The Future of Web - Progressive Web App

목차

1. PWA 소개
2. PWA 배경
3. PWA 특징
4. PWA 사례
    - Ali Express
    - The Washington Post
    - Solution Portal
5. PWA 기술
    - Wep App Manifest (Icon, Banner)
    - Service Worker (Cache, Web Push)
6. 참고 자료

프로필

- Open Source Contributor
  - Google Web Fundamentals (The Fundamental Knowledge for Web)
  - Google HTML5 Rocks (Top Notch Resources for Web Developer)
- Google Developer Group - Web Tech 커뮤니티 리더

PWA 소개

> "Progressive Web Apps are experiences that combine the best of the web and the best of apps" 최고의 모바일 앱과 최고의 웹 앱을 결합한 경험

- Progressive Web Apps 는 무엇인가?
  - 진보한 웹 앱
  - 최신 웹 기술들로 웹 앱에서도 모바일 앱과 같은 사용자 경험을 느낄 수 있는 웹앱
  - 빌드 X, 배포 X, 설치 O, 오프라인 O, 알람 O ...
  - **최적화된 웹 성능에 모바일 Native 기능을 결합한 최신 웹 앱**

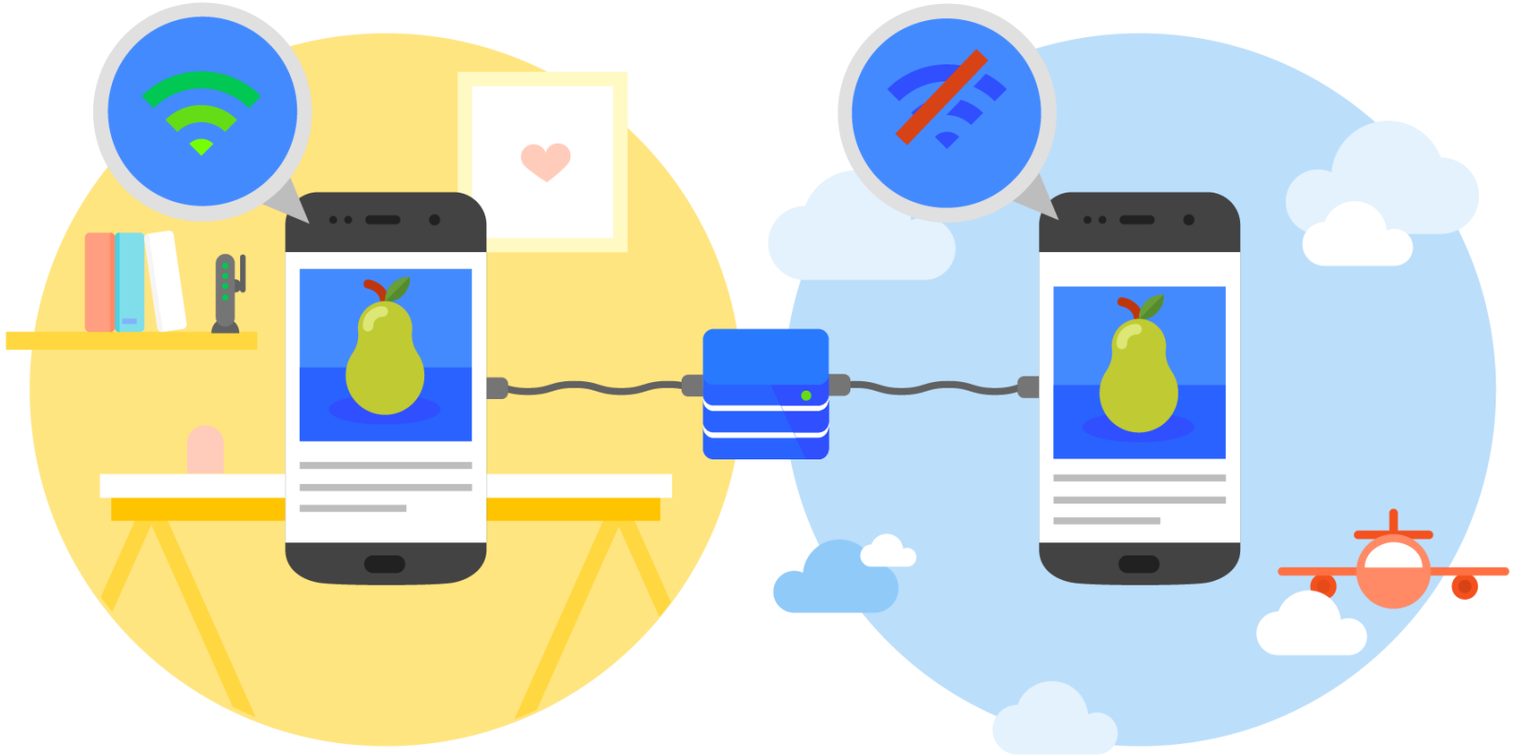
PWA 등장 배경

- 모바일 시장의 양분화 : Android, iOS 중심의 모바일 생태계 / 각 OS 특화된 개발 필요
- Hybrid App 이라는 기존의 대안 : Native UI 의 성능을 따라가기가 어려움
- Offline Web App 의 필요성 대두 : 느린 인터넷의 보급으로 모바일 단말기에서 웹 앱의 사용성이 떨어짐

