

Web App Manifest - Install Banner & App Icon

개요

- PWA 의 주요 기술 중 하나인 Web App Manifest 파일의 세부 속성을 학습
- 모바일 앱 설치의 편의성을 가진 Install banner 의 장점과 동작조건을 학습
- Web App Manifest 를 가지고 개발할 때 주의할점 학습 (Navigation Scope, Deep link)

목차

- Web App Manifest 소개
- Web App Manifest 파일 구조 & 등록
- Web App Manifest 주요 구성 정보
 - App Icon
 - Launch Image
 - Start URL
 - Display Type
 - Display Orientation
- Web App Install Banner
- Web App Manifest Navigation Scope
- Web App Manifest 디버깅
- Web App Manifest 실습

Web App Manifest 란?

- Progressive Web App 의 설치와 앱 구성정보를 담고 있는 json 형식의 설정 파일
 - 앱 아이콘, 화면 런처 방식 및 배경색, 시작 페이지 등을 설정할 수 있는 JSON 파일

- 앱 관련 구성정보에는 아래와 같은 항목들이 설정됩니다.
 - Start URL : 웹 앱이 시작되는 지점
 - Launch Image : 웹 앱 시작 화면
 - Display Type : 웹 앱의 화면 형태
 - Display Orientation : 웹 앱 화면 방향
 - App Icon : 앱 아이콘 이미지와 크기

주의! 완전 새로운게 아니라 기존에 있던 겁니다.

```
<!-- iOS -->
<meta name="apple-mobile-web-app-capable" content="yes">
<meta name="apple-mobile-web-app-status-bar-style" content="black">
<link rel="apple-touch-icon" sizes="72x72" href="/apple-touch-icon-72x72.png">
<link rel="apple-touch-startup-image" href="/splash-startup.png">
```

```
<!-- Android & Chrome -->
<meta name="mobile-web-app-capable" content="yes">
<link rel="icon" sizes="72x72" href="/icon72.png">
<link rel="icon" sizes="57x57" href="/icon57.png">
```

```
<!-- IE 10 -->  
<meta name="application-name" content="my app"/>  
<meta name="msapplication-TileColor" content="#000000"/>  
<meta name="msapplication-square150x150logo" content="square.png"/>  
<meta name="msapplication-wide310x150logo" content="wide.png"/>
```

그리하여 Web App Manifest 종합세트가..

Web App Manifest 파일 구조

```
{
  "short_name": "앱 아이콘 이름",
  "name": "하단 설치 배너에 표기될 이름 & 앱에서 검색시 키워드",
  "icons": [
    {
      "src": "dist/images/icons/icon-32x32.png",
      "type": "image/png",
      "sizes": "32x32"
    },
    {}
  ],
  "background_color": "#1E88E5",
  "display": "standalone",
  "start_url": "./"
}
```


Web App Manifest 파일 등록

- 메인 html 파일의 head 에 아래와 같은 meta 태그 추가

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

Wev App Manifest 주요 구성 정보

1) App Icon

- 해당 웹 사이트가 모바일 화면에서 표시될 아이콘 이미지 지정



```
{
  "icons": [{
    "src": "images/touch/icon-128x128.png",
    "type": "image/png",
    "sizes": "128x128"
  }, {
    "src": "images/touch/apple-touch-icon.png",
    "type": "image/png",
    "sizes": "152x152"
  }, {
    "src": "images/touch/chrome-touch-icon-192x192.png",
    "type": "image/png",
    "sizes": "192x192"
  }],
}
```

- `src` : 로딩할 이미지 파일 경로
- `type` : 로딩할 이미지 타입
- `sizes` : 로딩할 이미지 크기

주의사항

- app icon 미지정시 html 파일의 `<link rel="icon">` 태그를 검색한다.
- Safari 의 경우 아래의 meta 태그를 head 에 별도로 추가해주어야 한다.

```
<link rel="apple-touch-icon" href="touch-icon-iphone.png">  
<link rel="apple-touch-icon" sizes="152x152" href="touch-icon-ipad.png">  
<link rel="apple-touch-icon" sizes="180x180" href="touch-icon-iphone-retina.png">  
<link rel="apple-touch-icon" sizes="167x167" href="touch-icon-ipad-retina.png">
```

2) Launch Image - Splash Screen

- 웹앱이 시작될 때 거치는 시작 화면을 설정 가능
- **모바일 앱의 시작과 동일한 느낌을 가져감**
- 화면의 조합 : 아이콘 + 배경색 + 아이콘 이름



