

Progressive Web App

What is Progressive Web App ?!

2015년, 구글 엔지니어 Alex Russell

“Progressive Web Apps: Escaping Tabs Without Losing Our Soul”

<https://infrequently.org/2015/06/progressive-apps-escaping-tabs-without-losing-our-soul/>

“생동감있는 앱을 만드는데에
더이상 웹 기술을 포기할 필요가 없어졌다.”

App-like Web But Not App

"Progressive Web App은 웹의 장점과 앱의 장점을 결합한 환경입니다."

<https://developers.google.com/web/fundamentals/codelabs/your-first-pwapp/?hl=ko>

PWA 특징 - 0

GUIDES CODE LABS SAMPLES

노드입니다.

Home

[Your First Progressive Web App](#)

Debugging Service Workers

Your First Offline Web App

Frictionless Payment with Payment

Request API

Your First Web Push Notification

Find and Fix Web App Performance
Issues

Your First Multi-Device Site

Progressive Web App이란?

Progressive Web App의 특징은 다음과 같습니다.

- **프로그레시브** - 점진적인 개선을 통해 작성되므로, 어떤 브라우저를 선택하든 상관없이 모든 사용자에게 적합합니다.
- **반응형** - 데스크톱, 모바일, 태블릿 등 모든 폼 팩터에 맞습니다.
- **연결 독립적** - 서비스 워커를 사용하여 오프라인이나 느린 네트워크에서 작동하도록 향상됩니다.
- **앱과 유사** - 앱 셀 모드에서 작성되기 때문에 앱 스타일의 상호작용 및 탐색 기능을 사용자에게 제공합니다.
- **최신 상태** - 서비스 워커 업데이트 프로세스 덕분에 항상 최신 상태로 유지됩니다.
- **안전** - HTTPS를 통해 제공되므로 스누핑이 차단되며, 콘텐츠가 변조되지 않도록 보장합니다.
- **검색 가능** - W3C 매니페스트 및 서비스 워커 등록 범위 덕분에 '애플리케이션'으로 식별되므로, 검색 엔진에서 검색이 가능합니다. **제약사항: https:// , 127.0.0.1, localhost에서만 작동함**
- **재참여 가능** - 푸시 알림과 같은 기능을 통해 쉽게 재참여가 가능합니다.
- **설치 가능** - 앱 스토어에서 씨름할 필요 없이 사용자가 자신에게 가장 유용한 앱을 홈 화면에 유지할 수 있습니다.
- **링크 연결 가능** - URL을 통해 손쉽게 공유할 수 있으며 복잡한 설치 작업이 불필요합니다.

이 코드랩에서는 Progressive Web App을 만드는 방법을 안내하며, 여기에는 Progressive Web App의 주요 원칙을 앱이 충족하도록 보장하는 디자인적인 고려 사항과 구체적인 구현 내용이 포함됩니다.

PWA 특징 - 주목할만한 특징 1

푸시알림이 가능해진다.

데스크탑(맥, 윈도우) & 모바일(안드로이드)

~~아이폰은 ;;;.... 팀쿡아 일하자..~~

PWA 특징 - 주목할만한 특징 2

오프라인에서도 사용가능(홈 화면 등록)

+

느린 와이파이 환경에서도 빠른 사이트

데이터를 미리 브라우저에 저장(캐싱)해두고, 재접속시에 저장되었는 데이터를 사용하기 때문에 빠름.

PWA 웹 예시

<https://bit.ly/2OVUAEs> 푸시알림 예시 영상

<https://bit.ly/2OoBRo1> 오프라인 예시 영상

데이터를 브라우저에 저장하고,
푸시 알림을 받는거는 누가하지???

Service Worker

서비스워커가 뭐냐

브라우저의 **백그라운드**에서 실행되는 **스크립트**

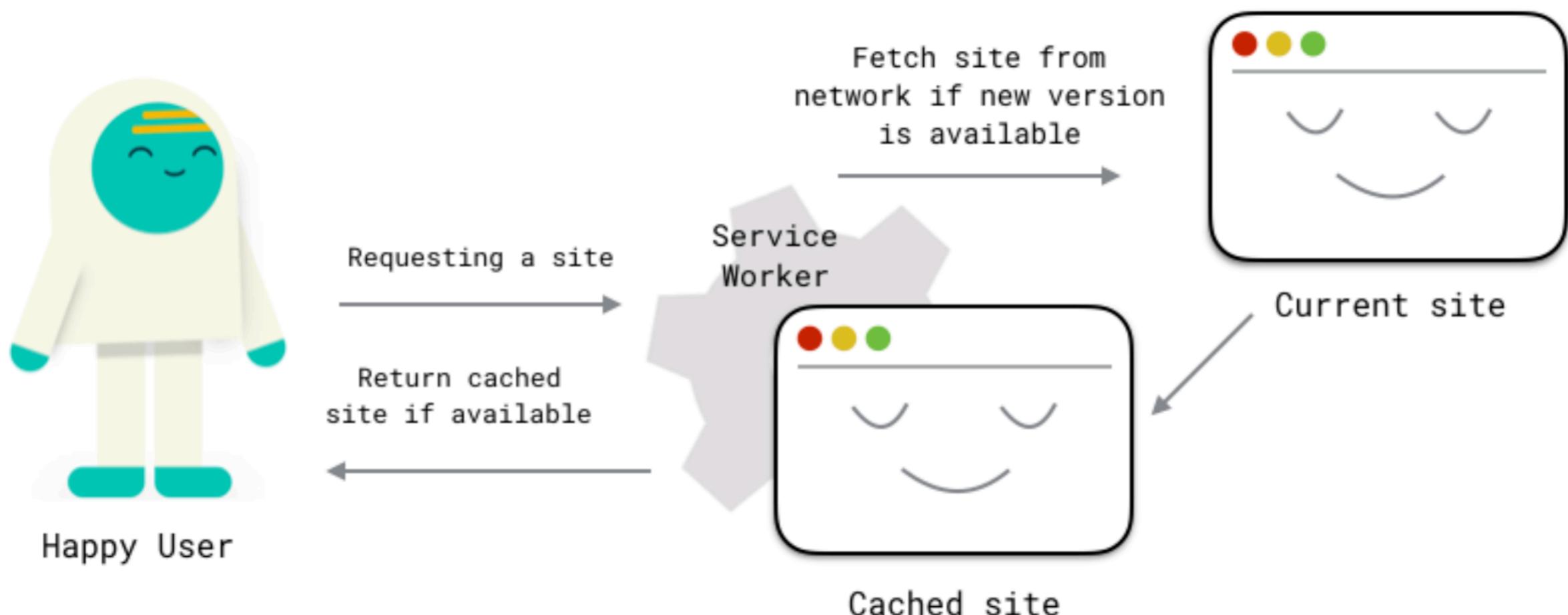
웹페이지와는 별개로 작동

서비스워커덕분에 오프라인 서비스, 푸시알림이 가능해짐.

브라우저에서 돌아가므로 자바스크립트 기반

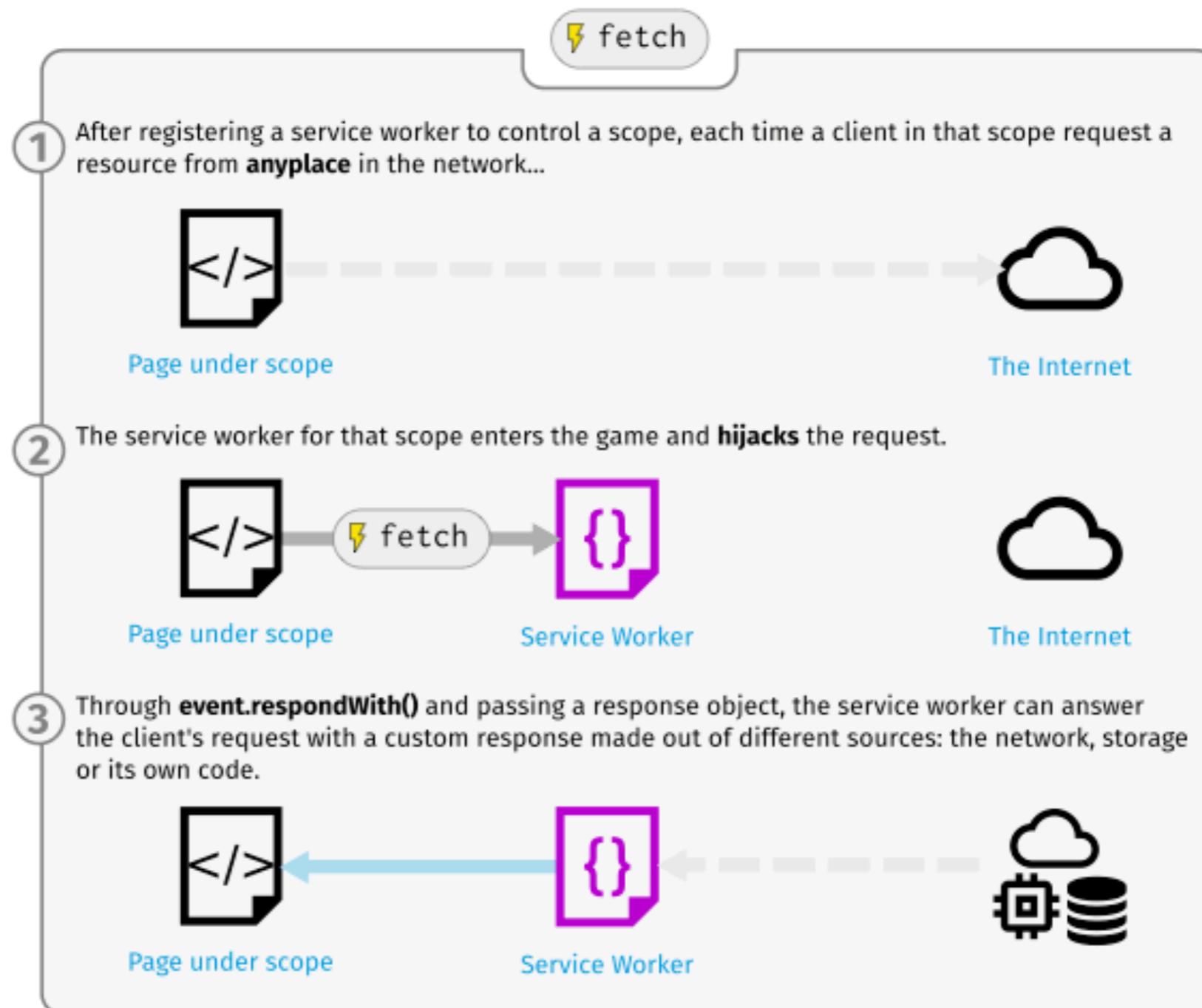
<https://developers.google.com/web/fundamentals/primers/service-workers/>

서비스워커가 뭐냐



이미지: <https://www.netlify.com/blog/2017/10/31/service-workers-explained/>

서비스워커가 뭐냐



이미지: https://developer.mozilla.org/en-US/docs/Web/API/Service_Worker_API/Using_Service_Workers

서비스워커가 뭐냐

즉 **서버에 요청해서 받아온 데이터는 서비스워커를 통해 저장(캐싱)해두었다가,**
나중에 똑같은 요청이 들어왔을때 저장된 데이터를 먼저 화면에 띄워줌.

(기존 데이터가 업데이트된 경우에는 저장되어있는 데이터를 새로운 데이터로 바꿔줄수있음.)

좀더 자세한 내용은 이 링크에서 확인가능하다.

