeitplanung	5
4.2 Testfälle erstellen	5
4.3 Service Workers, was muss man wissen	5
5 Realisation	6
5.1 Basis	6
5.2 Request	6
5.3 Urheberrecht Songkick	7
5.4 Manifest	7
5.5 Warum Google Chrome	8
5.6 HTTPS	9
5.7 Service Worker	9
5.7.1 Grundarchitektur:	10
5.7.2 Registration des Service Workers in pwajs	10
5.7.3 Kontrolle: Wurde der SW registriert?	11
5.7.4 Installation und Aktivierung: Füllen des Caches:	12
5.7.5 Hierarchie der Files(spielt eine grosse Rolle und wird schlecht beschrieben)	13
5.8 Optimierung	14
5.8.1 Fehlgeschlagene Cache-Anfragen abfangen	14
5.9 Optimierung der Usability	14
5.9.1 Push Notifications	15
5.10 Weitere Funktionen einbauen für bessere Usability	16
5.11 Design	16
6 Testing	16

7	Fazit / Auswertung / Zielerreichung / Zielüberprüfung	18
8	Selbständigkeitserklärung.....	19
9	Glossar / Verzeichnisse	20
10	Anhang.....	21

2 Einführung

2.1 Ausgangslage

2.2 Aufgabenstellung / Ziele / Organisation / Hilfsmittel

4 Projektplanung

4.1 Zeitplanung

Planung Use Case Diagramm, Anwendungsfalldiagramm, Gantt-Diagramm

Testfälle

4.2 Testfälle erstellen

4.3 Service Workers, was muss man wissen

Die Recherchen bezüglich des Requests haben mich etwas verwirrt, es stand geschrieben, dass der normale Ajax Request nicht unterstützt wird. Damit habe ich automatisch an den Request gedacht, mit dem ich die Daten hole. Aus Vorsorge habe ich dann alles mit einem fetch-Request gemacht. Bis ich dann später herausgefunden habe, dass damit der Service Worker selbst gemeint ist und dass der die Daten nur so cachen kann. Ich habe es dann trotzdem noch ausprobiert, die Daten mit dem normalen Ajax zu holen und es funktioniert alles gut.

5 Realisation

5.1 Basis

Um eine PWA zu erstellen, haben sich 3 Techniken durchgesetzt die man benötigt: HTML5, CSS und Javascript. Somit brauchen wir ein HTML-File für die Darstellung, ein CSS-File für das Styling und ein JS für den Request. Weiter brauchen wir dann noch ein JS für den Service Worker und ein Manifest-JSON.

5.2 Request

Ajax Request und fetch Request. Erweiterung um Suchfunktion.

Upcoming Events of Angerfist

- REVOLT 2019, 2019-05-25
Bad Kreuznach, Germany
Type: Festival
[more infos](#)
- World Club Dome 2019, 2019-06-07
Frankfurt, Germany
Type: Festival
[more infos](#)
- Harmony Of Hardcore 2019, 2019-06-08
Erp, Netherlands
Type: Festival
[more infos](#)
- Open Beatz 2019, 2019-07-18
Herzogenaurach, Germany
Type: Festival
[more infos](#)

(Ergebnis des Requests)

```
artistid 144843  
result  
▼(4) [{...}, {...}, {...}, {...}] ⓘ  
  ▼0:  
    ageRestriction: null  
    displayName: "REVOLT 2019"  
    ▶end: {date: "2019-05-25", time: null, datetime: null}  
      flaggedAsEnded: false  
      id: 37349469  
    ▶location: {city: "Bad Kreuznach, Germany", lat: 49.8431, lng: 7.86331}  
    ▶performance: (52) [{...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}  
      popularity: 0.00478  
    ▶series: {displayName: "REVOLT"}  
    ▶start: {date: "2019-05-25", datetime: null, time: null}  
      status: "ok"  
      type: "Festival"  
      uri: "http://www.songkick.com/festivals/73091-revolt/id/37349469-revolt-2019?utm_source=..."  
    ▶venue: {id: null, displayName: "Unknown venue", uri: null, metroArea: {...}, lat: null, lng: null}  
      __proto__: Object  
    ▶1: {id: 35322974, displayName: "World Club Dome 2019", type: "Festival", uri: "http://www.songkick.com/events/35322974-world-club-dome-2019?utm_source=..."}  
    ▶2: {id: 37047509, displayName: "Harmony Of Hardcore 2019", type: "Festival", uri: "http://www.songkick.com/events/37047509-harmony-of-hardcore-2019?utm_source=..."}  
    ▶3: {id: 37413254, displayName: "Open Beatz 2019", type: "Festival", uri: "http://www.songkick.com/events/37413254-open-beatz-2019?utm_source=..."}  
      length: 4  
      __proto__: Array(0)
```