- 배경색 설정은 `background_color` 속성 이용

```
"background_color": "#2196F3",
```

- 아이콘은 icon 에 지정한 이미지 중 128dp = 192px 에 가장 가까운 크기로 지정
- 따라서, 192px 크기의 이미지는 꼭 지정

dp : 다양한 모바일 화면 크기에서 동일한 비율로 출력되게 하는 픽셀 단위

3) Start URL

- 앱이 시작될 때 로딩될 페이지 위치 지정

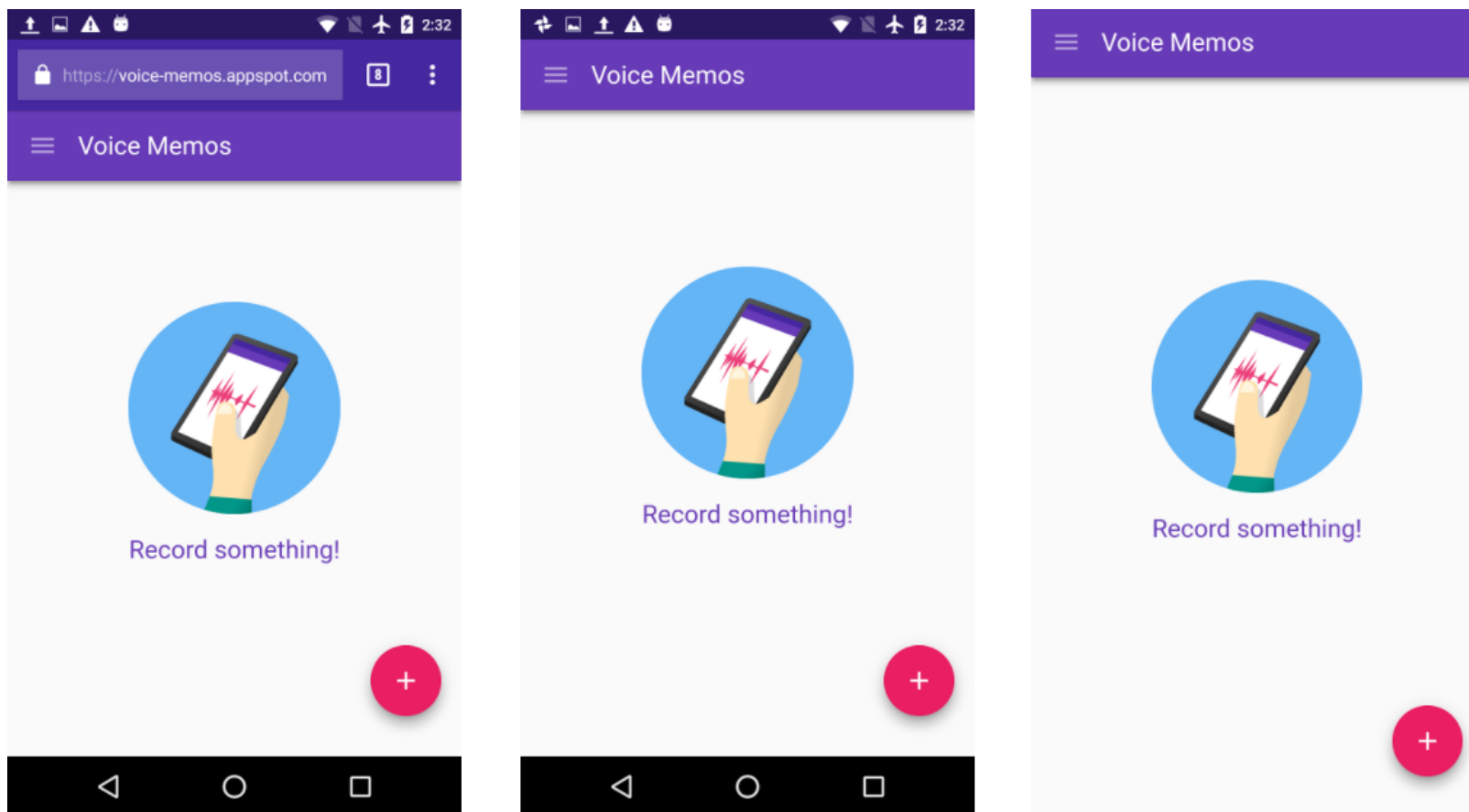
```
"start_url": "./"
```

- GA 분석이나 기타 목적으로 query string 을 뒤에 붙일 수 있다.

```
"start_url": "index.html?launcher=homescreen"
```

4) Display Type

- 웹앱 화면의 전체적인 모양을 정할 수 있다.
- 웹앱이 모바일 앱의 느낌을 가져갈 수 있도록 결정짓는 속성




```
"display": "standalone"
```