<https://developers.google.com/web/fundamentals/instant-and-offline/offline-cookbook/>

Push Notification

With Service Worker

구독하는 프로세스



메세지 보내는 서버

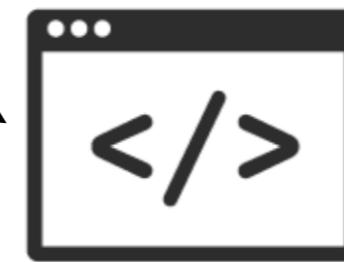


fcm.googleapis.com

사용자의 컴퓨터/폰



1. 사용자가 (서비스워커가 구축되었는)사이트 접속



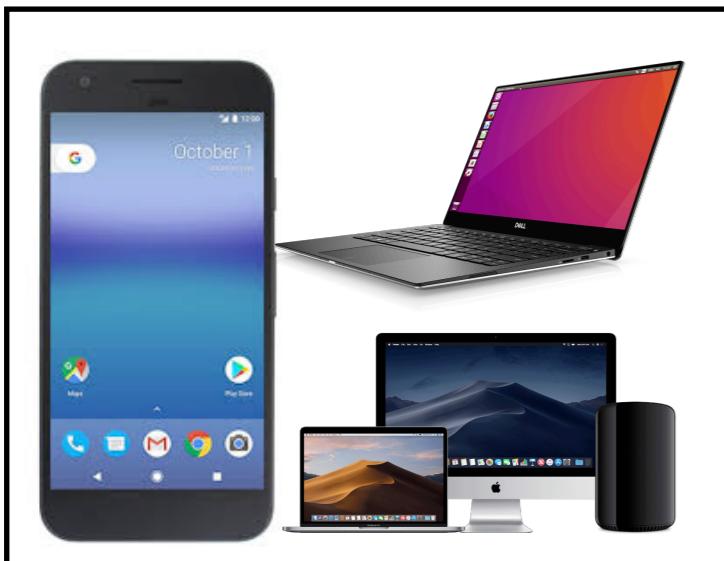
Page

웹서버



구독하는 프로세스

사용자의 컴퓨터/폰

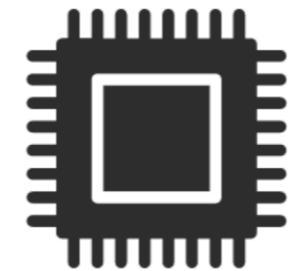


1. 사용자가 사이트 접속



2. 구독 버튼 클릭

Page



ServiceWorker

(사용자의 브라우저 내부에 있고
대체로 브라우저를 꺼도 서비스워커는 돌아감)

메세지 보내는 서버

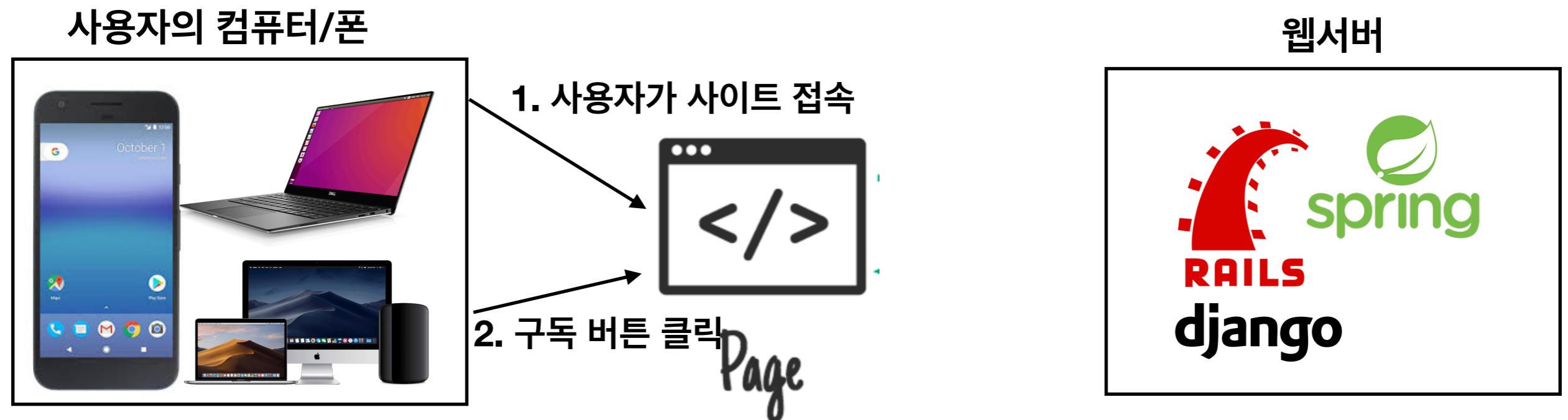
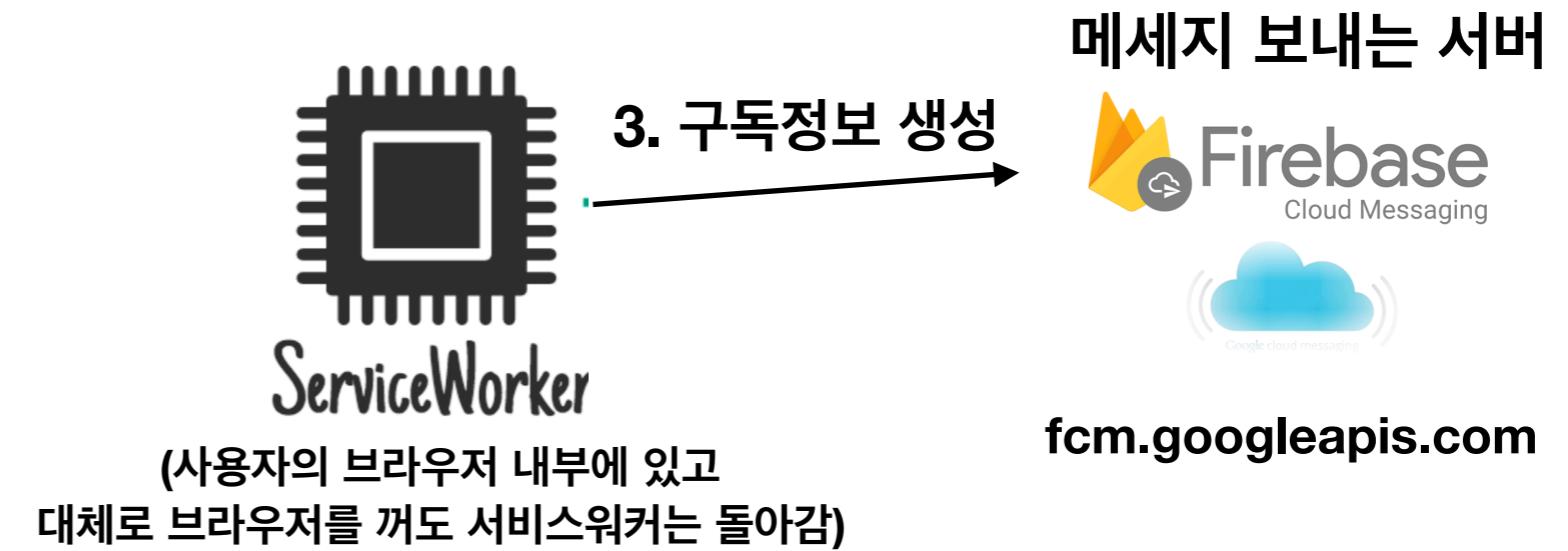


fcm.googleapis.com

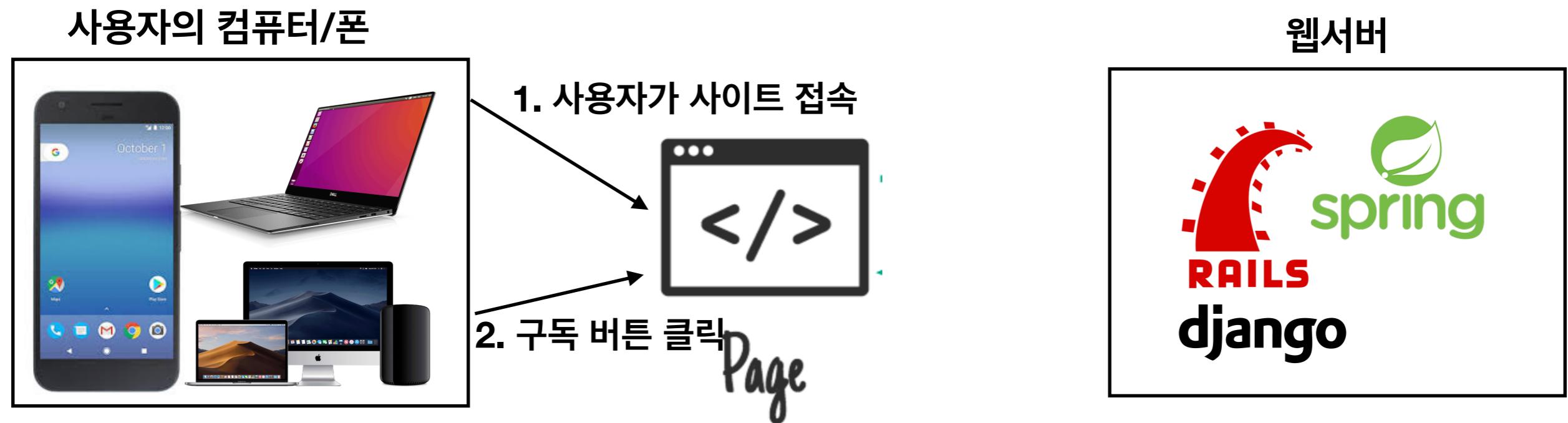
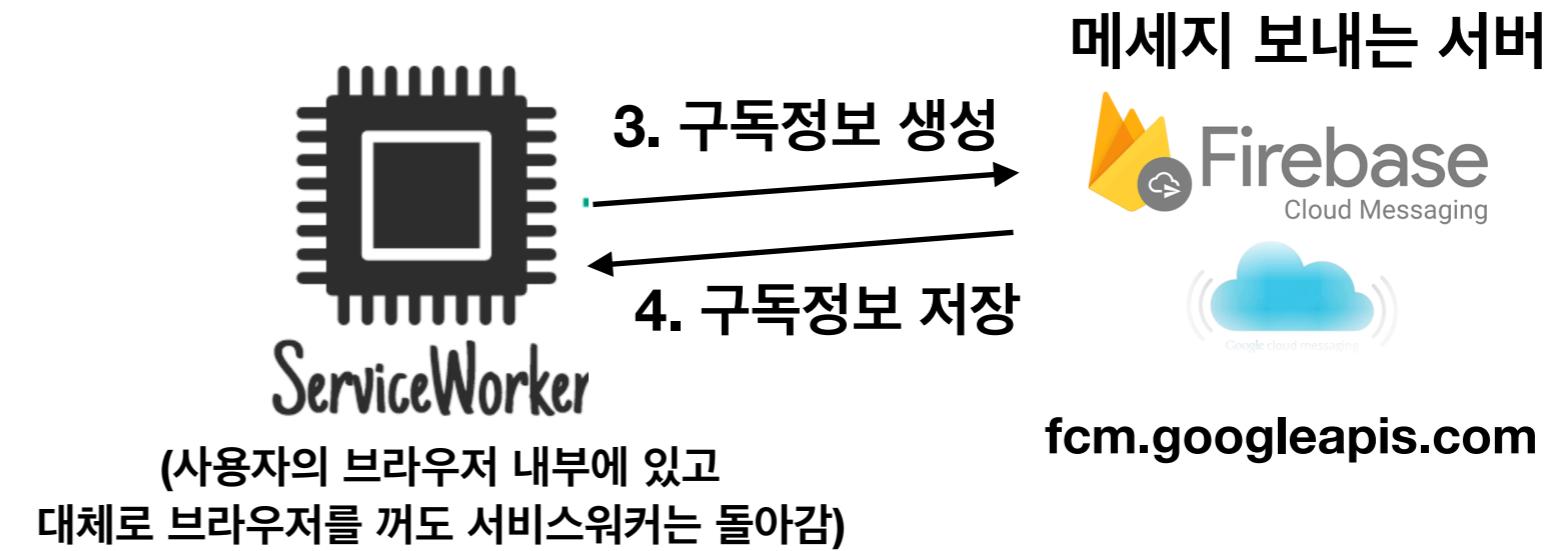
웹서버