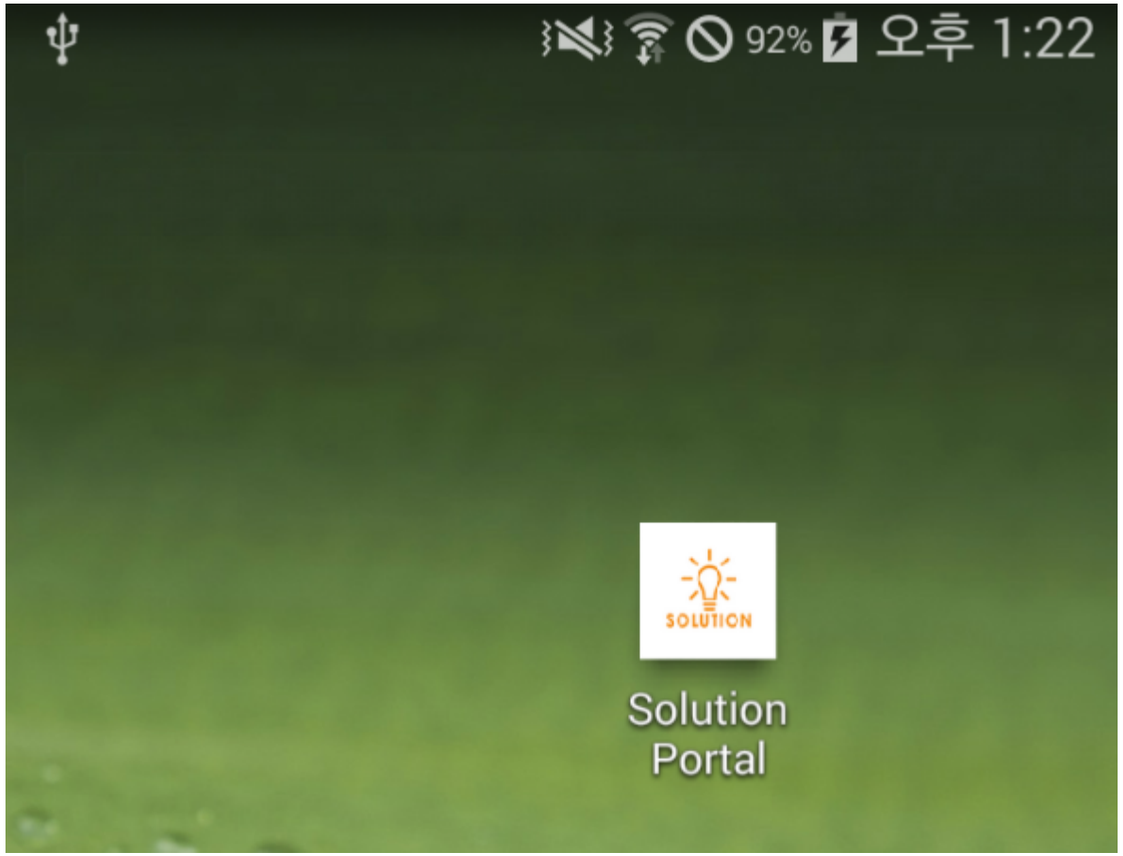
PWA 기술 특징

- **Responsive : PC, Mobile, Tablet 에 관계없이 해당 기기에 최적화된 UI**

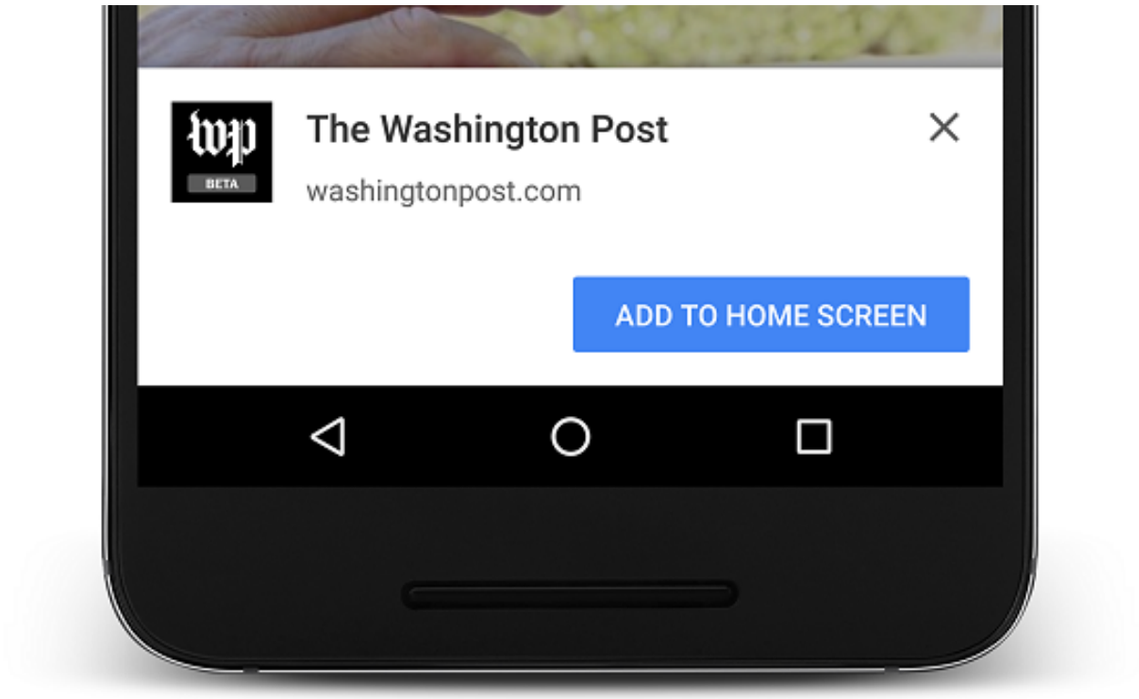
