

Laboratorium 2

Progresywna Web Aplikacja (PWA) – tryb offline

Celem laboratorium jest utworzenie strony internetowej umożliwiającej działanie bez dostępu do Internetu.

Aplikacje internetowe mogą zapewniać działanie w trybie offline przy użyciu dwóch technik. Starsza implementacja, Application Cache, jest szeroko zaimplementowana w przeglądarkach, ale obecnie jest wycofywana z powodu różnych wad koncepcyjnych i projektowych.

Nowoczesna alternatywa nazywa się **Cache API** i jest dostępna w ramach Service Worker – oddzielnej jednostki kodu, poprzez którą aplikacje internetowe działające na HTTPS mogą zażądać zainstalowania przez przeglądarkę internetową. Jednostka ta jest następnie uruchamiana w oderwaniu od aplikacji internetowej będącej jej właścicielem, komunikując się z nią za pośrednictwem zdarzeń. Service Worker jest podstawowym budulcem idei Progressive Web Apps (PWA). Oprócz tego, że umożliwia korzystanie z wielu złożonych interfejsów API, takich jak powiadomienia push, synchronizacja w tle lub geofencing, może działać jako w pełni funkcjonalny sieciowy serwer proxy. Może przechwytywać wszystkie żądania HTTP, zmieniać ich zawartość lub zachowanie, a przede wszystkim zarządzać buforowaniem offline.

Treść dodawana do Cache API może być dodatkowo indeksowana i udostępniana przeglądarce za pomocą Content Indexing API. Od wiosny 2020 r. jest to wczesna propozycja Google Chrome, dostępna tylko w tej przeglądarce na Androidzie w ramach eksperymentów Origin Trial. Wpisy dodawane do indeksu powinny być buforowane w Cache API i obsługiwane w trybie offline przez Service Worker. Przeglądarka może następnie przedstawić użytkownikowi zindeksowane wpisy w trybie offline.

Zadania do wykonania:

1. Zmodyfikowanie strony internetowej zrealizowanej na laboratorium 1 poprzez dodanie skryptu wyświetlającego alert na stronie po jej otwarciu.
2. Zmodyfikowanie pliku /service-worker.js

```
function isSuccessful(response) {
  return response &&
    response.status === 200 &&
    response.type === 'basic';
}

self.addEventListener('fetch', function (event) {
  event.respondWith(
    caches.match(event.request)
      .then(function (response) {
        if (response) {
          return response; // Cache hit
        }
      })

    return fetch(event.request.clone())
  )
})
```

```
.then(function (response) {
  if (!isSuccessful(response)) {
    return response;
  }

  caches.open(CACHE_NAME)
    .then(function (cache) {
      cache.put(event.request, response.clone());
    });

  return response;
})
);
});
```

3. Poinformowanie użytkownika aplikacji o statusie połączenia sieci Internet
4. Umieszczenie plików aplikacji na serwerze www
 - a. Dane serwera www
 - i. Host: <ftp.peckos.pl>
 - ii. Login: lab@peckos.pl
 - iii. Hasło: Lab123!@#.
5. Sprawdzenie aplikacji