구독하는 프로세스

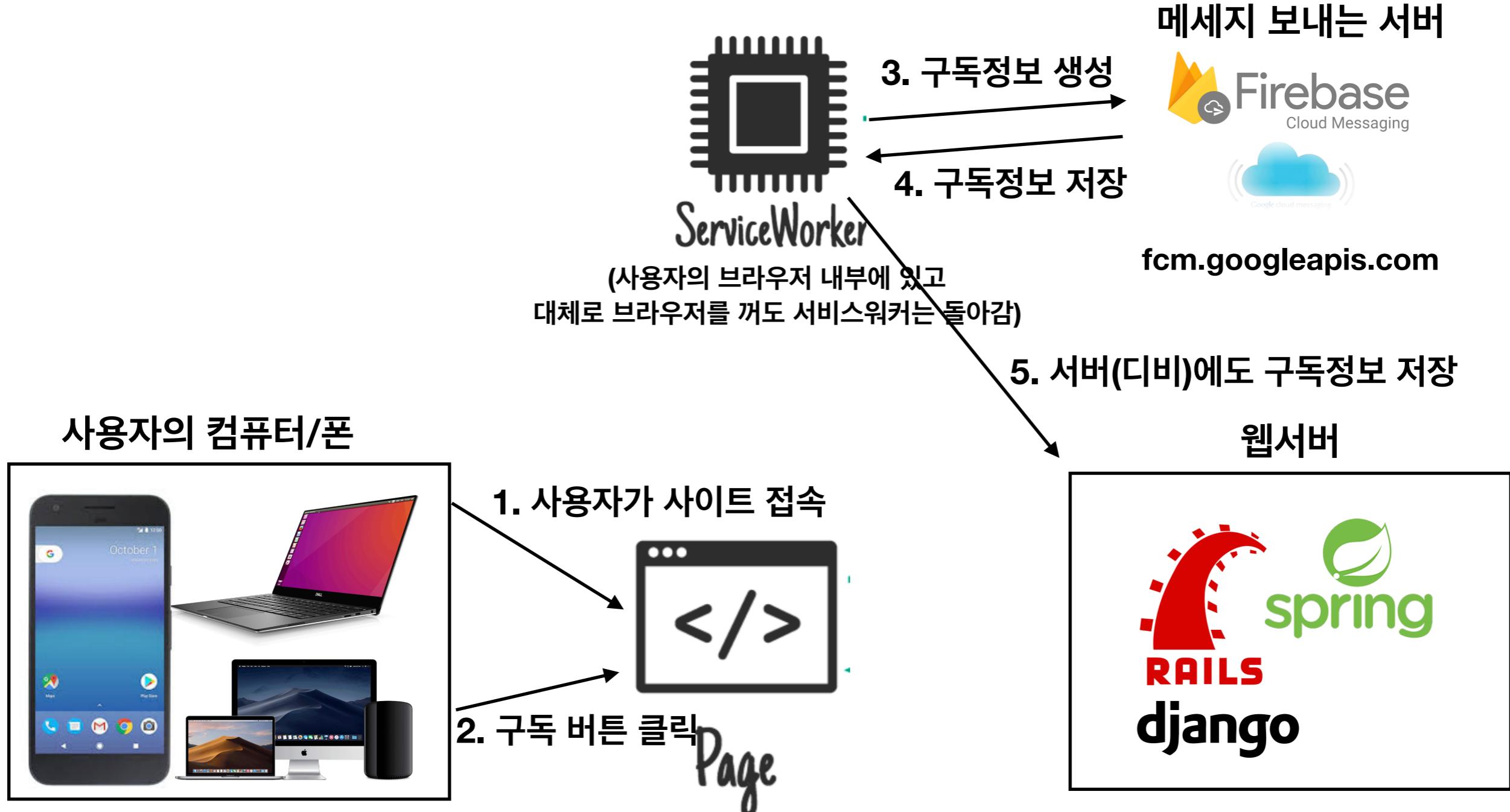


구독하는 프로세스



“구독”은 공식 API형태로 제공해주기에 구독정보 생성, 저장하는 코드는 메소드 호출하는 정도로 구현가능
(실제로 메소드이름이 subscribe임)

구독하는 프로세스

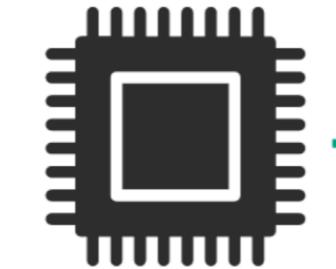


구독되어있는 상태에서 메세지 받는 프로세스

메세지 보내는 서버



fcm.googleapis.com



ServiceWorker

(사용자의 브라우저 내부에 있고
대체로 브라우저를 꺼도 서비스워커는 돌아감)

사용자의 컴퓨터/폰



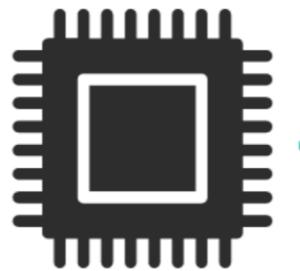
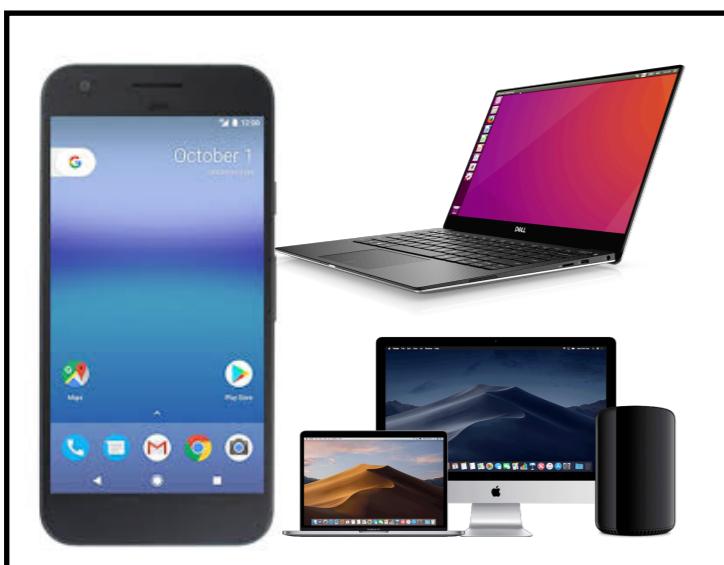
Page

웹서버



1. 새로운 게시글이 생성됨

구독되어있는 상태에서 메세지 받는 프로세스



ServiceWorker

(사용자의 브라우저 내부에 있고
대체로 브라우저를 꺼도 서비스워커는 돌아감)



Page

메세지 보내는 서버



fcm.googleapis.com

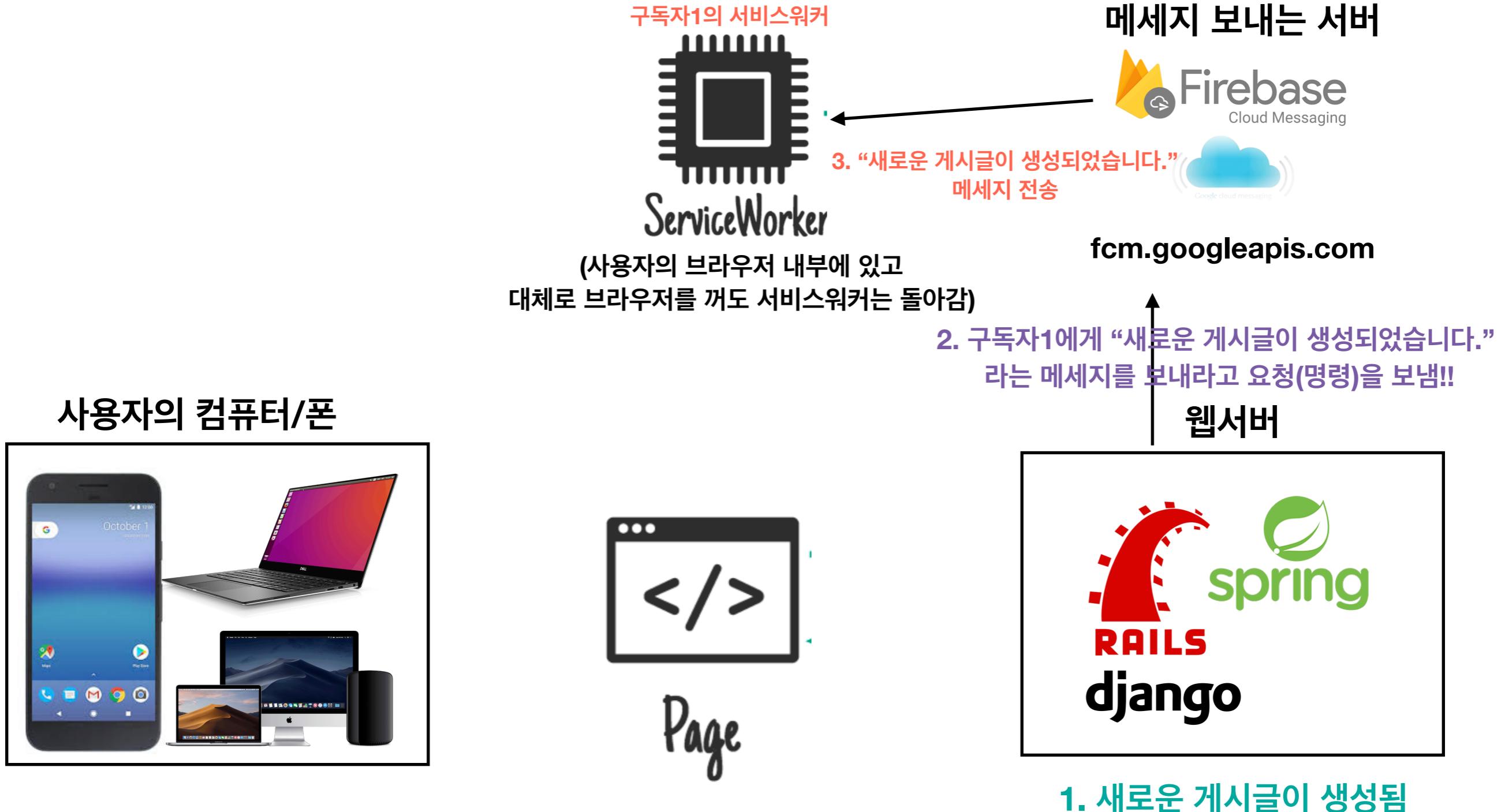
2. 구독자1에게 “새로운 게시글이 생성되었습니다.”
라는 메세지를 보내라고 요청(명령)을 보냄!!

웹서버

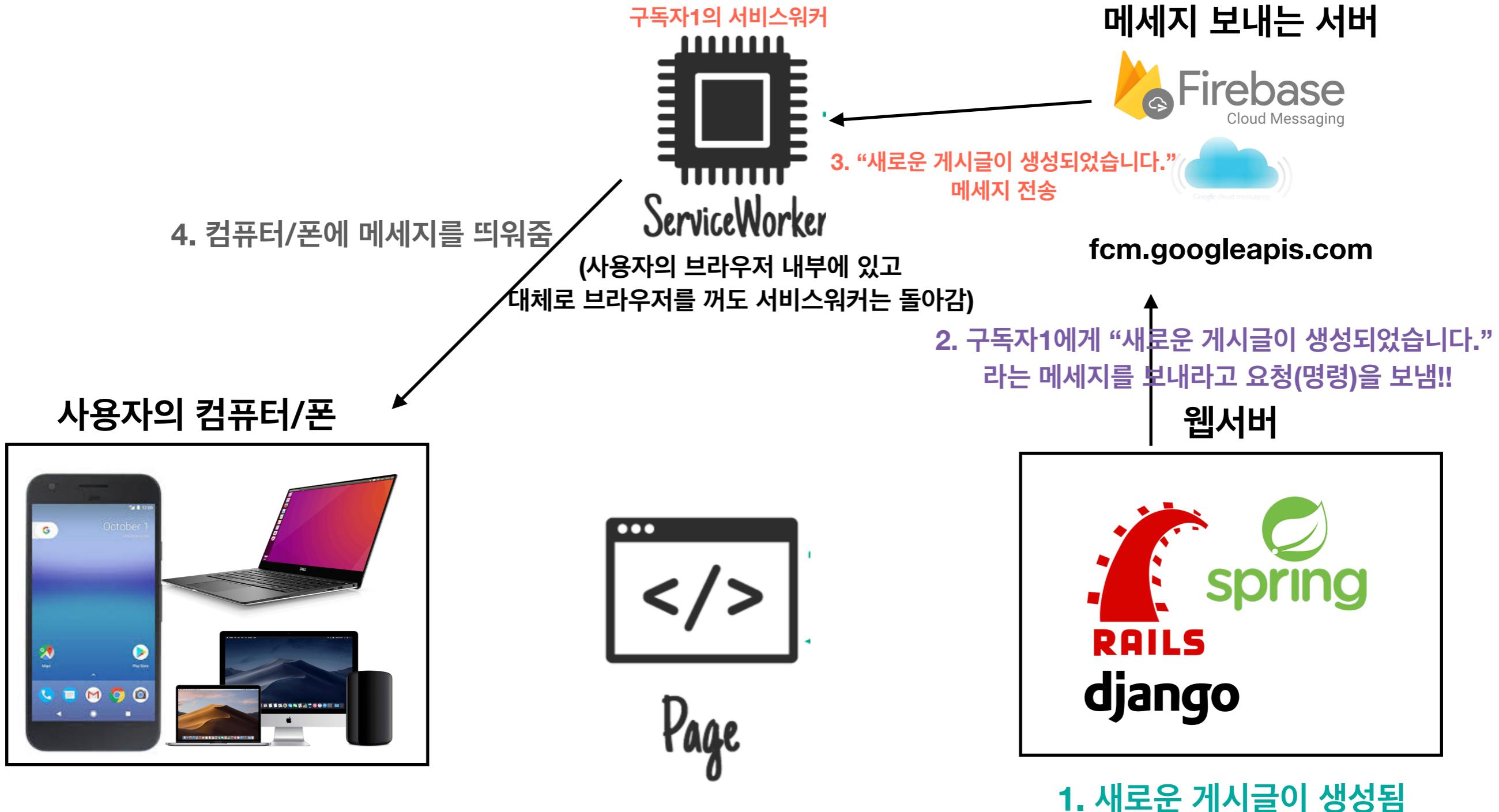


1. 새로운 게시글이 생성됨

구독되어있는 상태에서 메세지 받는 프로세스



구독되어있는 상태에서 메세지 받는 프로세스



구독되어있는 상태에서 메세지 받는 프로세스



사용자의 컴퓨터/폰



1. 새로운 게시글이 생성됨

만약 사용자가 페이지에 접속해있는 상태라면,
알림을 페이지에 띄우게 할수도 있다!
(이번 강의에서 다루지는 않음)



Home



Notifications



Messages

구현해보자

구현해보자

1. 서비스워커를 사용할수있는 사이트를 구현한다.
2. 서비스워커를 이용하여 푸시알림을 구현한다

Workspace name	Description
pwa-gogogo	Make a short description of your workspace

Team

Choose a team for this workspace...