- **Connectivity : 저속의 네트워크 환경, 오프라인 환경에서도 사용 가능**



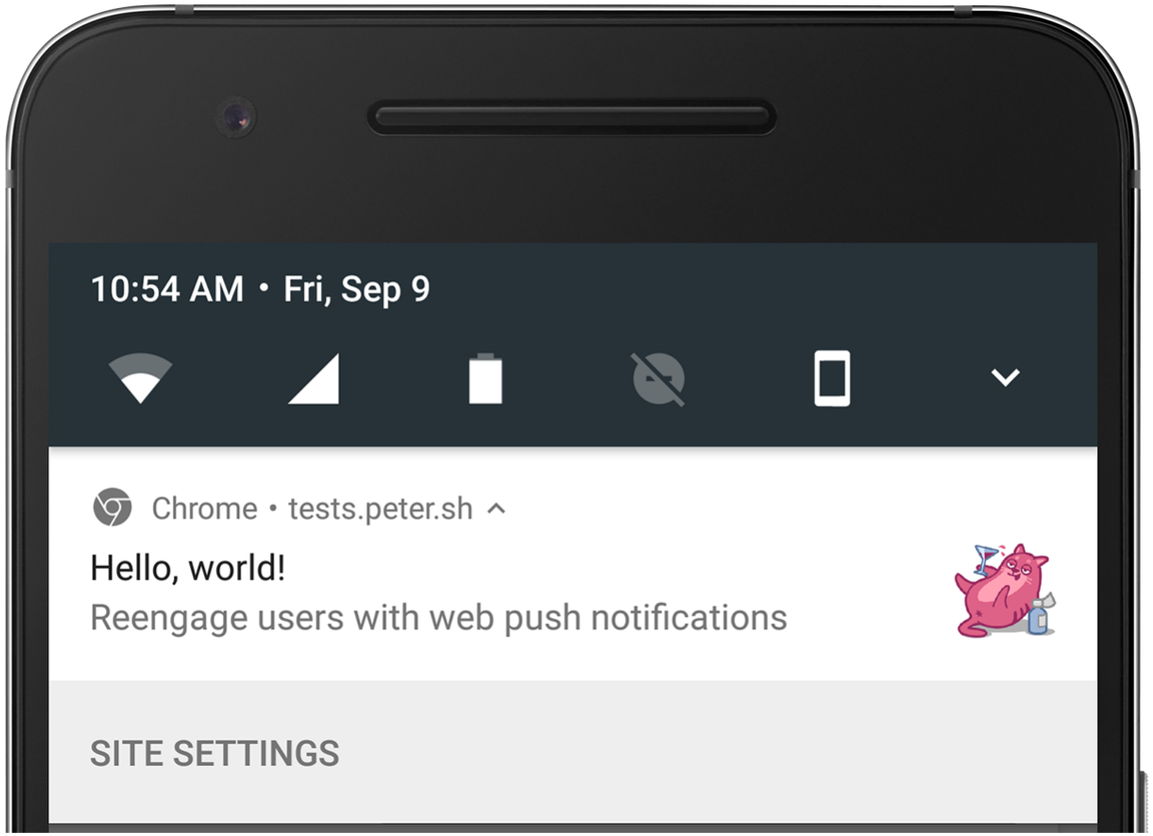
- **App-like : Mobile App 과 동일한 실행방식 (icon) 과 사용자 인터랙션 (UX) 을 제공**