Erweiterter Fetch mit Suche (und normaler Ajax) und Suchfeld etc mit keyupt / click / timeout...
tolowercase and replace spaces with dashes. Weil die wuche nur so funktioniert.

Modifications der Daten, etc. Daten laden und ausgeben.

5.3 Urheberrecht Songkick

Die Daten die ich hole, kommen von songkick. Sie stellen die API zur Verfügung und bei Anfrage bekommt man den API-Key. Im Gegenzug dafür, dass sie die Daten bereitstellen, muss man bei jedem Event auf die Originalseite verlinken und ihr Logo einfügen. Die Daten sind als JSON oder XML verfügbar und man kann diverses auslesen, nächste Events, vergangene Events oder einfach über den Künstler selbst, etc. Ebenfalls liefern sie die Logos und Banners als SVG oder PNG.

5.4 Manifest

Das Manifest ist nur eine JSON-Datei, die alle Metadaten der Progressiven Web-App beschreibt. Informationen wie Titel, Farben, Icon, Start-URL etc. Diese Information zeigt Browsern an, wie sie die Anwendung anzeigen können, wenn man sie als Verknüpfung speichert.

```
{
  "name": "songkick events",
  "short_name": "sk-events",
  "start_url": "https://cyon.netlive.ch/pwa/",
  "display": "standalone",
  "background_color": "#F31C56",
  "description": "get upcoming events",
  "theme_color": "#F31C56",
  "icons": [
    {
      "src": "img/sk-badge-pink.png",
      "sizes": "512x512",
      "type": "image/png"
    }
  ]
}
```

Das File ist ein JSON und wird in den links oben im HTML Head eingebunden:

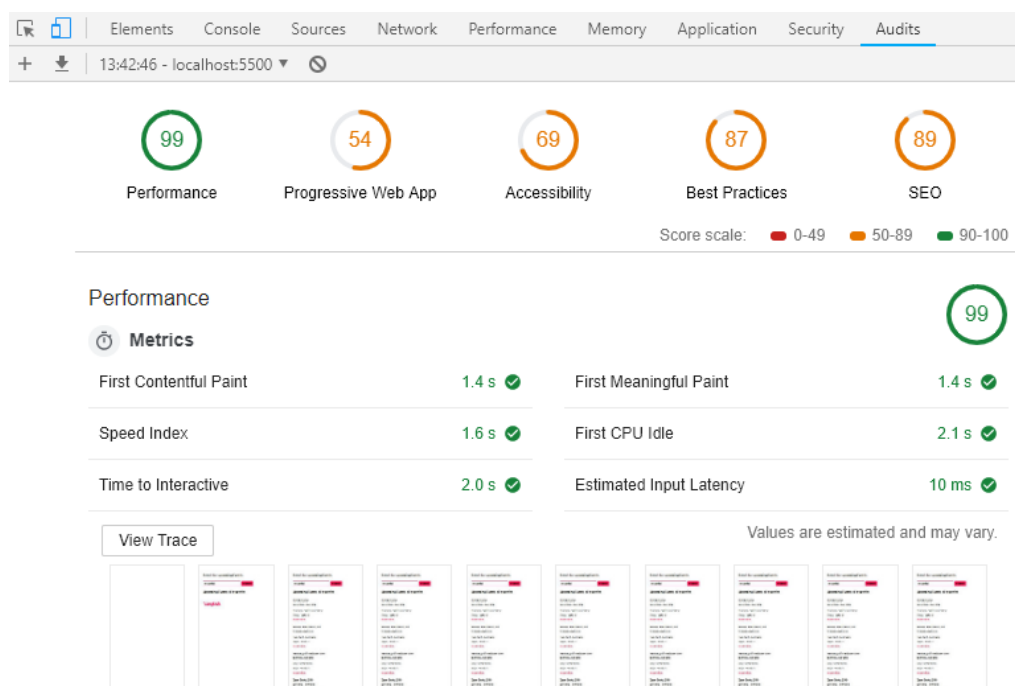
```
<link rel="manifest" href="css/manifest.json">
```