[Hosted workspace](#) [Clone workspace](#) [Remote SSH workspace](#) [Salesforce](#)

 **Private**
This is a workspace for your eyes only

 **Public**
This will create a workspace for everybody to see

Clone from Git or Mercurial URL (optional)

e.g. ajaxorg/ace or git@github.com:ajaxorg/ace.git

Choose a template

 **HTML5**

 **Node.js**

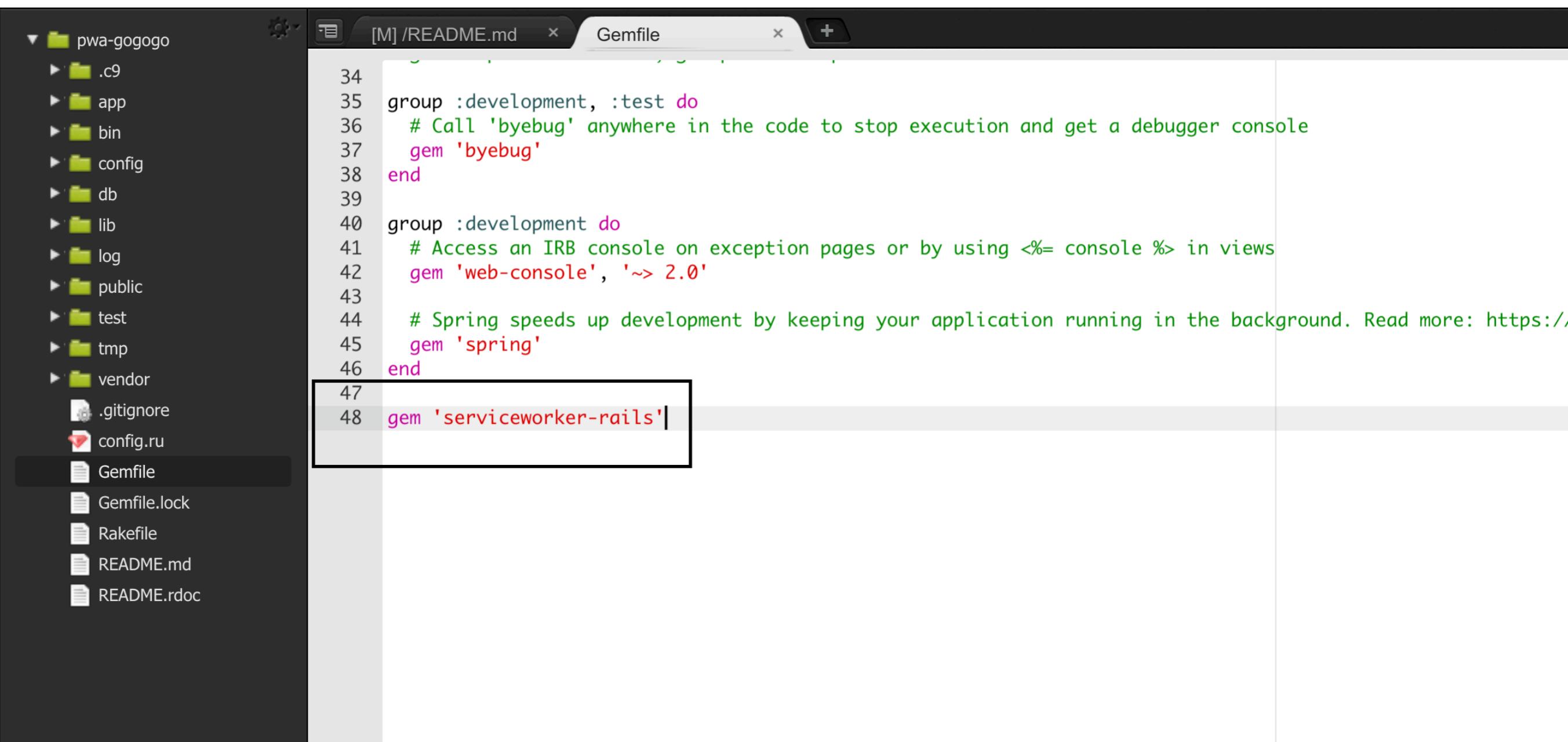
 **PHP**, Apache ...

 **Python**

 **django**
Django

 **Rails**
Ruby

편의를 위해 C9으로 진행하지만, 나중에 혼자 따라할때 Rails5 로컬환경에서 진행해도 별반 차이는 없다.
하지만, 주의할점은 https를 적용하지않고 배포하는 경우에는 서비스워커 코드를 짜도
서비스워커가 작동되지않는다. (단, 127.0.0.1 또는 localhost에서는 서비스워커가 작동됨)



The screenshot shows a code editor with a dark theme. On the left is a file tree for a project named 'pwa-gogogo'. The 'Gemfile' tab is active, showing the following code:

```
34
35 group :development, :test do
36   # Call 'byebug' anywhere in the code to stop execution and get a debugger console
37   gem 'byebug'
38 end
39
40 group :development do
41   # Access an IRB console on exception pages or by using <%= console %> in views
42   gem 'web-console', '~> 2.0'
43
44   # Spring speeds up development by keeping your application running in the background. Read more: https://...
45   gem 'spring'
46 end
47
48 gem 'serviceworker-rails'
```

The line 'gem 'serviceworker-rails'' is highlighted with a black rectangle.

gem 'serviceworker-rails'
\$ bundle install

```
limdongjin:~/workspace $ rails g serviceworker:install
Running via Spring preloader in process 1706
  create  app/assets/javascripts/manifest.json.erb
  create  app/assets/javascripts/serviceworker.js.erb
  create  app/assets/javascripts/serviceworker-companion.js
  create  config/initializers/serviceworker.rb
  append  app/assets/javascripts/application.js
  append  config/initializers/assets.rb
  insert  app/views/layouts/application.html.erb
  create  public/offline.html
```

rails g service worker:install

```
limdongjin:~/workspace $ rails g serviceworker:install
Running via Spring preloader in process 1706
  create  app/assets/javascripts/manifest.json.erb
  create  app/assets/javascripts/serviceworker.js.erb
  create  app/assets/javascripts/serviceworker-companion.js
  create  config/initializers/serviceworker.rb
  append  app/assets/javascripts/application.js
  append  config/initializers/assets.rb
  insert  app/views/layouts/application.html.erb
  create  public/offline.html
```

“그럼 서비스워커쓸려면 젬 무조건 써야함?” 또는 “젬으로 만든거니까 저 js 코드들도 레일즈에서만 작동?”

“ㄴㄴ 결국 이 젬은 서비스워커를 적용하기위해 필요한 [자바스크립트 코드](#),
설정파일코드를 생성해주는 정도일뿐임. 그렇기에 백엔드 서버의 언어가 무엇인지는 큰 상관이 없음.
왜냐면 프론트엔드의 로직이니까.”

```
limdongjin:~/workspace $ rails g serviceworker:install
Running via Spring preloader in process 1706
  create  app/assets/javascripts/manifest.json.erb
  create  app/assets/javascripts/serviceworker.js.erb
  create  app/assets/javascripts/serviceworker-companion.js
  create  config/initializers/serviceworker.rb
  append  app/assets/javascripts/application.js
  append  config/initializers/assets.rb
  insert  app/views/layouts/application.html.erb
  create  public/offline.html
```

“그럼 이 젬을 안쓰고 다른언어(프레임워크)로 서비스워커 구현할려면 어떻게함?”

“다른 언어에서도 이런 기본적인 틀을 자동으로 짜주는 라이브러리가 존재하는 경우가 많음.
그리고 애초에 저 코드들이 그리 복잡하지않고 **튜토리얼문서에 있는 코드들과 거의 유사해서**
굳이 젬이나 라이브러리 안쓰고도 문서보면서 충분히 구현 가능함.”

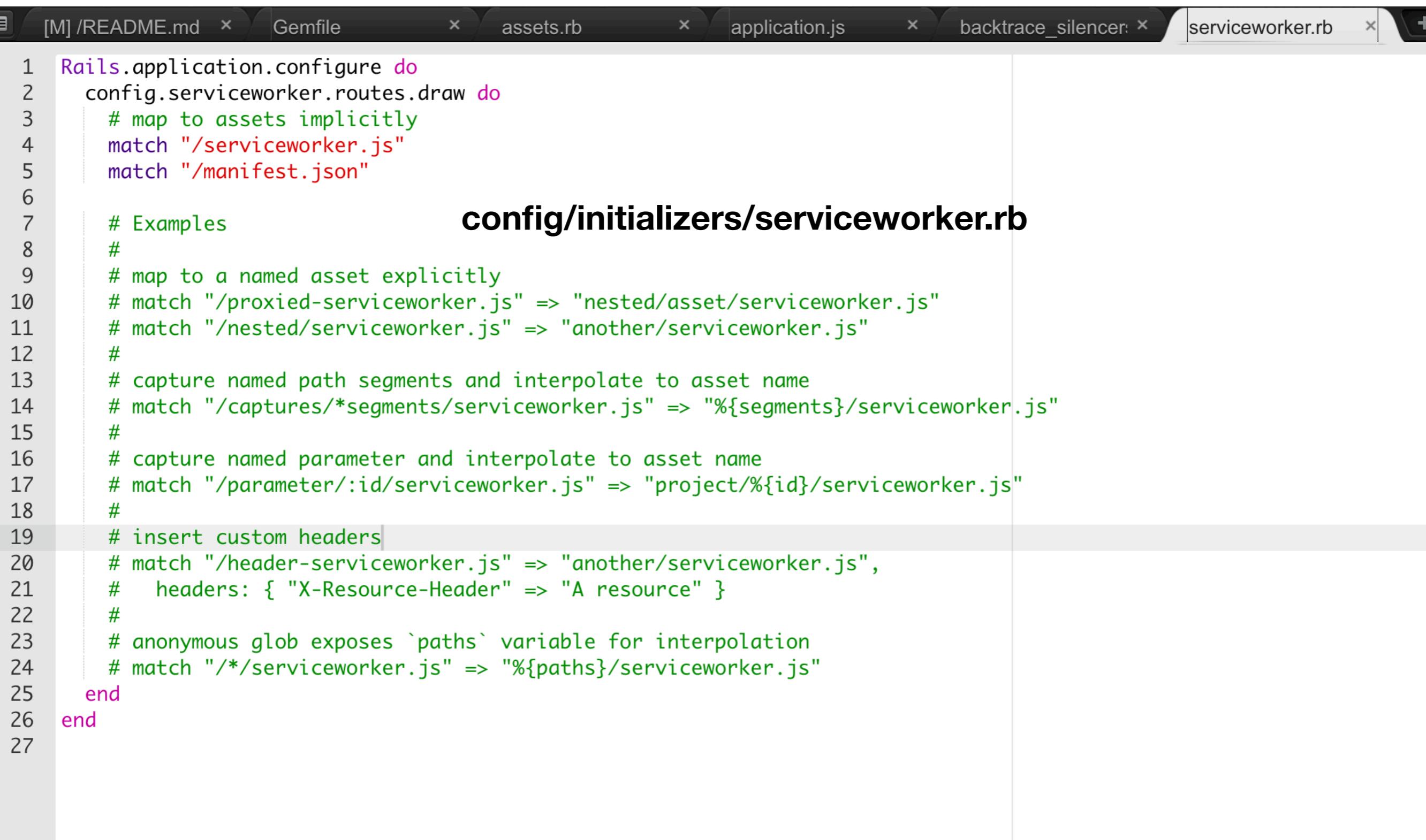
서비스워커를 사용할수있는 사이트를 구현하자.