- **Discoverable : URL 로 사이트 접근 후 원클릭 설치 가능 (banner)**



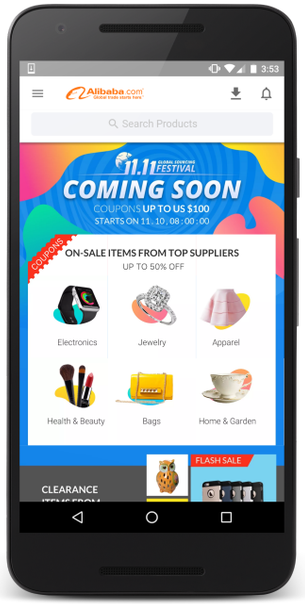
- **Engageable : Push 알람으로 사용자의 재방문 유도가 용이**



- **Safe : HTTPS 통신으로 기밀 정보의 유지가 가능**

외부 적용 사례

- Alibaba
  - 전세계 최대 규모의 B2B 전자상거래 시스템
  - Add-Home Screen 기능으로 모바일 Active Users 44% 증가
  - Case Study 자료

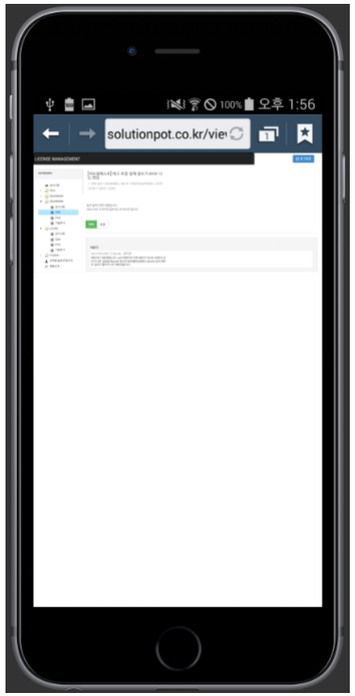


- The Washington Post
  - 미국 메이저 신문사
  - AMP로 사이트 재방문 사용자수 23% 증가
  - Case Study 자료