5.5 Warum Google Chrome

Google Chrome verfügt über praktische Tools, die einem bei solchen Anwendungen gut helfen. Zum einen gibt es hier eine mobile Ansicht, das heisst, ich sehe die Seite wie sie auf dem Smartphone oder Tablet aussehen würde zum anderen sind das die vielen hilfreichen Tools.

Sehr nützlich ist das Tool Audits (Entwicklerumgebung, dann oben im Menu zu Audits wechseln), lässt man die Anwendung überprüfen, so gibt es eine Analyse zurück und sagt einem, wie die Performance ist oder was noch fehlt für eine Progressive Web App.

Folgendes Ergebnis ist gekommen, als ich noch kein Service Worker programmiert habe:



Weiter steht dann zu jeder überprüften Eigenschaft in Details, was noch fehlt, beziehungsweise was nicht funktioniert.

Progressive Web App

These checks validate the aspects of a Progressive Web App, as specified by the baseline [PWA Checklist](#).

54

1	Does not respond with a 200 when offline	▲ ▼
2	User will not be prompted to Install the Web App Failures: Site does not register a service worker.	▲ ▼
3	Does not redirect HTTP traffic to HTTPS	▲ ▼
4	Does not register a service worker	▲ ▼
5	Address bar does not match brand colors Failures: No `<meta name="theme-color">` tag found.	▲ ▼
🔍	Additional items to manually check	3 audits ▼
✓	Passed audits	7 audits ▼

5.6 HTTPS

Ursprünglich wollte ich den Web Server von Chrome benutzen, aber da dieser HTTPS ist, wie ich dann herausgefunden habe, musste ich eine andere Lösung finden.

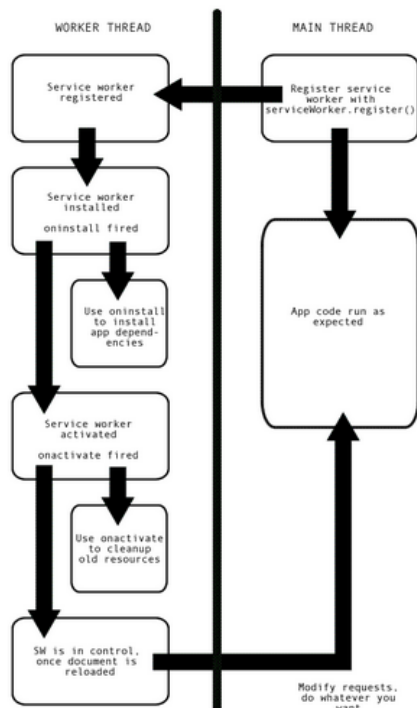
5.7 Service Worker

Hier ist alles Magie. Service worker ist im wesentlichen ein Javascript-Teil, der als Vermittler zwischen dem Browser und dem Host fungiert. Es wird automatisch in unterstützten Browsern installiert, kann Anfragen, die an Ihre Website gesendet werden, abfangen und anders reagieren.

Man kann einen neuen SW einrichten, indem man einfach eine Javascript-Datei im Stammverzeichnis des Projekts erstellt. In unserem Fall heisst sie sw.js. Der Inhalt dieser Datei hängt davon ab, was man erreichen möchte.