생성된 코드 둘러보기 - 0

```
1 // This is a manifest file that'll be compiled into application.js, which will include all t
2 // listed below.
3 //
4 // Any JavaScript/Coffee file within this directory, lib/assets/javascripts, vendor/assets/j
5 // or any plugin's vendor/assets/javascripts directory can be referenced here using a relati
6 //
7 // It's not advisable to add code directly here, but if you do, it'll appear at the bottom c
8 // compiled file.
9 //
10 // Read Sprockets README (https://github.com/rails/sprockets#sprockets-directives) for detai
11 // about supported directives.
12 //
13 //= require jquery
14 //= require jquery_ujs
15 //= require turbolinks
16 //= require tree .
17 //= require serviceworker-companion
18 
```

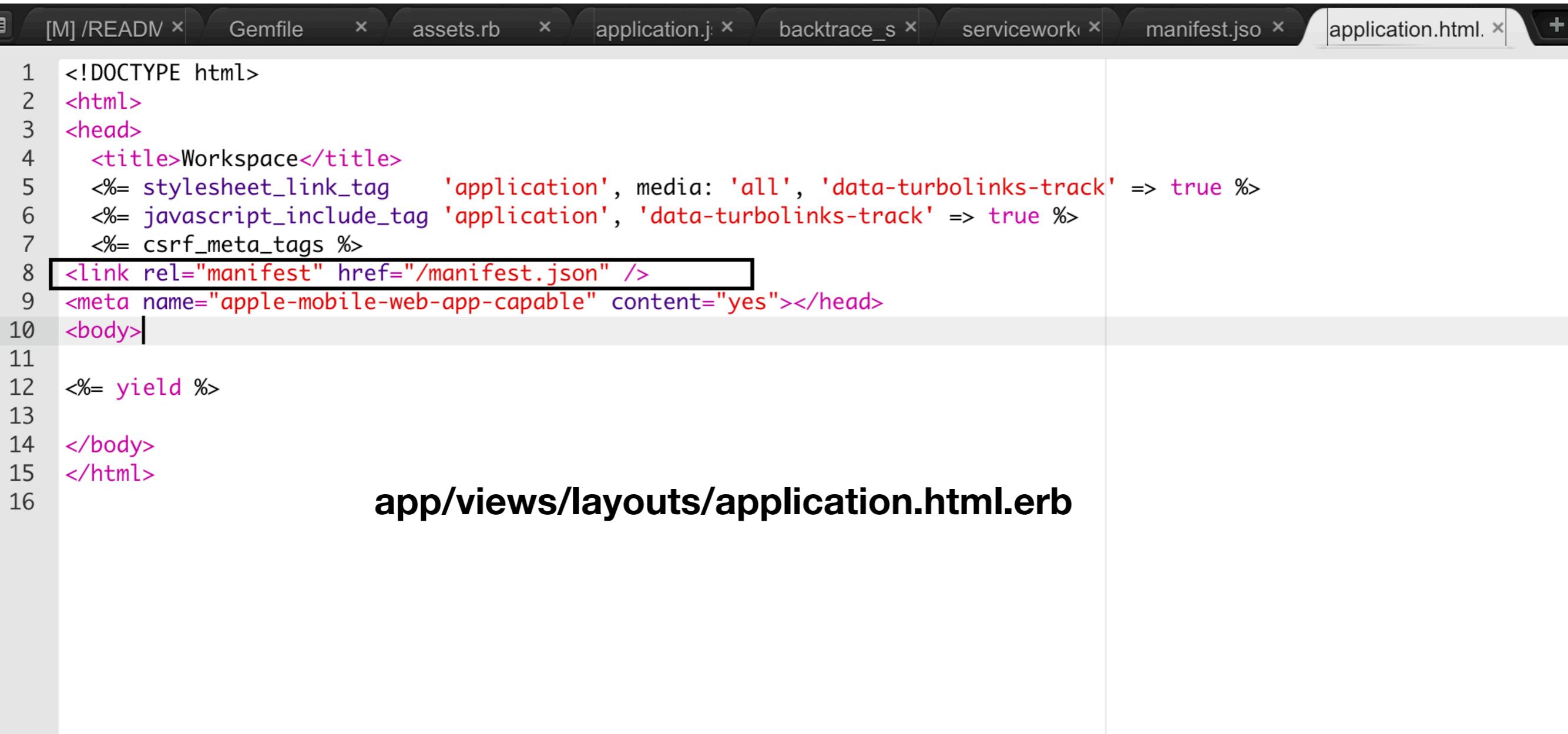
```
2
3 # Version of your assets, change this if you want to expire all your assets.
4 Rails.application.config.assets.version = '1.0'
5
6 # Add additional assets to the asset load path
7 # Rails.application.config.assets.paths <- %w[ #{Rails.root}/app/assets ]
8
9 # Precompile additional assets.
10 # application.js, application.css, and all non-JS/CSS in app/assets folder are already added.
11 # Rails.application.config.assets.precompile += %w( search.js )
12 Rails.configuration.assets.precompile += %w[serviceworker.js manifest.json]
13 
```

음 별거 없군



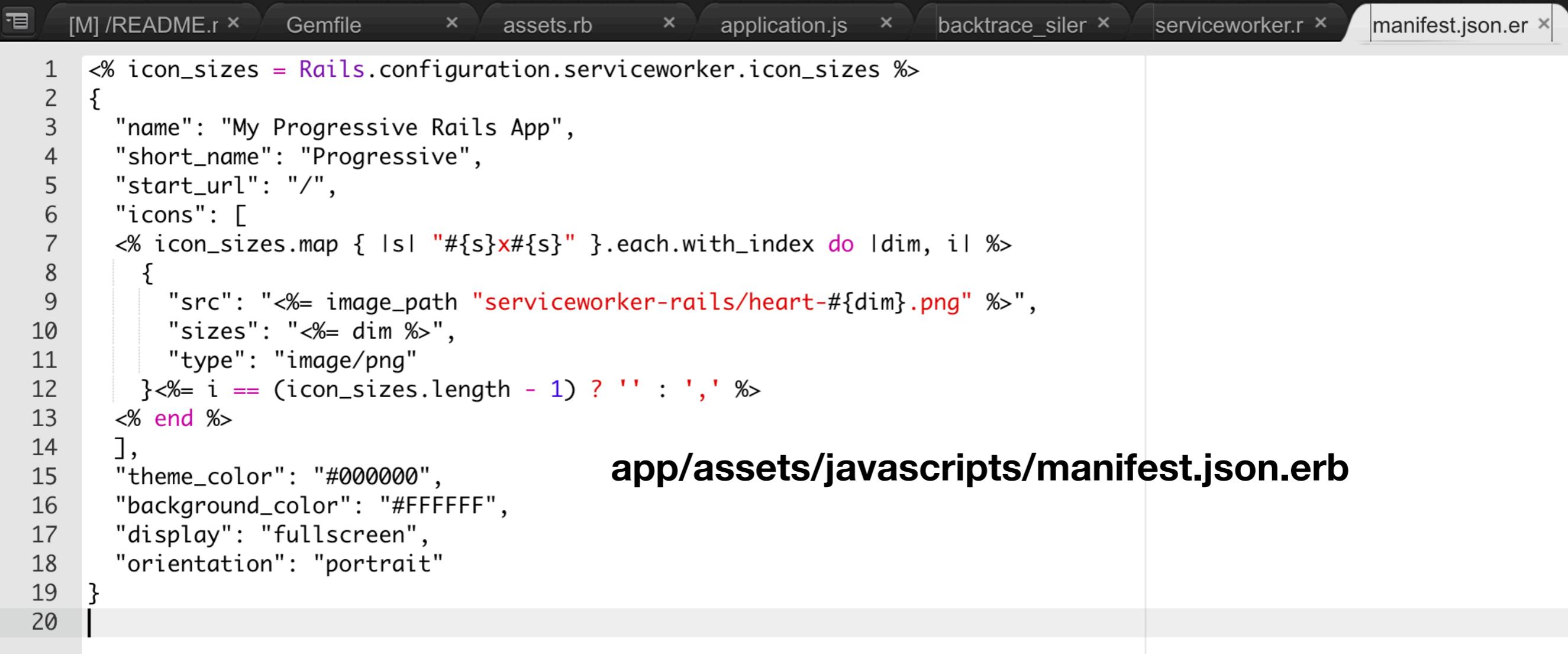
```
[M] /README.md x Gemfile x assets.rb x application.js x backtrace_silencer x serviceworker.rb x +  
1 Rails.application.configure do  
2   config.serviceworker.routes.draw do  
3     # map to assets implicitly  
4     match "/serviceworker.js"  
5     match "/manifest.json"  
6  
7     # Examples  
8     #  
9     # map to a named asset explicitly  
10    # match "/proxied-serviceworker.js" => "nested/asset/serviceworker.js"  
11    # match "/nested/serviceworker.js" => "another/serviceworker.js"  
12    #  
13    # capture named path segments and interpolate to asset name  
14    # match "/captures/*segments/serviceworker.js" => "%{segments}/serviceworker.js"  
15    #  
16    # capture named parameter and interpolate to asset name  
17    # match "/parameter/:id/serviceworker.js" => "project/%{id}/serviceworker.js"  
18    #  
19    # insert custom headers  
20    # match "/header-serviceworker.js" => "another/serviceworker.js",  
21    #   headers: { "X-Resource-Header" => "A resource" }  
22    #  
23    # anonymous glob exposes `paths` variable for interpolation  
24    # match "/*/serviceworker.js" => "%{paths}/serviceworker.js"  
25  end  
26 end  
27
```

config/initializers/serviceworker.rb



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Workspace</title>
5   <%= stylesheet_link_tag 'application', media: 'all', 'data-turbolinks-track' => true %>
6   <%= javascript_include_tag 'application', 'data-turbolinks-track' => true %>
7   <%= csrf_meta_tags %>
8   <link rel="manifest" href="/manifest.json" />
9   <meta name="apple-mobile-web-app-capable" content="yes"></head>
10 <body>
11
12 <%= yield %>
13
14 </body>
15 </html>
```

app/views/layouts/application.html.erb



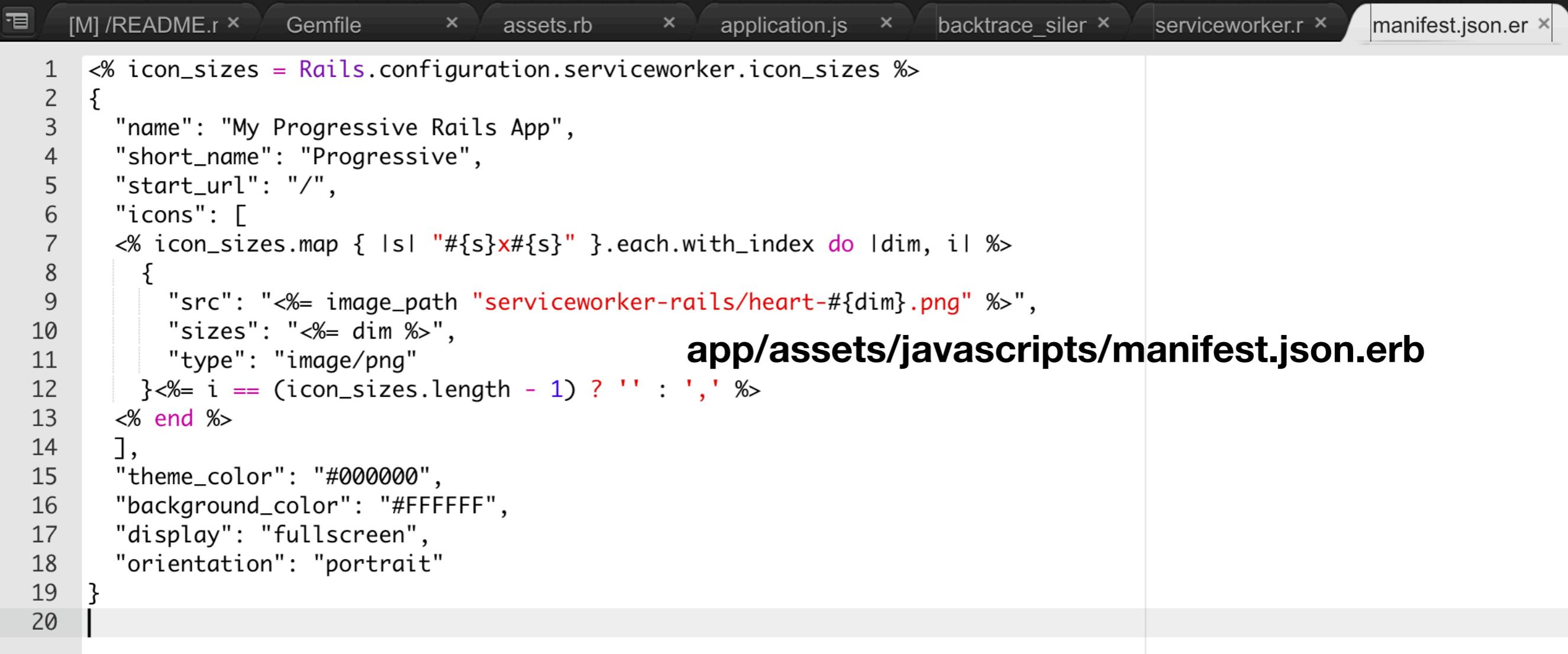
The screenshot shows a code editor with multiple tabs at the top: README.r, Gemfile, assets.rb, application.js, backtrace_siler, serviceworker.r, and manifest.json.er. The manifest.json.er tab is active, displaying the following Ruby code:

```
1 <% icon_sizes = Rails.configuration.serviceworker.icon_sizes %>
2 {
3   "name": "My Progressive Rails App",
4   "short_name": "Progressive",
5   "start_url": "/",
6   "icons": [
7     <% icon_sizes.map { |s| "##{s}x#{s}" }.each.with_index do |dim, i| %>
8       {
9         "src": "<%= image_path \"serviceworker-rails/heart-#{dim}.png\" %>",
10        "sizes": "<%= dim %>",
11        "type": "image/png"
12      }<%= i == (icon_sizes.length - 1) ? '' : ',' %>
13    <% end %>
14  ],
15   "theme_color": "#000000",
16   "background_color": "#FFFFFF",
17   "display": "fullscreen",
18   "orientation": "portrait"
19 }
20 |
```

app/assets/javascripts/manifest.json.erb