내부 적용 사례

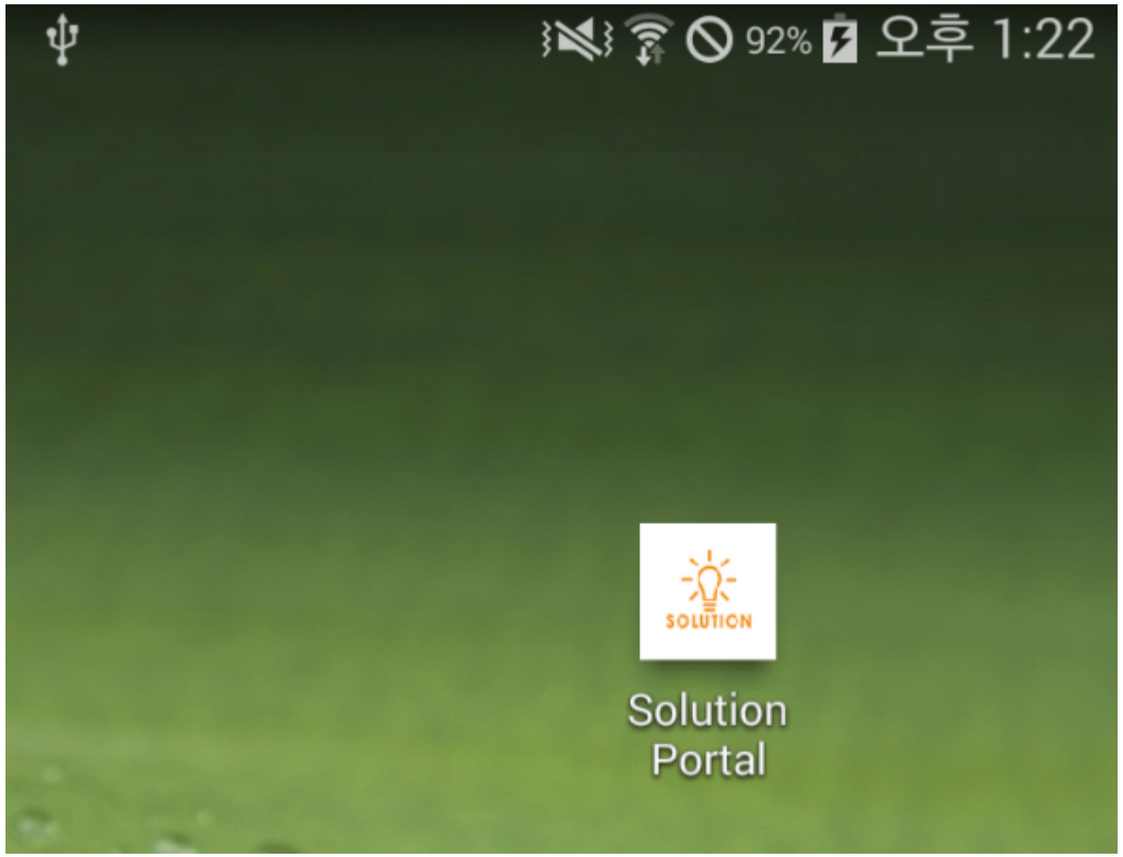
- SW 기반솔루션 포탈 (Solution Portal)
  - 포스코 ICT 솔루션 (Glue, Glue Mobile, Posbee) 기술질의 게시판
  - Web Push 알람을 이용한 기술질의 응답속도 3배 증가

- 기존 Solution Portal 의 문제점
  - Mobile 기기 미지원 : 게시글 내용 확인이 어렵고, Mobile 솔루션 홍보에 부적절
  - 실시간 게시글 확인 미지원 : 고객의 게시글 여부를 Manual 하게 일일이 확인해야함

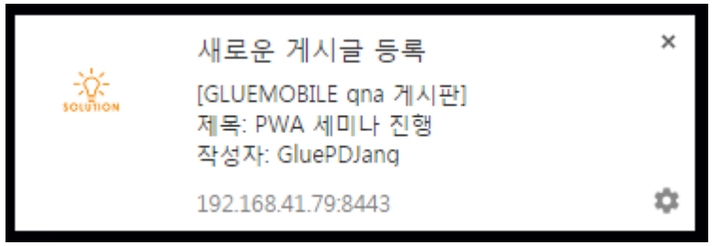


- PWA 를 적용한 Solution Portal
  - Responsive : 반응형 웹 디자인을 적용한 Mobile 기기 지원

- PWA 를 적용한 Solution Portal
  - App-like : 모바일 Icon 을 이용한 사이트 접속 및 모바일 UX 제공



- PWA 를 적용한 Solution Portal
  - Engageable : Web & Mobile Push 알람을 이용한 **실시간** 게시글 알림



PWA 주요 기술

Wep App Manifest (Icon & Install Banner)