Es gibt viele coole Dinge, die mit Service Workern gemacht werden können. Eine davon ist die Möglichkeit, Inhalte zwischenspeichern, lokal zu speichern und somit verfügbar zu machen, wenn sich der Benutzer im Offline-Modus befindet. Selbst wenn es Online ist, wird dies die Ladezeit der Seite erheblich beeinflussen, da Anfragen das Netzwerk einfach umgehen können und der Zugriff auf die Assets sofort erfolgt.

Neben dem traditionellen Browser-Caching kann man bei Service Workern eine Liste von Ressourcen für das Caching definieren.



5.7.1 Grundarchitektur:

URL des SW aufgerufen via:

```
ServiceWorkerContainer.register()
```

//überarbeiten, text über die Funktion eines SW

War dies erfolgreich, wird der Service-Worker in einem `ServiceWorkerGlobalScope` ausgeführt; Im Prinzip ist dies eine besondere Art von Worker-Kontext, welcher außerhalb des Threads des Hauptskripts läuft und keinen Zugriff auf das DOM hat.

Der Service-Worker kann nun Events verarbeiten.

Die Installation des Workers wird versucht, wenn Seiten, die durch Service-Worker verwaltet werden, wiederholt besucht werden. Ein Install-Event ist immer das erste, welches an einen Service-Worker gesendet wird (Dies kann beispielsweise benutzt werden, um den Prozess zu starten, der die IndexedDB

befüllt und Seiten-Assets cacht). Das ist der gleiche Ablauf, der bei der Installation einer nativen oder FirefoxOS App stattfindet - alles für die Offline-Nutzung vorbereiten.

Sobald der `oninstall`-Handler abgeschlossen ist, wird der Service-Worker als installiert betrachtet.

Der nächste Schritt ist Aktivierung. Sobald der Service-Worker installiert wurde, erhält dieser ein `activate`-Event. Der Hauptnutzen von `onactivate` ist das Aufräumen von Ressourcen, die in vorherigen Versionen des Service-Worker-Skripts genutzt wurden.

Der Service-Worker kann nun Webseiten verwalten, aber nur wenn diese nach dem erfolgreichen Abschluss von `register()` aufgerufen wurden. So wird ein Dokument, welches ohne Service-Worker gestartet wurde bis zu einem Neuladen nicht von einem Service-Worker verwaltet werden.

5.7.2 Registration des Service Workers in pwa.js

```
if ('serviceWorker' in navigator){
  navigator.serviceWorker.register('js/sw.js', {scope: 'js/sw.js'}).then(function(reg){
    //registration successfull
    console.log('SW registered! Scope is ', reg.scope);
  }).catch(function(err){
    //registration failed
    console.log('registration failed with ', +err);
  })
}
```

//überarbeiten (beschreibung zum Codeteil von oben:

Der äußere Block kümmert sich um die Feature-Erkennung um sicher zu stellen, dass Service-Worker unterstützt werden, bevor einer registriert wird.

Als nächstes nutzen wir die `ServiceWorkerContainer.register()`-Funktion, um den Service Worker für diese Seite zu registrieren. Dieser ist nur eine JavaScript-Datei innerhalb unserer App. (Beachte, dass die URL der Datei relativ zum Ursprung ist, und nicht zur JavaScript-Datei, die sie referenziert.)