```
[M] /README.rx Gemfile x assets.rb x application.js x backtrace_siler x serviceworker.r x manifest.json.er x   
1 <% icon_sizes = Rails.configuration.serviceworker.icon_sizes %>  
2 {  
3   "name": "My Progressive Rails App",  
4   "short_name": "Progressive",  
5   "start_url": "/",  
6   "icons": [  
7     <% icon_sizes.map { |s| "##{s}x#{s}" }.each.with_index do |dim, i| %>  
8       {  
9         "src": "<%= image_path \"serviceworker-rails/heart-#{dim}.png\" %>",  
10         "sizes": "<%= dim %>",  
11         "type": "image/png"  
12       }<%= i == (icon_sizes.length - 1) ? '' : ',' %>  
13     <% end %>  
14   ],  
15   "theme_color": "#000000",  
16   "background_color": "#FFFFFF",  
17   "display": "fullscreen",  
18   "orientation": "portrait"  
19 }  
20 |
```

app/assets/javascripts/manifest.json.erb



The screenshot shows a code editor with multiple tabs at the top: README.r, Gemfile, assets.rb, application.js, backtrace_siler, serviceworker.r, and manifest.json.er. The manifest.json.er tab is active, displaying the following Ruby code:

```
1 <% icon_sizes = Rails.configuration.serviceworker.icon_sizes %>
2 {
3   "name": "My Progressive Rails App",
4   "short_name": "Progressive",
5   "start_url": "/",
6   "icons": [
7     <% icon_sizes.map { |s| "##{s}x#{s}" }.each.with_index do |dim, i| %>
8       {
9         "src": "<%= image_path \"serviceworker-rails/heart-#{dim}.png\" %>",
10        "sizes": "<%= dim %>",
11        "type": "image/png"
12      }<%= i == (icon_sizes.length - 1) ? '' : ',' %>
13    <% end %>
14  ],
15   "theme_color": "#000000",
16   "background_color": "#FFFFFF",
17   "display": "fullscreen",
18   "orientation": "portrait"
19 }
20 |
```

app/assets/javascripts/manifest.json.erb

manifest.json은 PWA로 구축된 웹앱의 프로필과 같은 역할을 한다고 보면 됨.

결국 저 `manifest.json.erb`는 최종적으로 아래와 같은 형태의 json 파일로 렌더링이 됨.

< > | N | 🔍 <https://tojung.me> <https://tojung.me/manifest.json>

C | M | ☆

```
{  
  "name": "Tojung",  
  "short_name": "Tojung",  
  "start_url": "/?utm_source=homescreen",  
  "icons": [  
    {  
      "src": "assets/logo-144-483eac25feeeadda7d95a4a587764e728ec9ceb72410c1ff784d40c0aca2a742.png",  
      "sizes": "144x144",  
      "type": "image/png"  
    }  
  ],  
  "theme_color": "#000000",  
  "background_color": "#FFFFFF",  
  "display": "fullscreen",  
  "orientation": "portrait"  
}
```

홈 화면에 저장할 로고는 뭘로 할건지, 풀스크린으로 사이트를 보여줄것인지, 배경색, 등등

The screenshot shows a code editor window with multiple tabs at the top, including 'Gemfile', 'assets.r', 'application.js', 'backtrace', 'service-worker.js', 'manifest.json', 'application.js', and 'service-worker.js'. The main editor area displays the following JavaScript code:

```
1 if (navigator.serviceWorker) {  
2   navigator.serviceWorker.register('/serviceworker.js', { scope: './' })  
3   .then(function(reg) {  
4     console.log('[Companion]', 'Service worker registered!');  
5   });  
6 }  
7  
app/assets/javascripts/serviceworker-companion.js
```

The code registers a service worker named 'serviceworker.js' under the current directory ('./'). It logs a message to the console when the registration is successful.

가장 첫번째 단계인 서비스워커를 페이지에 등록하는 단계이다.
serviceworker.js 파일을 실행하고 install 이벤트가 처음으로 실행된다.

app/assets/javascripts/serviceworker.js

```
62 self.addEventListener('install', onInstall);
63 self.addEventListener('activate', onActivate);
64 self.addEventListener('fetch', onFetch);

4 function onInstall(event) {
5   console.log('[Serviceworker]', "Installing!", event);
6   event.waitUntil(
7     caches.open(CACHE_NAME).then(function prefill(cache) {
8       return cache.addAll([
9         // make sure serviceworker.js is not required by application.js
10        // if you want to reference application.js from here
11        '<%= asset_path "application.js" %>',
12        '<%= asset_path "application.css" %>',
13        '/offline.html',
14      ]);
15    })
16  );
17 }
18
19
20
21 }
22
```

install 이벤트가 발생하여 **onInstall** 함수가 실행된다.
데이터를 캐싱(저장)하고나서 서비스워커가 설치된다.

app/assets/javascripts/serviceworker.js

```
62 self.addEventListener('install', onInstall);
63 self.addEventListener('activate', onActivate);
64 self.addEventListener('fetch', onFetch);

41 // Borrowed from https://github.com/TalAter/UpUp
42 function onFetch(event) {
43   event.respondWith(
44     // try to return untouched request from network first
45     fetch(event.request).catch(function() {
46       // if it fails, try to return request from the cache
47       return caches.match(event.request).then(function(response) {
48         if (response) {
49           return response;
50         }
51         // if not found in cache, return default offline content for navigate requests
52         if (event.request.mode === 'navigate' ||
53             (event.request.method === 'GET' && event.request.headers.get('accept').includes('text/html'))) {
54           console.log('[Serviceworker]', "Fetching offline content", event);
55           return caches.match('/offline.html');
56         }
57       })
58     })
59   );
60 }
```

서비스 워커를 설치된후에 사용자(브라우저)가 요청을 할려하면
서비스 워커가 onFetch함수가 실행됨.

app/assets/javascripts/serviceworker.js

```
62 self.addEventListener('install', onInstall);
63 self.addEventListener('activate', onActivate);
64 self.addEventListener('fetch', onFetch);

41 // Borrowed from https://github.com/TalAter/UpUp
42 function onFetch(event) {
43   event.respondWith(
44     // try to return untouched request from network first
45     fetch(event.request).catch(function() {
46       // if it fails, try to return request from the cache
47       return caches.match(event.request).then(function(response) {
48         if (response) {
49           return response;
50         }
51         // if not found in cache, return default offline content for navigate requests
52         if (event.request.mode === 'navigate' ||
53             (event.request.method === 'GET' && event.request.headers.get('accept').includes('text/html'))) {
54           console.log('[Serviceworker]', "Fetching offline content", event);
55           return caches.match('/offline.html');
56         }
57       })
58     })
59   );
60 }
```

이 onFetch코드는 오프라인 지원은 될지라도 기존 방식과 성능 차이가 크게 없을것으로 보임.
왜냐하면 이 코드는 이미 저장되어있는 응답값이 있더라도 서버에 먼저 요청을 하고,
요청이 실패한 경우에야 저장된 데이터를 탐색하는 방식이므로 성능이 기존 방식과 차이가 없음.

기존 **serviceworker.js** 파일 내용 전부 지우고 복붙

<https://raw.githubusercontent.com/limdongjin/rails4-pwa-boilerplate/go/app/assets/javascripts/serviceworker.js.erb>

0

```
// Borrowed from https://github.com/TalAter/UpUp
function onFetch(event) {
  console.log(event.request);
  if(event.request.method != 'GET'){
    console.log("Not GET");
    return;
  }
}
```

1-A

```
event.respondWith(
  caches.match(event.request)
  .then(function(response) {
    // Cache hit - return response
    if (response) {
      return response;
    }
  })
)
```

1-B

```
// IMPORTANT: Clone the request. A request
// can only be consumed once. Since we
// once by cache and once by the browser
// to clone the response.
var fetchRequest = event.request.clone();

return fetch(fetchRequest)
  .then(
    function(response) {
      // Check if we received a valid response
      if(!response || response.status !== 200 || response.type !== 'basic') {
        return response;
      }

      // IMPORTANT: Clone the response. A response is a stream
      // and because we want the browser to consume the response
      // as well as the cache consuming the response, we need
      // to clone it so we have two streams.
      var responseToCache = response.clone();

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

)
);

event.waitUntil(
  update(event.request)
  .then(refresh)
)
```

2

0. GET 메소드의 요청이 아니면 함수를 종료시킴

1. 요청에 대한 응답값이 저장되어있는지 확인

1-A. 저장되어있다면 저장되어있는 데이터로 응답함

1-B. 저장되어있지않다면, 서버에 요청을 보내서 응답값을 받아오고 (조건에 따라서)응답값을 캐시에 저장하고 응답값 반환

2. (사용자에게 응답값을 보내준후)서버에 요청을 보내서 데이터를 받아오고 브라우저에 저장된 데이터를 업데이트함.

fetch는 ajax요청을 보내는 함수라고 생각하면됨

<https://raw.githubusercontent.com/limdongjin/rails4-pwa-boilerplate/go/app/assets/javascripts/serviceworker.js.erb>

iPad Pro ▼ 1024 x 1366 100% Offline ▼

Elements Console Sources Network Application » 7 | : X

Listing Posts

Title	Content
aa	aa Show Edit Destroy
a	a Show Edit Destroy
s	s Show Edit Destroy
s	s Show Edit Destroy
ss	ss Show Edit Destroy
a	a Show Edit Destroy
sss	sss Show Edit Destroy
dasdasda	Show Edit Destroy
ss	ss Show Edit Destroy
ㄱㄴ고	Show Edit Destroy
ㅁㅁㅁㅁ	Show Edit Destroy
d	a Show Edit Destroy
d	a Show Edit Destroy
d	a Show Edit Destroy
ddd	a Show Edit Destroy
ddd	a Show Edit Destroy
aaa	ss Show Edit Destroy
aaa	Show Edit Destroy
ssssssssss	Show Edit Destroy
ssssssssss	Show Edit Destroy
aa	aa Show Edit Destroy
aa	aa Show Edit Destroy
andlaaskasn	ss Show Edit Destroy
saaslksanlsa	ssaas Show Edit Destroy
ssasadsa	ssasadsa Show Edit Destroy

[New Post](#)

Application

- Manifest
- Service Workers
- Clear storage

Storage

- Local Storage
- Session Storage
- IndexedDB
- Web SQL
- Cookies

Cache

- Cache Storage
 - v1:sw-cache- - https://pwa-gogc
 - Application Cache

Frames

- top

Path

Path	Content	Content	Time
assets/serviceworker.self-5ab5aa10c...	application	2,274	2018...
assets/turbolinks.self-1d1fdd91adc3...	application	35,295	2018...
manifest.json	application	2,350	2018...
offline.html	text/html	1,455	2018...
posts	text/html	9,620	2018...
posts/3	text/html	2,136	2018...
posts/new	text/html	2,694	2018...
serviceworker.js	application	3,664	2018...

Headers Preview

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Workspace</title>
5   <link rel="stylesheet" media="all" href="/assets/
6   <link rel="stylesheet" media="all" href="/assets/sc
7   <link rel="stylesheet" media="all" href="/assets/ap
8   <script src="/assets/jquery.self-bd7ddd393353a8d2
9   <script src="/assets/jquery_ujs.self-784a997f672603
10  <script src="/assets/turbolinks.self-1d1fdd91adc38
11  <script src="/assets/posts.self-877aeef30ae1b040ab8a
12  <script src="/assets/serviceworker-companion.self-f
13  <script src="/assets/serviceworker.self-988e056dc8c
14  <script src="/assets/application.self-f8806224e027f
15  <meta name="csrf-param" content="authenticity_tok
16  <meta name="csrf-token" content="7xwj8TAiAkTc2ey2QM
17  <link rel="manifest" href="/manifest.json" />
18  <meta name="apple-mobile-web-app-capable" content=""
19  <body>
20
21  <p id="notice"></p>
22
23  <h1>Listing Posts</h1>
24
25

```

Console

CHROME EXTENSION IS unsupported
at VM19 serviceworker.js:103

An unknown error occurred when fetching the script.
Failed to load resource: net::ERR_INTERNET_DISCONNECTED

한번이상 접속했던 페이지는 오프라인에서도 접속 가능. 물론 오프라인에서 글쓰기 같은거는 안먹힘.

서비스 워커로 데이터 캐싱하는거는
제대로 사용하면 정말 좋지만,
쉽지는 않다.

좀더 제대로 사용하고자 한다면 이 링크가 도움이 많이 될것.
<https://serviceworke.rs/strategy-cache-and-update.html>

그리고 서비스워커를 그냥 푸시알림정도만 구현하고 캐시는 별로 안쓸생각
으로 사용하고싶다면 ‘serviceworker-rails’젬이
처음에 생성해준 (onFetch)코드는 수정할일이 별로 없다.