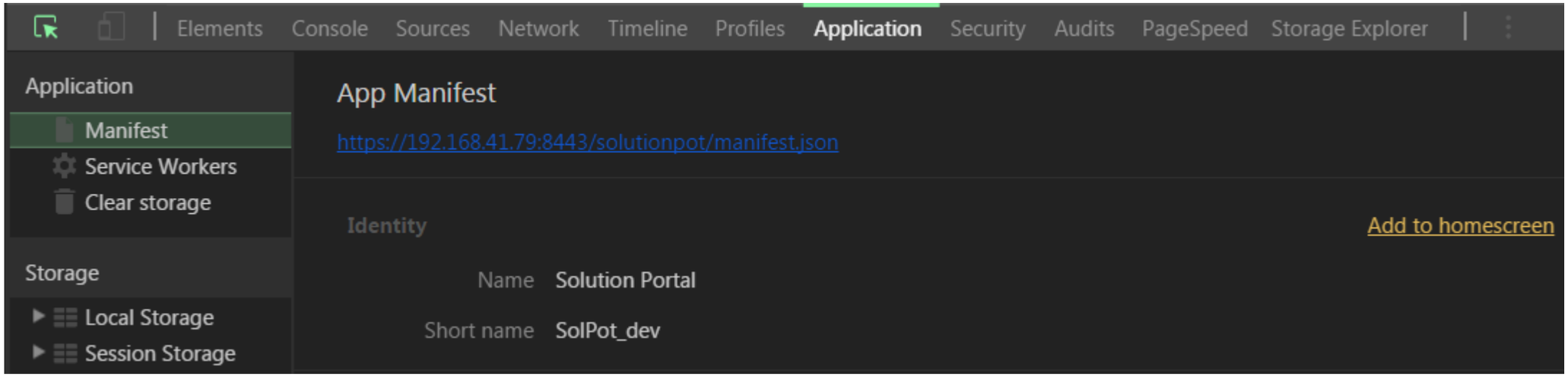
- 앱 아이콘, 화면 런쳐 방식 및 배경색, 시작 페이지 등을 설정할 수 있는 JSON 파일

```
{
  "short_name": "SolPot_dev",
  "name": "Solution Portal",
  "icons": [
    {
      "src": "dist/images/icons/icon-32x32.png",
      "type": "image/png",
      "sizes": "32x32"
    },
    ...
  ],
  "background_color": "#1E88E5",
  "display": "standalone",
  "start_url": "./"
}
```

- Web App Manifest 파일에서 지원하는 기능들
  - 홈 화면에 Web App Icon 추가
  - 사이트 방문시 하단에 설치 배너 표시 (5분 간격 2번 이상 방문시 표시됨)
  - 런쳐화면 방향, 크기, 배경색 설정