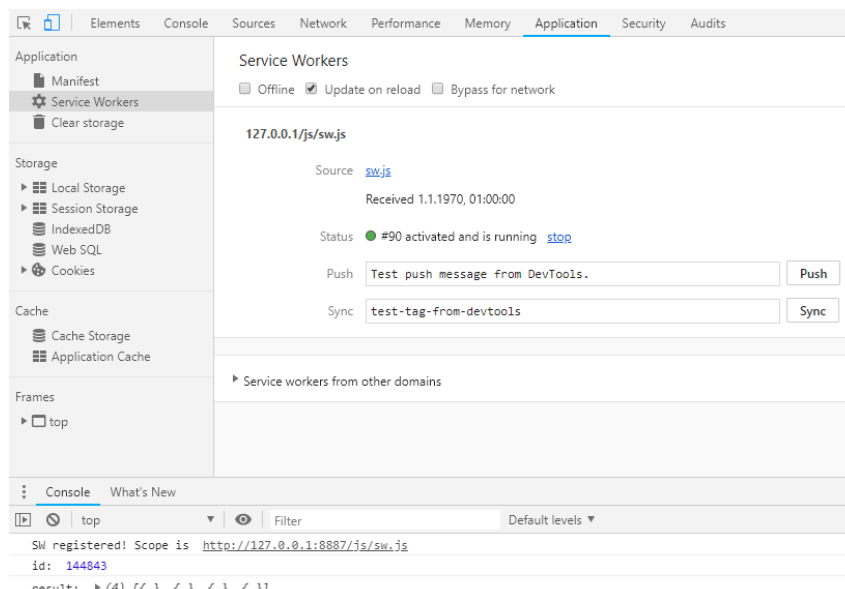
Der `scope` Parameter ist optional, und kann benutzt werden, um den Teil der Applikation einzuschränken, den der Service-Worker kontrollieren soll. In diesem Fall, haben wir `'/sw-test/'` spezifiziert, was alle Inhalte unter dem App-Ursprungsverzeichnis beinhaltet. Wird der Parameter weggelassen, würde standardmäßig dieses Verzeichnis gewählt werden. Wir haben es hier zu Anschauungszwecken jedoch spezifiziert..

Die `.then()` Promise-Funktion wird genutzt, um den Erfolgsfall an unsere Promise-Struktur zu hängen. Wenn die Promise erfolgreich auflöst, wird der Programmcode innerhalb dieser Funktion ausgeführt.

Zuletzt fügen wir die `.catch()`-Funktion ans Ende an, die ausgeführt wird, sollte die Promise zurückgewiesen werden.

5.7.3 Kontrolle: Wurde der SW registriert?

In Google Chrome in der Entwicklerumgebung bei «Application», gibt es ein Register Service Workers.



Dort wird jetzt der jeweilige SW aufgeführt. Ebenfalls erfolgt die Meldung in der Konsole.

5.7.4 Installation und Aktivierung: Füllen des Caches:

//überarbeiten

Hier wird dem Service-Worker (mit Hilfe von `this`) ein `installEvent`-Listener hinzugefügt, und die `ExtendableEvent.waitUntil()`-Methode dem Event angehängt — das stellt sicher, dass der Service-Worker nicht installiert wird, bis der Code innerhalb von `waitUntil()` erfolgreich ausgeführt wurde.

Innerhalb von `waitUntil()` benutzen wir die `caches.open()`-Methode um einen neuen Cache mit dem Namen `v1` zu erstellen, welcher die erste Version des Ressourcenspeichers unserer Seite sein wird. Das gibt eine Promise für den erstellten Cache zurück; sobald diese aufgelöst ist, rufen wir eine Funktion auf, die wiederum `addAll()` auf dem erstellten Cache aufruft und als Parameter einen Array mit zum Ursprung relativen URLs der zu cachenden Ressourcen enthält.

Sollte die Promise zurückgewiesen werden, schlägt die Installation fehl und der Worker tut nichts. Das ist in Ordnung und bietet die Möglichkeit, den Code zu korrigieren und es bei der nächsten Registrierung erneut zu versuchen.

Nach erfolgreicher Installation wird der Service-Worker aktiviert. Wurde der Service-Worker zum ersten mal installiert und aktiviert ist dies nicht weiter von Bedeutung, aber wichtig, wenn der Service-Worker aktiviert wird. (Siehe den Abschnitt [Aktualisieren des Service-Workers](#) im späteren Verlauf.)

```
this.addEventListener('install', function (event) {
  event.waitUntil(
    caches.open('v2').then(function (cache) {
      console.log("install successful");
      return cache.addAll([
        '/',
        '/pwa/index.html',
        '/pwa/css/style.css',
        '/pwa/js/pwa.js',
        '/pwa/js/jquery-3.3.1.min.js',
        '/pwa/img/by-songkick-pink.png',
        '/pwa/img/sk-badge-pink.png'
      ])
    })
    .catch(function (err) {
      console.log("this is an error ", err);
    })
  );
});
```

im SW ist nur asynchron erlaubt

erste Variante, muss noch angepasst werden, aber dieser Screenshot sollte vorhanden sein und nachfolgend obiger Text.

