

---

## 서비스워커 기술 및 표준화 동향

황현서\* · 김성현\* · 정용진\* · 박종근\* · 김태용\* · 김태환\* · 문일영\*

\*한국기술교육대학교

### Service Worker Technology and Standardization

Hyun-seo Hwang\* · Sung-hyun Kim\* · Yong-jin Jung\* · Jong-geun Park\* · Tae-yong Kim\* ·

Tae-hwan Kim\* · Il-young Moon\*

\*Korea University of Technology and Education

E-mail : smilebear1@koreatech.ac.kr

### 요 약

최근 구글과 모질라에 의해 개발되고 있는 새로운 브라우저 표준인 ‘서비스워커’로 인해 미래의 사용자들은 즐겨 찾는 웹 사이트들을 오프라인 상태에서 이용할 수 있을 것으로 전망된다. 구글에서는 웹 사이트가 사용자의 요청에 항상 응답하도록 만드는 새로운 브라우저 표준을 개발하고 있다. 서비스워커는 사용자의 브라우저에 오프라인 작업을 할 수 있는 공간을 확보해 웹 사이트가 각종 문서 정보를 저장하고 필요한 리소스를 제공한다. 그리고 브라우저와 서버 사이의 데이터 교환 작업을 크게 줄여주기 때문에 웹 페이지의 속도가 빨라진다. 웹앱을 오프라인에서도 쓸 수 있는 네이티브 앱처럼 만들어줄 뿐 아니라, 설치하지 않고도 실행되는 기존 웹앱의 특성도 더욱 강화해 준다는 점에서 파괴력이 높은 기술이다. 서비스워커 규격은 웹앱 사용 경험을 매우 향상 시킬 수 있으며 향후 플랫폼으로 발전해 가는 웹의 버전을 보여주는 혁신적인 기술이다. 서비스워커는 HTML5 최종 표준안에는 포함되지 않았으나 현재 표준화작업이 계속 진행 중에 있다. 향후 서비스워커 기술이 웹 브라우저에 적용이 되면 어떠한 일들이 펼쳐질지 기대해 본다.

### ABSTRACT

Recently, due to the standard of a new browser developed by the Google and Mozilla “Service Worker“, future users is expected to be able to make use of favorite Web sites offline. Google’s is, Web sites have developed a standard of a new browser so as to always respond to user requests. Service Worker, websites that provide space capable of offline work to the user’s browser to store various document information, to provide the necessary resources. Then, in order to greatly reduce the data exchange operations between the browser and the server, the speed of the Web page increases. Not only cooks as native app that can use the Web application offline, in that us to also further enhance the characteristics of an existing Web application that is running without installing destructive high technology. Service worker specifications, use experience of Web application is very can be improved, is an innovative technology indicates the version of the web evolve as the future of the platform. Service Worker is not included in HTML5 standard final, is currently being continued standardization. Future Service Worker technology I expect what kind of thing unfolds when applied to the Web browser.

### 키워드

Service Worker, HTML5, WebGL, Web Component, App Cache

## I. 서론

초기 웹은 다양한 서버에 저장되어 있는 문서를 연결하기 위해 요청 및 응답의 매커니즘을 기반으로 동작하도록 설계되었다. 웹은 서버와 클라이언트가 네트워크를 통해 리소스를 송수신하도록 처리하고 있으며, 웹 페이지 자체의 성능 외에도 근본적으로 오프라인을 기반으로 하는 실행환경은 아니다. 인터넷이 끊긴 브라우저로 애플리케이션을 쓸 수 있게 해줄 웹 표준 기술이 개발되고 있다. 웹 표준화단체인 W3C 주관으로 열린 삼성오픈소스컨퍼런스에서 구글과 모질라의 참여로 표준화 작업이 한창인 기술 서비스워커 소개와 전망을 구체적으로 제시하였다.

서비스워커는 1차적으로 이러한 오프라인 문제를 해결하기 위한 시작점이다. 서비스워커는 오프라인 상태에서도 브라우저를 통해 웹앱이 작동가능하도록 함으로써, 웹앱을 네이티브앱처럼 사용할 수 있게 해주는 기술이다. 현재의 네이티브 앱들의 기능과 유사한 형태의 웹 애플리케이션을 구현할 때 가장 어려운 부분 중 하나가 오프라인이다. 서비스워커는 브라우저 측에서 동작하는 이벤트 기반의 시스템 워커이며, 이를 이용하여 오프라인 앱을 위한 리소스 관리, 원격 푸시 등을 직접 관리할 수 있다. 서비스워커 이전에 App Cache 라는 웹 상에서 사용자에게 오프라인 경험을 제공하는 또 다른 API가 있었다. App Cache의 주요이슈는 싱글 페이지 웹 앱에만 특별히 잘 동작하지만 멀티 페이지 사이트에는 그러하지 않은 디자인이다.

