

Cache mit Service Worker

Verzeichnisstruktur

```
|- index.html
|- service-worker.js
|- css /
|   |- app.css
|- js /
|   |- app.js
```

Service Worker registrieren in app.js

```
if('serviceWorker' in navigator){
  navigator.serviceWorker
    .register('./service-worker.js')
    .then(function() {
      console.log('Service Worker wurde registriert');
    });
}else{
  console.log('Browser bietet keine Unterstützung für Service Worker');
}
```

service-worker.js

```
// Daten
let cacheName = 'NameDesProjektes';
let filesToCache = [
  './',
  './index.html',
  './css/app.css',
  './js/app.js'
];

// Dateien in Cache speichern
self.addEventListener('install', function(event) {
  console.log('[Service Worker] Installation wird gestartet');
  event.waitUntil(
    caches.open(cacheName).then(function(cache) {
      console.log('[Service Worker] Dateien werden gecached');
      return cache.addAll(filesToCache);
    })
  );
});

// Dateien vom Cache laden
self.addEventListener('fetch', function(event) {
  console.log('[Service Worker] Angeforderte URL: ', event.request.url);
  event.respondWith(
    caches.match(event.request).then(function(response){
      if(response){
        console.log('[Service Worker] Ressource aus dem Cache geladen: ', event.request.url);
        return response;
      }else{
        console.log('[Service Worker] Ressource aus dem Netzwerk geladen: ', event.request.url);
        return fetch(event.request);
      }
    })
  );
});
```