Bemerkung zu den Pfaden machen: wichtig, dass sie korrekt sind und dass in diesem Fall `/pwa/` vorne steht und kein Punkt. Berücksichtigen, dass es Unterschiede gibt ob localhost oder auf dem Server.

`event.respondWith()` Methode weist den Browser an, das Ergebnis des Ereignisses in der Zukunft zu bewerten. `Cache.match(Ereignis.request)` nimmt die aktuelle Webanforderung, die das Fetch-Ereignis ausgelöst hat, und sucht im Cache nach einer Ressource, die übereinstimmt. Das `match` wird durch einen Blick auf die URL-Zeichenfolge durchgeführt. Die `match`-Methode gibt ein Versprechen zurück, das auch dann aufgelöst wird, wenn die Datei nicht im Cache gefunden wird. Das bedeutet, dass Sie

eine Wahl darüber haben, was Sie tun. In Ihrem einfachen Fall, wenn die Datei nicht gefunden wird, möchten Sie Sie einfach aus dem Netzwerk Holen und an den Browser zurückgeben.

Dies ist der einfachste Fall, es gibt viele andere caching-Szenarien. Zum Beispiel könnten Sie inkrementell alle Antworten für zuvor nicht zwischengespeicherte Anfragen Zwischenspeichern, so dass Sie alle in Zukunft aus dem cache zurückgegeben werden.

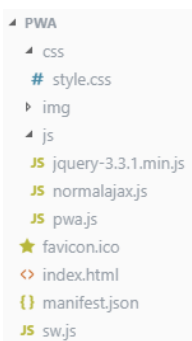
Um zuerst zu testen, ob er fetch funktioniert, kann man diese in der Konsole ausgeben und dann erhalten wir das Ergebnis:

```
console.log(event.request.url);
```

```
https://cyon.netlive.ch/pwa/css/style.css          SW, js:24
https://cyon.netlive.ch/pwa/img/by-songkick-pink.png SW, js:24
https://cyon.netlive.ch/pwa/js/jquery-3.3.1.min.js  SW, js:24
https://cyon.netlive.ch/pwa/js/pwa.js              SW, js:24
https://fonts.googleapis.com/css?family=Hind+Siliguri SW, js:24
https://fonts.gstatic.com/s/hindsiliguri/v5/ijwTs5juQtsylLR5jM4cx8FoTjzax8s3Jik.woff2  SW, js:24
SW registered! Scope is https://cyon.netlive.ch/pwa/  PWA, js:64
https://api.songkick.com/api/3.0/search/artists.json?apikey=AmrQIBJ7cwQMJaIm&query=angerfist SW, js:24
id: 144843                                          PWA, js:25
https://api.songkick.com/api/3.0/artists/144843/calendar.json?apikey=AmrQIBJ7cwQMJaIm  SW, js:24
https://cyon.netlive.ch/pwa/manifest.json          SW, js:24
result: ▶ (4) [{-}, {-}, {-}, {-}]                PWA, js:37
https://cyon.netlive.ch/pwa/img/sk-badge-pink.png  SW, js:24
```

```
this.addEventListener('fetch', function (event) {
  console.log(event.request.url);
  event.respondWith(
    caches.match(event.request).then(function (response) {
      return response || fetch(event.request);
    })
  );
});
```

5.7.5 Hierarchie der Files(spielt eine grosse Rolle und wird schlecht beschrieben)



Warum Icon nicht? Iphone special URL

```
<link rel="apple-touch-icon" sizes="512x512" href="img/sk-badge-pink.png">
```

Aktueller Stand bei der Überprüfung mit dem Audits:



Progressive Web App

These checks validate the aspects of a Progressive Web App, as specified by the baseline [PWA Checklist](#).

- 1 Does not redirect HTTP traffic to HTTPS

If you've already set up HTTPS, make sure that you redirect all HTTP traffic to HTTPS. [Learn more](#).