- 인스톨 배너를 이용한 앱 설치 및 실행
  - 데모
- 홈 화면 Icon 추가 시연
  - 안드로이드 폰

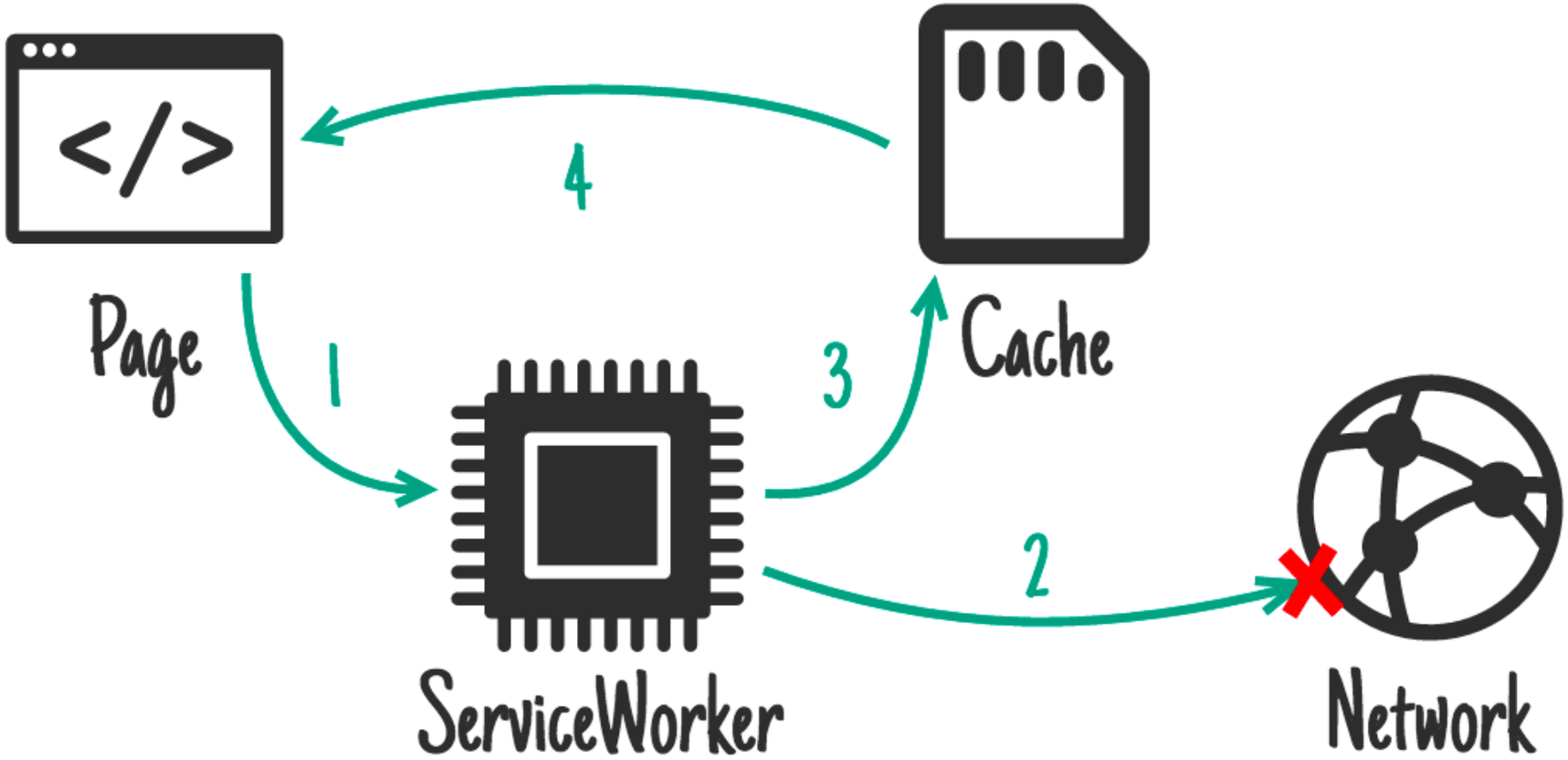
- 크롬 개발자 콘솔의 Application 탭에서 디버깅 및 조작 가능



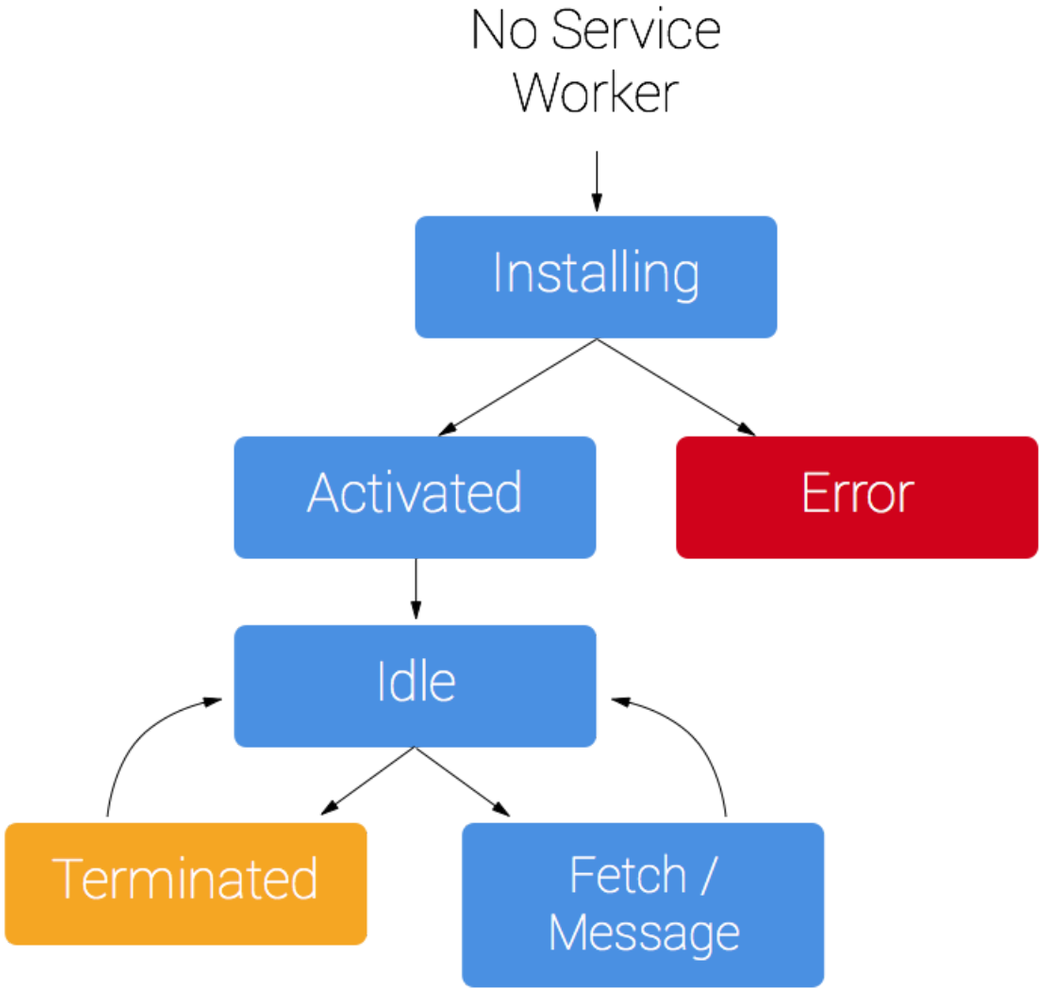
Service Worker (Offline, Cache, Web Push)