좀더 제대로 사용하고자 한다면 이 링크가 도움이 많이 될것.
<https://serviceworke.rs/strategy-cache-and-update.html>

이제 푸시알림을 구현해보자

Create a new workspace

Workspace name

Description

Team

Choose a team for this workspace...

푸시알림도 아까 코드에서 확장되는 것이긴하지만,

타이핑해야할게 많아서 거의 완성된 프로젝트를 클론해서 시작한다.

[Hosted workspace](#)

[Clone workspace](#)

[Remote SSH workspace](#)

[Salesforce](#)



Private

This is a workspace for your eyes only

bundle install
rake db:migrate



Public

This will create a workspace for everybody to see

Clone from Git or Mercurial URL (optional)

Choose a template



HTML5



Node.js



PHP, Apache ...



Python



Django



Ruby

**bundle install
rake db:migrate**

```
ruby - "limdongjin-fin" x Immediate x Ruby on Rails - Run +  
limdongjin:~/workspace (master) $ rails c  
Running via Spring preloader in process 1251  
Loading development environment (Rails 4.2.5)  
2.3.4 :001 > require("webpush")  
=> false  
2.3.4 :002 > vapid_key = Webpush.generate_key  
=> #<Webpush::VapidKey:229c708 :public_key=BKj7t_NaUlEibmZ1LvP35YpBUjYYEMsT59mRvcnsCmVeT_oCewYB1UUvGWA...  
0aCaR8TeXCXw= :private_key=k06PNvi8NClwCQV26c1DA_GBC...=>  
2.3.4 :003 > puts vapid_key.public_key  
BKj7t_NaUlEibmZ1LvP35YpBUjYYEMsT59mRvcnsCmVeT_oCewYB1UUvGWA... 복사해서 메모장에 저장  
=> nil  
2.3.4 :004 > puts vapid_key.private_key  
k06PNvi8NClwCQV26c1DA_GBC... 복사해서 메모장에 저장  
=> nil  
2.3.4 :005 > □
```

Key 값은 각자 다르기 때문에 직접 콘솔에 아래 코드를 따라쳐서 나오는 값을 저장하자.

```
# rails console  
require("webpush")  
vapid_key = Webpush.generate_key  
puts vapid_key.public_key  
puts vapid_key.private_key
```

[P] /README.md x application.yml +

```
1 # Add configuration values here, as shown below.
2 #
3 # pusher_app_id: "2954"
4 # pusher_key: 7381a978f7dd7f9a1117
5 # pusher_secret: abdc3b896a0ffb85d373
6 # stripe_api_key: sk_test_2J0l093x0yW72XUYJHE4Dv2r
7 # stripe_publishable_key: pk_test_ro9jV5SNwGb1yYlQfzG17LHK
8 #
9 # production:
# VAPID_PUBLIC_KEY: 퍼블릭키값
# stripe_api_key: sk_live_FeHnL644j6z04Tyg41KdV9H
# VAPID_PRIVATE_KEY: 프라이빗키값
10 # stripe_publishable_key: pk_live_9LcthxpsIIBGwmd094101XVU
11
12 # VAPID_PUBLIC_KEY: B0Qbi4caR01XiT9kG005IN-fQk1rLGy2KSc-xWFQ7Pw5z74lpEq5-6FjNrKMXXMMHKbscqUebg0Yg-K_mxHTTyD0=
13 # VAPID_PRIVATE_KEY: hBpXfn6yc7QV-TwJ2j8XKFogMfb8GTyDDpx9CjQPI4Y=
14 VAPID_PUBLIC_KEY: BHL-J-56u4-rXHFGk-Ay4WorwL_KP85zevYwCFgf66wK70CrEwWQBaRGy4HLwmT94mWmWgXgb_J7SVci4En7EdI=
15 VAPID_PRIVATE_KEY: m5moupfDn9xmmFgf2veyx4RrQqFDLCpcJajSprq6kcY=
```

아까 메모장에 저장한 값을 아래와 같은 형식으로 config/application.yml에 넣어준다.

13:61 YAML Spaces: 4

```
ruby - "limdongjin-fin" x Immediate x Ruby on Rails - Run ⚡ + x
limdongjin:~/workspace (master) $ rails c
Running via Spring preloader in process 1251
Loading development environment (Rails 4.2.5)
2.3.4 :001 > require("webpush")
=> false
2.3.4 :002 > vapid_key = Webpush.generate_key
=> #<Webpush::VapidKey:229c708 :public_key=BKj7t_NaUlEibmZ1LvP35YpBUjYYEMsT59mRvcnsCmVeT_oCewYB1UUlvGWAmMdZvDgdyu7Fptz
0aCaR8TeXCXw= :private_key=k06PNvi8NClwCQV26c1DA_GBCtlqPVY1I2trIeQmHK8=>
2.3.4 :003 > puts vapid_key.public_key
BKj7t_NaUlEibmZ1LvP35YpBUjYYEMsT59mRvcnsCmVeT_oCewYB1UUlvGWAmMdZvDgdyu7Fptz0aCaR8TeXCXw=
=> nil
2.3.4 :004 > puts vapid_key.private_key
k06PNvi8NClwCQV26c1DA_GBCtlqPVY1I2trIeQmHK8=
=> nil
2.3.4 :005 > 
```

[P] /README.md x application.yml +

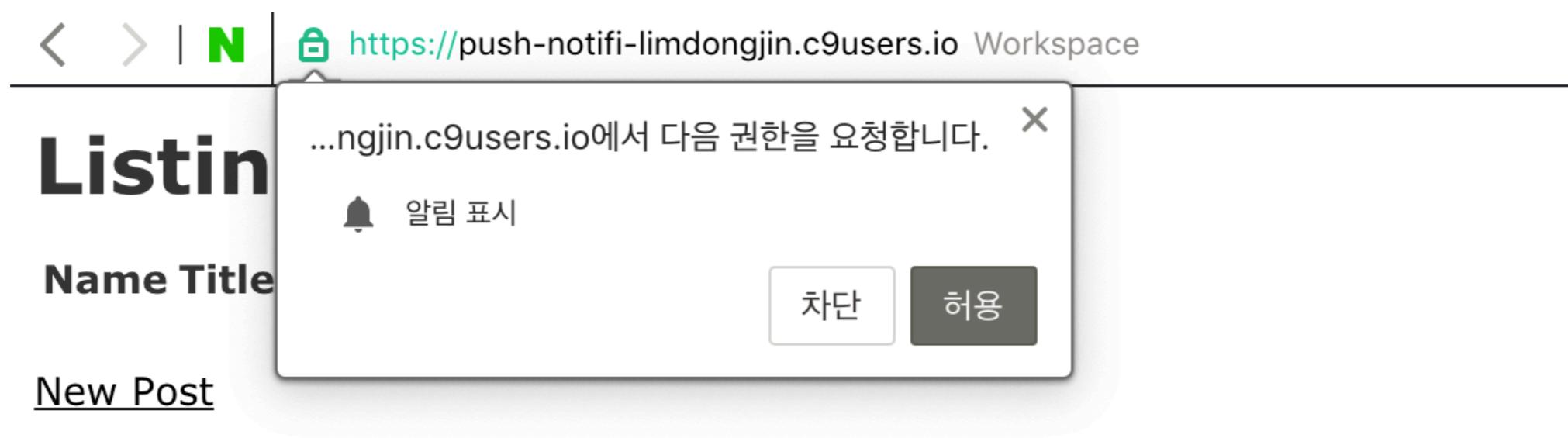
```
1 # Add configuration values here, as shown below.
2 #
3 # pusher_app_id: "2954"
4 # pusher_key: 7381a978f7dd7f9a1117
5 # pusher_secret: abdc3b896a0ffb85d373
6 # stripe_api_key: sk_test_2J0l093x0yW72XUYJHE4Dv2r
7 # stripe_publishable_key: pk_test_ro9jV5SNwGb1yYlQfzG17LHK
8 #
9 # VAPID_PUBLIC_KEY: 퍼블릭키값
10 # stripe_api_key: sk_live_FeHnL644j6zo4Tyg41KdV9H
11 # stripe_publishable_key: pk_live_9LcthxpS1BGwmd094101XVU
12
13 # VAPID_PUBLIC_KEY: B0Qbi4caR01XiT9kG005IN-fQk1rLGy2KSc-xWFQ7Pw5z74lpEq5-6FjNrKMXXMMHKbscqUebg0Yg-K_mxHTTyD0=
14 # VAPID_PRIVATE_KEY: hBpXfn6yc7QV-TwJ2j8XKFogMfb8GTyDDpx9CjQPI4Y=
15 VAPID_PUBLIC_KEY: BHL-J-56u4-rXHFGk-Ay4WorwL_KP85zevYwCFgf66wK70CrEwWQBaRGy4HLwmT94mWmWgXgb_J7SVci4En7EdI=
16 VAPID_PRIVATE_KEY: m5moupfDn9xmmFgf2veyx4RrQqFDLCpcJajSprq6kcY=
```

아까 메모장에 저장한 값을 아래와 같은 형식으로 config/application.yml에 넣어준다.

13:61 YAML Spaces: 4

주의점: 실제 서비스로 배포하는 경우에는 config/application.yml 을 .gitignore에 추가하여 깃허브에 푸시되지 않게 해야한다.

서버 실행(Run)
/posts 로 접속



허용 클릭

The screenshot shows the Cloud9 IDE interface with several tabs open at the top:

- push-notifi - Cloud9
- Workspace
- finfinf - Cloud9
- 내 삶을 바꾸는 입법안 쇼핑몰
- Progressive Web App (fea

The main workspace shows the file structure of the 'push-notifi' project. The 'application.yml' file is open, displaying configuration values:

```
1 # Add configuration values here, as shown below.
2 #
3 # pusher_app_id: "2954"
4 # pusher_key: 7381a978f7dd7f9a1117
5 # pusher_secret: abdc3b896a0ffb85d373
6 # stripe_api_key: sk_test_2J0l093x0yW72XUYJHE4Dv2r
7 # stripe_publishable_key: pk_test_ro9jV5SNwGb1yYlQfzG17LHK
8 #
9 # production:
10 #   stripe_api_key: sk_live_EeHnL644i6zo4Iyq4v1KdV9H
11 #   stripe_publishable_key: pk_live_9lcthxpSIHbGwmd094101XVU
12
13 # VAPID_PUBLIC_KEY: B0Qbi4caR01XiT9kG005IN-fQk1rLGy2KSc-xWFQ7Pw5z74lpEq5-6FjNrKMXXMHKbscqUebg0Yg-K_mxHTTyD0=
14 # VAPID_PRIVATE_KEY: hBpXfn6yc7QV-TwJ2j8XKFogMfb8GTyDDpx9CjQPI4Y=
15
16 VAPID_PUBLIC_KEY: BL5d3yDKMb640n0vgWSnFjNSg3ZEJLvPI8IXux_AFUXpN3d0Tm4jUyH0KLWPiyksqBsvZ0VJAFwE0-zMH0xNvwI=
17 VAPID_PRIVATE_KEY: JPg0S04Hky0jDM1SspZS2BUkgVyx_VxVxIdq9jqWqCw=
```

The bottom pane shows a terminal window with the following output:

```
JPg0S04Hky0jDM1SspZS2BUkgVyx_VxVxIdq9jqWqCw=
=> nil
2.3.4 :005 > exit
limdongjin:~/workspace (master) $ clear
limdongjin:~/workspace (master) $ rake my_tasks:noti
Go!!!!!!1
*****
BL5d3yDKMb640n0vgWSnFjNSg3ZEJLvPI8IXux_AFUXpN3d0Tm4jUyH0KLWPiyksqBsvZ0VJAFwE0-zMH0xNvwI=
JPg0S04Hky0jDM1SspZS2BUkgVyx_VxVxIdq9jqWqCw=
%%%%%%%
limdongjin:~/workspace (master) $ rake my_tasks:noti
Go!!!!!!1
*****
BL5d3yDKMb640n0vgWSnFjNSg3ZEJLvPI8IXux_AFUXpN3d0Tm4jUyH0KLWPiyksqBsvZ0VJAFwE0-zMH0xNvwI=
JPg0S04Hky0jDM1SspZS2BUkgVyx_VxVxIdq9jqWqCw=
%%%%%%%
limdongjin:~/workspace (master) $
```

