Griechisch App

Learnings

von

Kapitel 1

Progressive Web App

1.0.1 •

Inhaltsverzeichnis

Tabellenverzeichnis

Abbildungsverzeichnis

Anhang A

Code snippets

1. manifest:

```
{
  ”name”: ”Weather”,
  ”short_name”: ”Weather”,
  ”icons”: [{
    ”src”: ”/images/icons/icon−128x128.png”,
      ”sizes”: ”128x128”,
      ”type”: ”image/png”
    }, {
      ”src”: ”/images/icons/icon−144x144.png”,
      ”sizes”: ”144x144”,
      ”type”: ”image/png”
    }, {
      ”src”: ”/images/icons/icon−152x152.png”,
      ”sizes”: ”152x152”,
      ”type”: ”image/png”
    }, {
      ”src”: ”/images/icons/icon−192x192.png”,
      ”sizes”: ”192x192”,
      ”type”: ”image/png”
    }, {
      ”src”: ”/images/icons/icon−256x256.png”,
      ”sizes”: ”256x256”,
      ”type”: ”image/png”
    }, {
      ”src”: ”/images/icons/icon−512x512.png”,
      ”sizes”: ”512x512”,
      ”type”: ”image/png”
    }],
  ”start_url”: ”/index.html”,
  ”display”: ”standalone”,
  ”background_color”: ”#3E4EB8”,
  ”theme_color”: ”#2F3BA2”
}

\\ verlinkung mit dem index.html
<link rel=”manifest” href=”/manifest.json”>
```

2. ios meta tags und icons

```
<meta name="apple-mobile-web-app-capable" content="yes">
<meta name="apple-mobile-web-app-status-bar-style" content="black">
<meta name="apple-mobile-web-app-title" content="Weather␣PWA">
<link rel="apple-touch-icon" href="/images/icons/icon-152x152.png">
```

3. description für SEO

```
<meta name="description" content="A␣sample␣weather␣app">
```

4. theme-color

```
<meta name="theme-color" content="#2F3BA2" />
```

5. Registrierung des Service Workers

```
if ('serviceWorker' in navigator) {
  window.addEventListener('load', () => {
    navigator.serviceWorker.register('/service-worker.js')
        .then((reg) => {
          console.log('Service␣worker␣registered.', reg);
        });
  });
}
```

6. Array für die Sachen die es zu cachen gilt

```
const FILES_TO_CACHE = [
  '/offline.html',
];
```

7. Precache the offline page beim install event

```
evt.waitUntil(
     caches.open(CACHE_NAME).then((cache) => {
        console.log('[ServiceWorker]␣Pre-caching␣offline␣page');
        return cache.addAll(FILES_TO_CACHE);
     })
);
```

8. lösche alte cache daten beim activate event

```
evt.waitUntil(
     caches.open(CACHE_NAME).then((cache) => {
        console.log('[ServiceWorker]␣Pre-caching␣offline␣page');
        return cache.addAll(FILES_TO_CACHE);
     })
);
```

9. fetch event handler

```
if (evt.request.mode !== 'navigate') {
  // Not a page navigation, bail.
  return;
}
evt.respondWith(
    fetch(evt.request)
        .catch(() => {
          return caches.open(CACHE_NAME)
              .then((cache) => {
                return cache.match('offline.html');
              });
        })
);
```

10. prüft ob das object im cache liegt und bereit ist um geladen zu werden

```
if (!('caches' in window)) {
  return null;
}
const url = `${window.location.origin}/forecast/${coords}`;
return caches.match(url)
    .then((response) => {
      if (response) {
        return response.json();
      }
      return null;
    })
    .catch((err) => {
      console.error('Error␣getting␣data␣from␣cache', err);
      return null;
    });
```

11. call getForecast

```
getForecastFromCache(location.geo)
    .then((forecast) => {
      renderForecast(card, forecast);
    });
```

12. app.js

```
// If the data on the element is newer, skip the update.
if (lastUpdated >= data.currently.time) {
  return;
}
```

13. service-worker.js Konstanten für den cache

```
const CACHE_NAME = 'static-cache-v2';
const DATA_CACHE_NAME = 'data-cache-v1';
```

14. service-woreker.js Daten die gecached werden sollen

```
const FILES_TO_CACHE = [
  '/',
  '/index.html',
  '/scripts/app.js',
  '/scripts/install.js',
  '/scripts/luxon-1.11.4.js',
  '/styles/inline.css',
  '/images/add.svg',
  '/images/clear-day.svg',
  '/images/clear-night.svg',
  '/images/cloudy.svg',
  '/images/fog.svg',
  '/images/hail.svg',
  '/images/install.svg',
  '/images/partly-cloudy-day.svg',
  '/images/partly-cloudy-night.svg',
  '/images/rain.svg',
  '/images/refresh.svg',
  '/images/sleet.svg',
  '/images/snow.svg',
  '/images/thunderstorm.svg',
  '/images/tornado.svg',
  '/images/wind.svg',
];
```

15. service-worker.js update fetch handler

```
if (evt.request.url.includes('/forecast/')) {
  console.log('[Service␣Worker]␣Fetch␣(data)', evt.request.url);
  evt.respondWith(
      caches.open(DATA_CACHE_NAME).then((cache) => {
        return fetch(evt.request)
            .then((response) => {
              // If the response was good, clone it and store it in
                  the cache.
              if (response.status === 200) {
                cache.put(evt.request.url, response.clone());
              }
              return response;
            }).catch((err) => {
              // Network request failed, try to get it from the cache
                  .
              return cache.match(evt.request);
            });
      }));
  return;
}
evt.respondWith(
    caches.open(CACHE_NAME).then((cache) => {
      return cache.match(evt.request)
          .then((response) => {
            return response || fetch(evt.request);
          });
    })
);
```

16. install script verlinkung

```
<script src="/scripts/install.js"></script>
```

17. beforeinstallprompt event

```
window.addEventListener(’beforeinstallprompt’,
    saveBeforeInstallPromptEvent);
```

18. installation fehlt noch