- `display` 속성의 옵션 값은 아래와 같다.
 - `standalone` : 상단 URL 바 제거하여 네이티브 앱 느낌 제공
 - `browser` : 해당 OS 브라우저에서 웹앱 실행
 - `fullscreen` : 크롬이 아닌 기타 브라우저에서 네이티브 앱 느낌 제공
 - `minimal-ui` : `fullscreen` 과 비슷하나 네비게이션 관련 최소 UI 를 제공

주의사항 : iOS 에서 standalone 사용시

- `<a>` 를 이용한 네비게이션 이동 시 새 브라우저 열기로 인해 context 을 잃게 됨.
- 따라서, `location.href` 또는 SPA 를 이용한 네비게이팅으로 전체 UX 를 가져갈 필요가 있음

```
<meta name="apple-mobile-web-app-capable" content="yes">
```

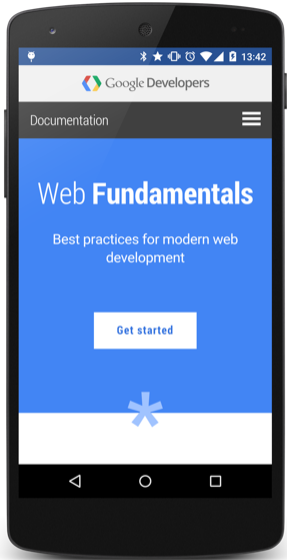
Theme Color

- `theme-color` 를 이용하여 앱 테마 색상을 정의할 수 있다.
- 홈 화면에서 시작해야 설정한 도메인의 모든 페이지에 적용됨

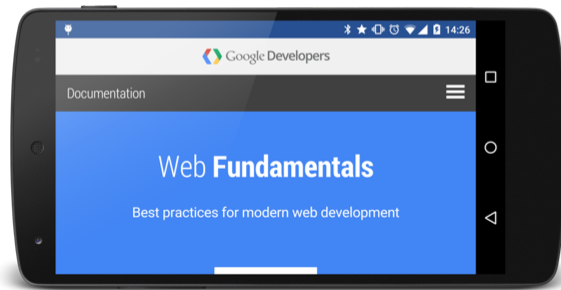
```
"theme_color": "#2196F3"
```

5) Display Orientation

- 화면 방향은 `orientation` 속성을 이용하고 옵션 값은 아래와 같다.
 - `portrait` : 세로 방향
 - `landscape` : 가로 방향



"orientation": "portrait"

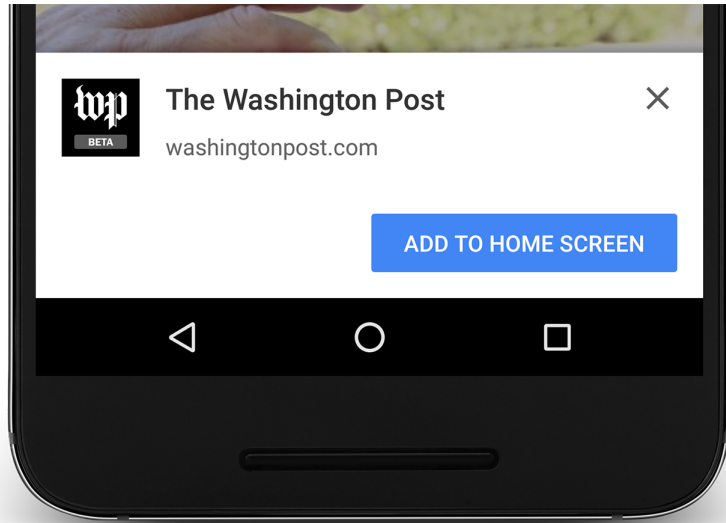


"orientation": "landscape"

"orientation": "landscape"

Web App Install Banner

- PWA 가 모바일적인 특징을 가지는 큰 부분의 하나
- 기존 모바일 앱 개발주기 : 구현 -> SDK 빌드 -> 스토어 배포 -> 검색 -> 앱 다운로드 -> 설치 -> 사용
- PWA 의 앱 개발주기 : 구현 -> 사이트 배포 -> 검색 -> 사용 (자동설치)



Install Banner 동작 조건

App Manifest 파일을 설정 후 아래 조건 만족시 동일 웹사이트에 대해 설치 배너 표시

- 최소 2 번 접속 (5 분 간격)
- `start_url` , `short_name` , `name` 설정
- 앱 아이콘 144 x 144 px 이미지 (PNG)
- Service Worker 등록
- HTTPS

beforeinstallprompt 로 설치 배너의 표시 시기를 지연하거나 disable 가능

```
window.addEventListener('beforeinstallprompt', function(e) {  
  console.log('beforeinstallprompt Event fired');  
  e.preventDefault();  
  
  //  
});
```