Additional items to manually check

3 audits

Passed audits

11 audits

5.8 Optimierung

5.8.1 Fehlgeschlagene Cache-Anfragen abfangen

//überarbeiten

Schreibe über die Möglichkeiten, den SW zu optimieren. Aber in meinem Fall ist das nicht nötig da ich keine grossen Ressourcen bereitstellen muss.

```
this.addEventListener('fetch', function (event) {
  event.respondWith(
    caches.match(event.request).then(function (response) {
      return response || fetch(event.request).then(function(respo){
        caches.open('pwacache').then(function(cache){
          cache.put(event.request, respo.clone());
        });
        return response;
      });
    }).catch(function(){
      console.log("You need an Internetconnection");
    })
  );
});
```

<https://habr.com/en/company/2gis/blog/345552/>

5.9 Optimierung der Usability

Für die Benutzerfreundlichkeit gibt es einiges zu Beachten. Man soll nie davon ausgehen, dass der User geübt ist oder alles auf Anhieb versteht. Deswegen sollte man die Applikation simpel und übersichtlich halten. Kein unnötiger Text und einfach beschrieben.

find all Events of an Artist

for example

[Angerfist](#)

[Martin Garrix](#)

[Rammstein](#)

last search

[Angerfist](#)

[Martin Garrix](#)

[Rammstein](#)

5.9.1 Push Notifications

Die Push Notifications sind je nach Einsatzgebiet ziemlich aufwändig zu Programmieren. Aber ich habe mich jetzt für die einfachste Variante entschieden, das bedeutet, ich frage den User ob ich Push Nachrichten senden kann und zeige dann bei jedem Reload der Seite eine Nachricht mit einer Action. Er kann die Nachricht schliessen oder direkt auf die Website.

Pwa.js:

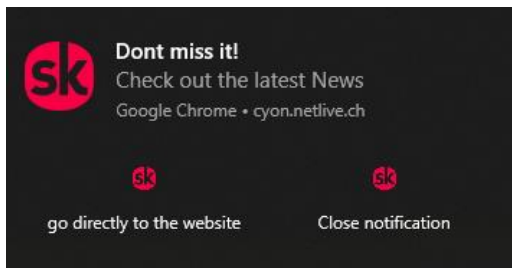
```
if ('Notification' in window && navigator.serviceWorker) {
  Notification.requestPermission(function(status){
    console.log("Notification permission status:", status);
  })
}

function displayNotification() {
  if (Notification.permission == 'granted') {
    navigator.serviceWorker.getRegistration().then(function(reg) {
      var options = {
        body: 'Check out the latest News',
        icon: 'img/sk-badge-pink.png',
        vibrate: [100, 50, 100],
        data: {
          dateOfArrival: Date.now(),
          primaryKey: 1
        },
        actions: [
          {action: 'explore', title: 'go directly to the website',
            icon: 'img/sk-badge-pink.png'},
          {action: 'close', title: 'Close notification',
            icon: 'img/sk-badge-pink.png'},
        ]
      };
      reg.showNotification('Dont miss it!', options);
    });
  }
}
```

Sw.js:

```
//push notification
this.addEventListener('notificationclick', function(e) {
  var notification = e.notification;
  var action = e.action;

  if (action === 'close') {
    notification.close();
    console.log("push close");
  } else {
    clients.openWindow('https://www.songkick.com');
    console.log("go to website");
    notification.close();
  }
});
```



5.10 Weitere Funktionen einbauen für bessere Usability

5.11 Design

Hinzufügen von «zuletzt gesucht» beschreiben, etc

Search for upcoming Events

miss k8

Upcoming Events of miss k8

REBIRTH Festival · Guardian of Eternity 2019

12.4.2019

Haaren, Netherlands

Type: Festival

[more infos](#)

Festival Panoramas 2019

12.4.2019

Morlaix, France

Type: Festival

[more infos](#)

songkick

6 Testing

7 Fazit / Auswertung / Zielerreichung / Zielüberprüfung

8 Selbständigkeitserklärung

9 Glossar / Verzeichnisse

API

Manifest

Service Workers

10 Anhang