서비스워커는 이러한 문제를 피할 수 있도록 디자인되었다. 서비스워커는 HTML5 최종표준안에는 포함되지 않았으나, 현재 표준화 작업이 계속 진행 중이다. 세계 브라우저 개발자 및 웹 개발자들의 많은 관심을 받고 있으며, 크롬과 파이어폭스 일부 버전에서는 서비스워커 기능을 제공하고 있다. 구글과 모질라에서 서비스워커 표준화 작업에 참여하고 있으며, 최종표준안으로 확정되기 전에 크롬과 파이어폭스에 서비스워커 기능이 도입될 것으로 예상된다.



그림 1. W3C의 ‘서비스 워커’ 표준화 현황

## II. 관련기술

### 2.1 웹 컴포넌트(Web Component)

웹 컴포넌트는 HTML 문서 내에 특정 기능을 수행하는 컴포넌트를 태그 형태로 삽입시킬 수 있는 기술로, HTML5 표준에 기반하고 있다. 웹 컴포넌트의 장점은 재사용 및 따로 수정이 가능하다. 또한, 다른 HTML 요소에 영향을 주지 않으면서 웹 컴포넌트 개체만 독립적으로 동작이 가능하다는 것이다. 개발자들은 이미 만들어 놓은 웹 컴포넌트를 사용하여 웹 개발에 소요되는 시간을 단축시킬 수 있으며, 유지보수를 용이하게 할 수 있다. HTML이 기본으로 제공하는 엘리먼트는 브라우저와 운영체제에 따라 다르게 보이고, 진화하는 웹 환경에 대응하기에 한계가 있어 자바스크립트 컴포넌트로 보완해왔었다. 그러나 자바스크립트는 사용이 어렵고 크기가 커서 실행속도가 느리다는 단점이 있으며, 이를 보완하기 위해 W3C는 기존의 컴포넌트 기술을 웹에서 적용할 수 있도록 웹 컴포넌트 기술을 만들었다. 웹 컴포넌트의 표준화로 인해 향후 컴포넌트를 쉽게 사용하기 위한 다양한 서비스들이 개발될 것으로 예상된다.

### 2.2 웹RTC(WebRTC)

웹RTC(Web Real Time Communication)는 HTML5에 포함된 기술로, 별도의 플러그인 없이 브라우저 내에서 오디오 및 비디오 커뮤니케이션을 가능케 하는 오픈소스 API이다. 기존의 실시간 통신 서비스들은 플러그인이 필요하고 클라이언트-서버간 데이터 통신 기능을 제공하고 있다. 하지만 웹RTC는 플러그인 없이 클라이언트간 P2P 통신 기능을 제공한다. 웹RTC는 서버를 거치지 않기 때문에 빠른 데이터 송수신이 가능하고, 플러그인 설치가 필요 없어 브라우저상에서 손쉽게 음성 및 화상통화가 가능하다는 장점으로 인해 차세대 커뮤니케이션 기술로 각광받고 있다. 또한, 구글이 주도하고 있는 웹RTC 시장에 2014년 마이크로소프트와 통신장비업체인 에릭슨까지 가세함으로써, 웹RTC 분야는 향후 치열한 경쟁터가 될 것으로 예상된다.

### 2.3 웹GL(WebGL)

웹GL(Web Graphic Library)은 웹 기반의 그래픽 라이브러리로, 플러그인 사용 없이 웹 브라우저에서 2D 및 인터랙티브한 3D 그래픽을 사용할 수 있게 하는 API기술이다. 브라우저가 사용자의 PC GPU(Graphic Processing Unit)를 직접 제어함으로써 웹 화면 렌더링을 빠르게 처리할 수 있는 장점이 있다. WebGL은 자바스크립트를 이용한

크로스 플랫폼 API으로 대부분의 브라우저에서 호환이 가능하다. WebGL은 비영리단체인 크로노스 그룹(Khronos Group)이 표준화를 주도하고 있으며, 구글, 모질라, 오페라, 애플 등이 WebGL 워킹그룹에서 활동하고 있다.

### III. 서비스워커 표준화동향

서비스워커는 오프라인 상태에서도 브라우저를 통해 웹앱이 작동 가능하도록 함으로써, 웹앱을 네이티브앱처럼 사용할 수 있게 해 주는 기술이다. 네트워크 연결이 끊기는 오프라인 상태가 되었을 때, 네트워크가 복구될 때까지 저장된 페이지나 다른 기능들을 제공함으로써 웹앱을 네이티브앱처럼 사용이 가능하다. 또한, 서비스워커는 기존의 웹앱이 매번 실행할 때마다 모든 리소스를 로딩하기 때문에 느려질 수 밖에 없는 단점을 보완함으로써, 온라인 상태에서도 웹앱의 속도와 성능을 네이티브 앱 수준으로 끌어올릴 수 있는 기술로 평가받고 있다.