Install Banner 디버깅

- 주소창에 `chrome://flags` 입력
- 설정 옵션 중 **사용자 참여 검사 우회** 체크하여 조건 충족

Web App Manifest Navigation Scope

- "같은 도메인 아래의 웹 페이지들을 같은 scope 에 있다" 한다.
 - scope : 일반적인 네이티브 앱의 context 와 동일

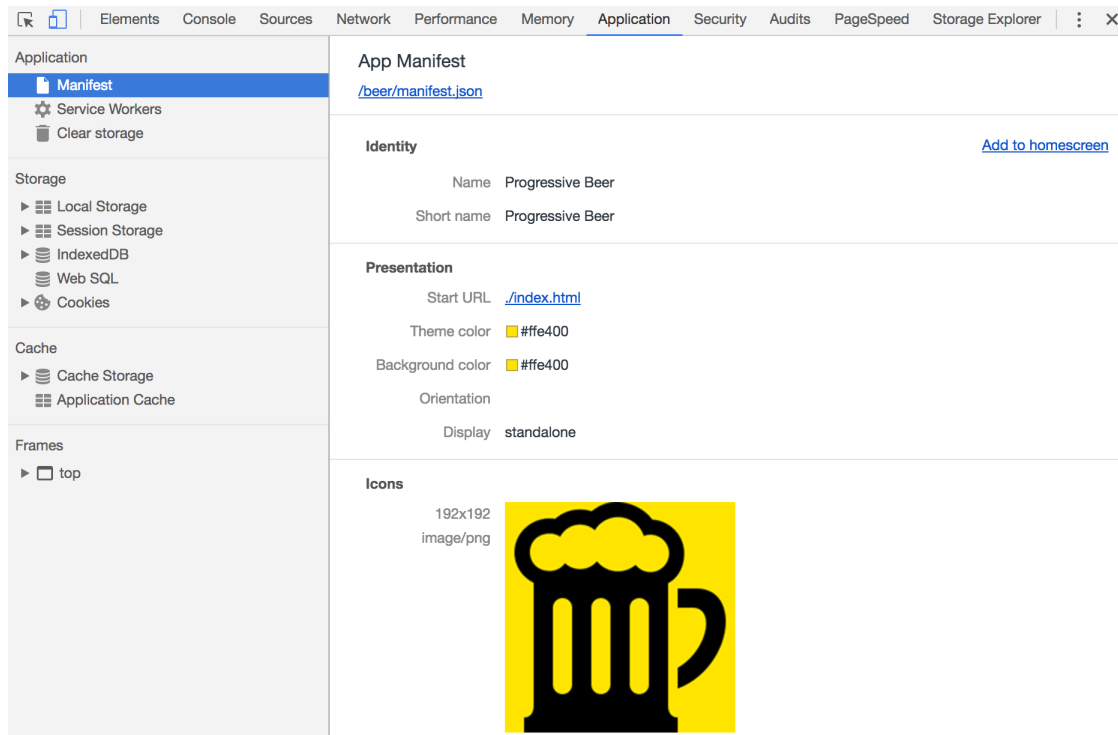
```
"scope" : "/myapp"
```

- scope 밖의 웹 페이지로 이동할 경우 새로운 브라우저를 실행한다.
- 위 문제를 피하는 방법으로 `` 사용하지 않고 deep link 를 사용
- deep link 를 이용해 모바일 앱과 웹 앱 간의 매끄러운 전환도 가능

'deep link' : 설치된 웹 앱의 지정된 scope 안의 url 들을 지칭 = 같은 도메인

Web App Manifest 디버깅

- 크롬 개발자 도구의 **Application tab** 을 이용하여 설정 정보 확인가능
- **앱 아이콘 설치** 등을 테스트 해볼 수 있다.



까먹기 전에 같이 실습

앱의 상세 정보를 갖는 manifest.json 파일을 생성하여 개발자 도구 `Application` 에서 확인

1. index.html 생성
2. manifest.json 생성
3. 앞에서 배운 name, icon, display, background_color 등 지정
4. node.js로 로컬 서버 구성
5. `node server.js` 로 서버 구동 후 `localhost:3000` 접속
6. 개발자도구 `Application` 으로 manifest 파일 내용 확인

지원되는 브라우저 (17년 5월 기준)

- Chrome 42 이상
- Firefox
- Opera
- Samsung Internet

참고

- [Web App Manifest Spec, W3C](#)
- [Web App Manifest Spec, html5doctor](#)
- [Web App Manifest Spec, MDN](#)
- [Getting Started w PWA, Addy](#)
- [Don't use iOS meta tags irresponsibly](#)
- [Understanding the manifest](#)
- [포켓몬 도감 PWA](#)
- [Fullscreen Image](#)

끝