- 오프라인 서비스, 푸쉬 알람 등의 모바일 기능을 웹에서 가능하게 하는 코어 기술
- 브라우저의 백그라운드에서 돌아가는 스크립트, 브라우저와 네트워크의 미들웨어
  - ex) 네이티브 모바일 앱과 비교하였을 때, 브라우저의 화면 쓰레드 이외에 브라우저 뒷단에서 돌아가는 쓰레드가 하나 더 있다고 생각

- 웹 페이지에서 네트워크 요청 발생시 해당 요청을 가로챔
- 캐쉬가 있을 경우 즉시 로딩하여 속도가 매우 빠름



- 웹 페이지와는 별개의 라이프 싸이클을 가진다.



Service Worker 실습

- 서비스 워커 등록 : `navigator.serviceWorker.register()`

```
if ('serviceWorker' in navigator) {
  navigator.serviceWorker.register('/sw.js').then(function(registration) {
    // 등록 성공
    console.log('ServiceWorker registration successful with scope: ', registration.scope
  }).catch(function(err) {
    // 등록 실패
    console.log('ServiceWorker registration failed: ', err);
  });
}
```

- 서비스 워커 설치 : `self.addEventListener('install', function(event) {});`

```
var CACHE_NAME = "my-first-cache";
var cacheFiles = [
  "/",
  "/pwa.png"
];

self.addEventListener('install', function(event) {
  event.waitUntil(
    caches.open(CACHE_NAME)
      .then(function(cache) {
        console.log('Opened cache');
        return cache.addAll(cacheFiles);
      })
  );
  console.log("서비스 워커 설치 완료");
});
```

- 서비스 워커 디버깅
  - Chrome Developer Tools 의 Application 탭
  - 실행중인 서비스 워커 확인은 `chrome://inspect/#service-workers`

- 서비스 워커를 이용한 Push 알람 구현
  - 솔루션 포탈
- Facebook 서비스 워커 소스 디버깅 실습
  - 페이스북 링크

- 서비스 워커 개발 환경
  - HTTPS 통신 + 최신 모던 브라우저

- 지원 브라우저
  - Google Chrome
  - Mozilla Firefox
  - Microsoft Edge
  - Opera
  - Samsung Mobile Browser
  - Safari (지원 예정)

마무리

- 아직은 디버깅이 어려운 서비스 워커
- 기 개발된 서비스에 적용하기에는 제약사항이 존재
- HTTP 와 HTTPS 의 어마어마한 차이
- 일부 모바일 앱을 대체할 수 있는 사용자 경험
- 현재 보다는 앞으로가 더 기대되는 PWA

참고 자료

- Google Web Fundamentals
- Google HTML5 Rocks
- Service Worker 지원 브라우저
- 지디넷 - 오프라인 웹이 온다

감사합니다.