서비스워커는 웹 리소스 중 캐싱할 대상을 지정해서 이를 사용자 디바이스에 저장하여, 사용자가 웹앱을 실행할 때 디바이스에 저장된 데이터를 먼저 가져오고 나머지를 온라인에서 가져오는 원리이다. 즉, 사용자가 특정 웹 사이트를 요청하면 처음에는 모든 웹 페이지를 로딩하지만, 다음부터는 디바이스에 캐싱된 웹 페이지를 먼저 불러온 후, 나머지 추가로 필요한 데이터만 네트워크에서 로딩하는 방식이다.

서비스워커는 모든 작업을 백그라운드에서 진행함으로써 브라우저를 실행하지 않은 상태에서도 푸시 API 사용이 가능하며, 이에 따라 기존에는 네이티브앱에서만 가능했던 푸시 서비스가 웹앱에서도 사용이 가능할 것으로 전망된다. 또한, 서비스워커는 웹 페이지와 완전히 별개인 생명주기(Life Cycle)를 가지고 있으며, register를 통해 백그라운드에서 서비스워커의 설치가 시작된다. 설치 과정에서 리소스를 캐싱하고자 할 때 모든 파일이 성공적으로 캐싱되고 나면 서비스워커는 설치 상태가 되며, 만약 어떠한 파일이라도 다운로드나 캐싱에 실패하면 서비스워커는 활성화되지 않는다. 활성화 단계 후 서비스워커는 페이지가 등록된 서비스워커는 범주 내에 있는 모든 페이지를 제어하게 되며 페이지로부터 네트워크 요청이나 메시지가 생성될 때 발생하는 fetch와 message를 제어하게 된다.

서비스워커는 지대한 영향을 줄 수 있는 서비스워커 모듈의 신뢰성을 위해 HTTPS 기반으로만 동작이 된다. 서비스워커는 HTML5 최종표준안에는 포함되지 않았지만, 현재 표준화 작업이 계속 진행 중에 있다.

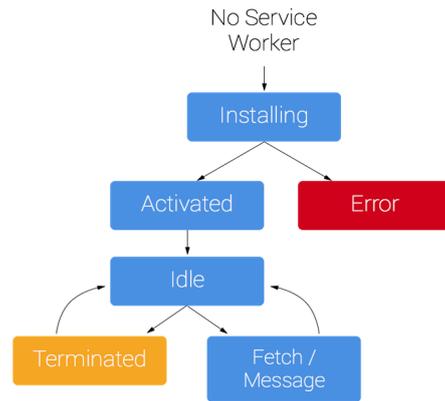


그림 2. 서비스워커의 생명주기

### IV. 결론

현재 네이티브 앱들의 기능과 유사한 형태의 웹 애플리케이션을 구현할 때 가장 어려운 부분이 오프라인이다. 서비스워커는 이러한 문제를 해결하기 위한 시작점이며, 네이티브 애플리케이션의 동작 흐름을 웹으로 가져오기 위한 가장 중요한 기능이다. 웹 컴포넌트와 마찬가지로 네이티브 애플리케이션 개발자 입장에서는 익숙한 개념들을 가지고 있다. 웹 컴포넌트가 컴포넌트의 재사용성을 추구하는 개념이라면 서비스 워커는 웹 애플리케이션의 근본적인 한계를 탈출하기 위한 내용을 담고 있다. 서비스워커는 시각적인 기능을 가지고 있지는 않지만, 이로 인해 가능해지는 많은 기능들이 네이티브 애플리케이션에서 많이 활용되는 기능들이므로 웹 애플리케이션 개발에 있어 큰 패러다임 변화를 이끌어낼 가능성이 있다.

따라서, 웹 애플리케이션의 오프라인 기능 지원이 시작되고 있는 지금 반드시 익숙해져야 하는 대표적인 기능이다. HTML5 최종표준안에는 포함되어 있지 않았지만, 현재 구글과 모질라의 참여로 표준화 작업이 진행 중에 있다. 표준화 작업이 완료되면 어떠한 일들이 벌어질지 기대해본다.

### 감사의 글

본 논문은 중소기업청에서 지원하는 2014년도 창업성장기술개발사업 이공계 꿈나무과제(NO. S2194494)의 연구수행으로 인한 결과물임을 밝힙니다.

## 참고문헌

- [1] 최현희, 김근형, "HTML5 기반 HTTP 스트리밍 환경에서의 서비스 이동성 연구", *Journal of Korea Multimedia Society*, vol. 14, no. 7, pp. 905-916, 2011
- [2] 정현주, 변정용, "웹서비스 기반의 컴포넌트 재사용에 관한 설계 및 구현", *Korea Information Science Society*, vol. 1, no. 1, pp. 252-256, 2007
- [3] Nguyen Ngoc Vinh, Younghan Kim, "Real-time Notification on WebRTC based application", *The Korean Institute of Communications and Information Science*, pp. 1072-1073, 2013
- [4] 이은민, "HTML5가 웹 환경에 미치는 영향", *Journal of Computing and Engineering*, vol. 29, no. 6, pp. 55-60, 2011
- [5] 안병현, 김병정, "HTML5 표준화 현황과 활용 사례", *Journal of Computing and Engineering*, vol. 30, no. 5, pp. 10-15, 2010