\$ rake my_tasks:noti

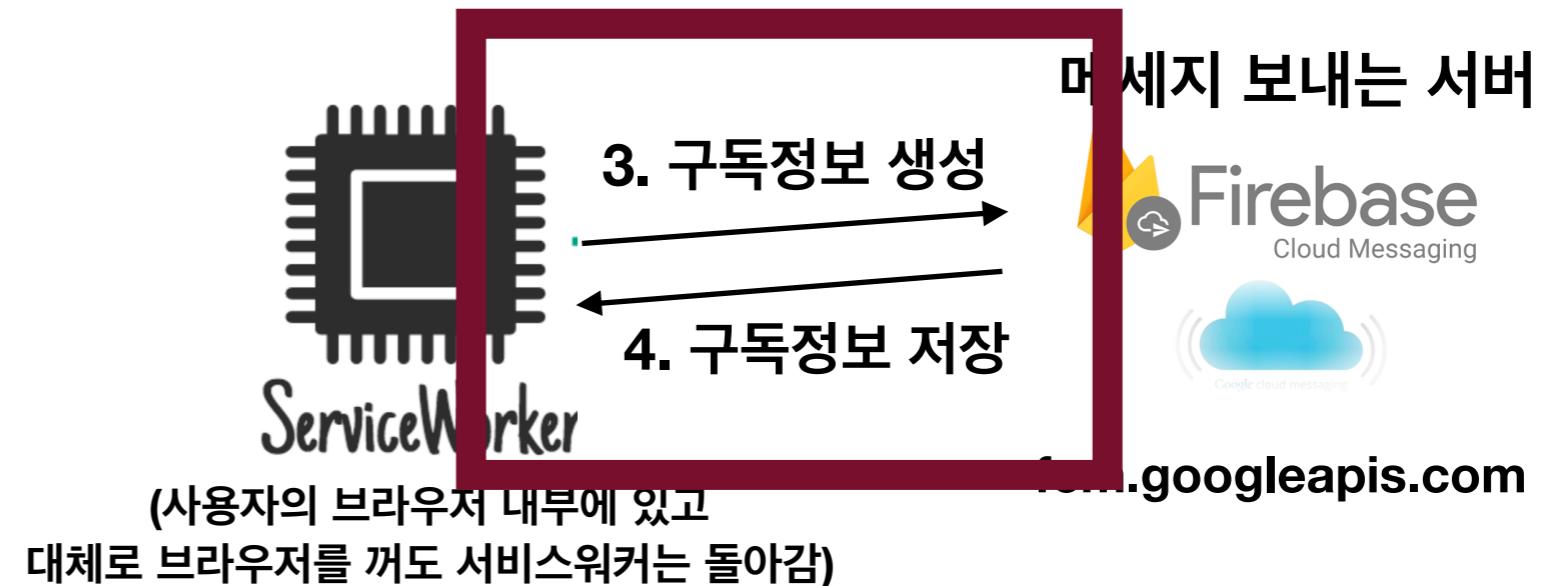
The screenshot shows a terminal window titled "ruby - 'limdongjin-fin'" with the tab "Immediate" selected. The command \$ rails c is run, followed by require("webpush"). The code then attempts to generate a vapid key, which fails (=> false). It then tries to print the public key of the generated key object, which also fails (=> nil). The private key is similarly printed and fails (=> nil). The configuration file contains comments for Pusher and Stripe keys, and then defines VAPID keys. The VAPID PUBLIC KEY is set to B0Qbi... and the VAPID PRIVATE KEY is set to hBpXfn6yc7QV-TwJ2j8XKFogMfb8GTYDDpx9CjQPI4Y=.

```
limdongjin:~/workspace (master) $ rails c
Running via Spring preloader in process 1251
Loading development environment (Rails 4.2.5)
2.3.4 :001 > require("webpush")
=> false
2.3.4 :002 > vapid_key = Webpush.generate_key
=> #<Webpush::VapidKey:229c708 :public_key=BKj7t_NaUlEibmZ1LvP35YpBUjYYEMsT59mRvcnsCmVeT_oCewYB1UUlvGWAmMdZvDgdyu7Fptz
0aCaR8TeXCXw= :private_key=k06PNvi8NClwCQV26c1DA_GBClqPVY1I2trIeQmHK8=>
2.3.4 :003 > puts vapid_key.public_key
BKj7t_NaUlEibmZ1LvP35YpBUjYYEMsT59mRvcnsCmVeT_oCewYB1UUlvGWAmMdZvDgdyu7Fptz0aCaR8TeXCXw=
=> nil
2.3.4 :004 > puts vapid_key.private_key
k06PNvi8NClwCQV26c1DA_GBClqPVY1I2trIeQmHK8=
=> nil
2.3.4 :005 > 
13:61   YAML   Spaces: 4
```

Webpush라는 API를 이용하면 구독정보 생성 및 메세지를 보내는 과정을 쉽게 구현할수있다. 이 API를 이용하기위해서는 위와같은 키값이 필요하다.

구독하는 프로세스

Webpush 젬이 구독정보 생성을 위한 키값을 만들어줌



사용자의 컴퓨터/폰



1. 사용자가 사이트 접속



2. 구독 버튼 클릭

Page

웹서버

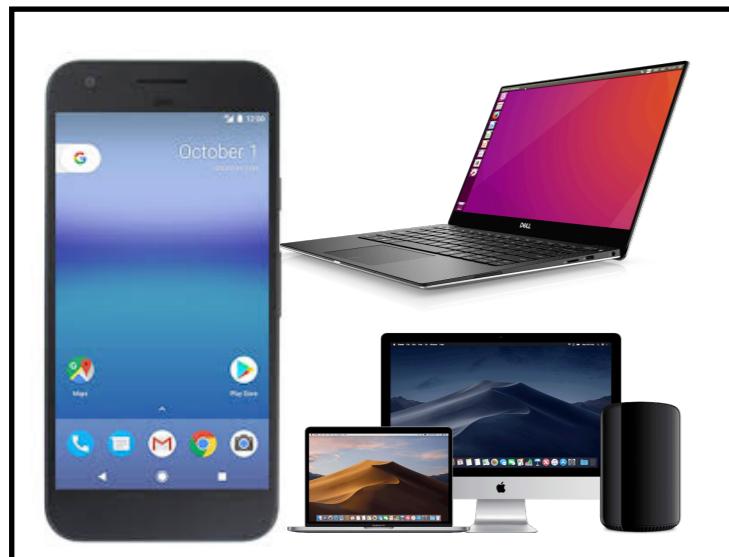


구독되어있는 상태에서 메세지 받는 프로세스



Webpush 젬이 이 과정의 구현을 도와줌

사용자의 컴퓨터/폰



Page



1. 새로운 게시글이 생성됨

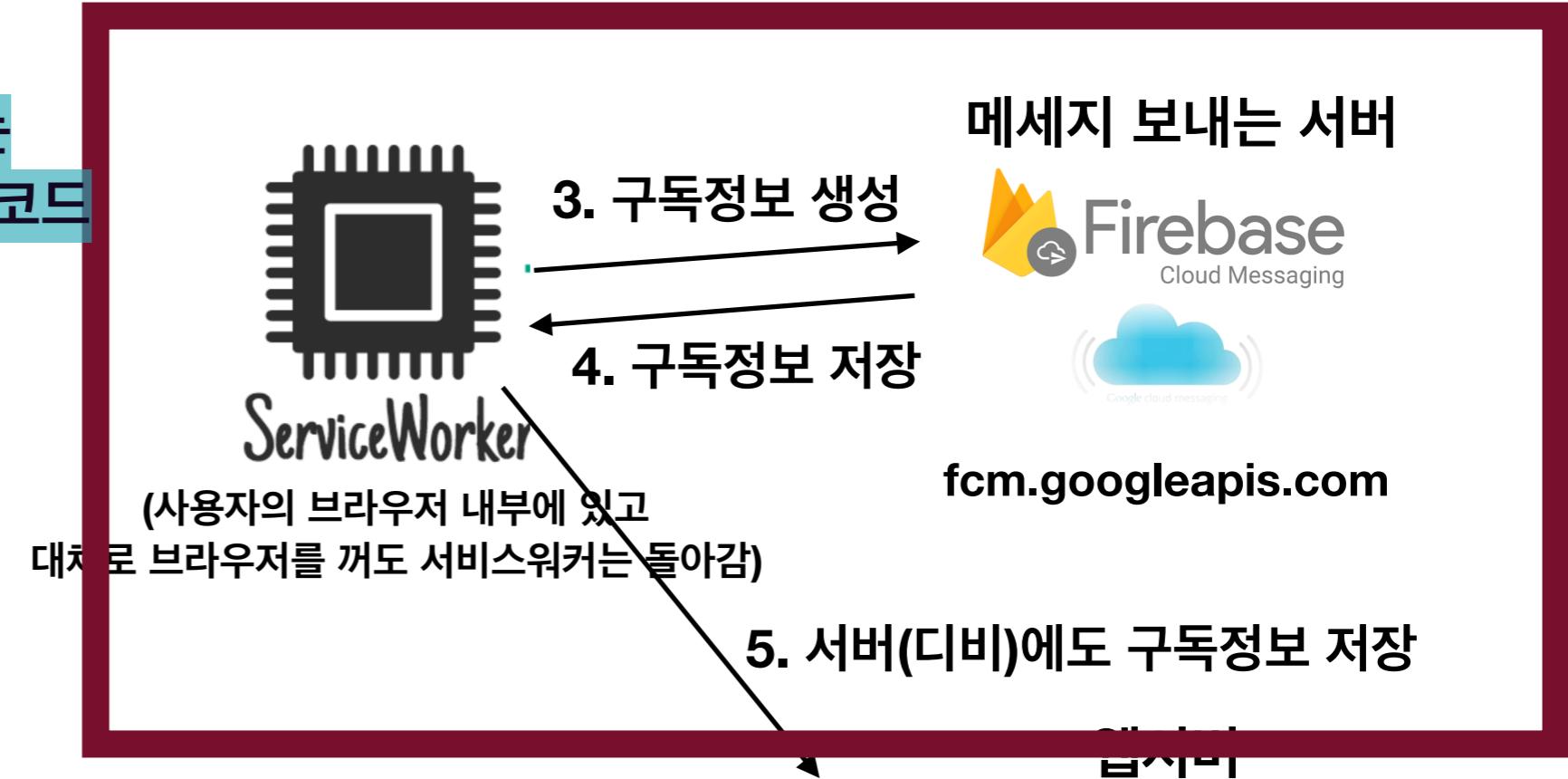
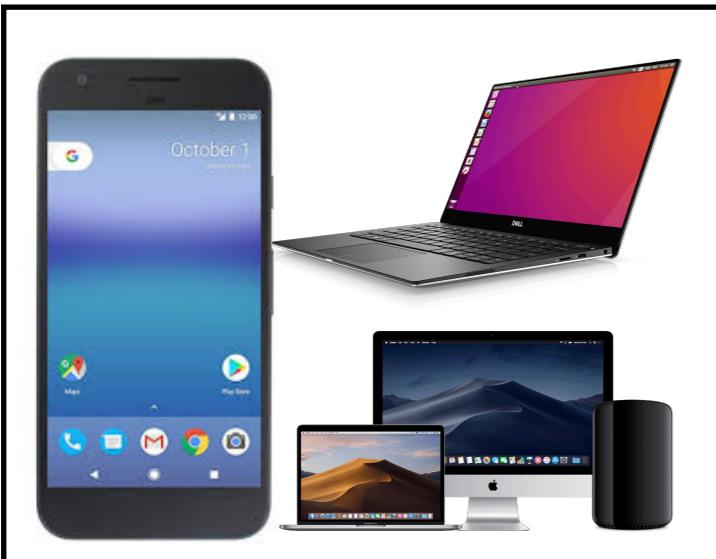
app/assets/javascripts/serviceworker-companion.js.erb 둘러보기

```
7  navigator.serviceWorker.ready
8    .then((swReg) => {
9      console.log("When SW is ready");
10     swReg.pushManager.getSubscription()
11       .then((subscription) => {
12         isSubscribed = !(subscription === null);
13
14         if (isSubscribed) {
15           // 이미 구독을 하고 있는 경우!
16           console.log('User IS subscribed.');
17           console.log(subscription)
18         } else {
19           // 아직 구독을 안하고 있는 경우!
20           // 그렇기 때문에 구독을 생성해줘야 한다.
21           console.log('User is NOT subscribed.');
22           return swReg.pushManager
23             .subscribe({
24               // 구독
25               userVisibleOnly: true,
26               applicationServerKey: window.vapidPublicKey
27             }).then(() => (swReg))
28             .then((swReg) => {
29               swReg.pushManager.getSubscription()
30                 .then((sub) => {
31                   // 서버에도 구독 정보를 저장해야 하므로
32                   let body = {
33                     subscription: sub.toJSON(),
34                     message: 'Hey!'
35                   };
36                   console.log(body);
37                   fetch('/push', {
38                     method: 'POST',
39                     body: JSON.stringify(body)
40                   })
41                 })
42               })
43             })
44           })
45         })
46       )
47     )
48   )
49   )
50   )
51   )
52   )
53   )
54   )
55   )
56   )
57   )
58   )
59   )
60   )
61   )
62   )
63   )
64   )
65   )
66   )
67   )
68   )
69   )
70   )
71   )
72   )
73   )
74   )
75   )
76   )
77   )
78   )
79   )
80   )
81   )
82   )
83   )
84   )
85   )
86   )
87   )
88   )
89   )
90   )
91   )
92   )
93   )
94   )
95   )
96   )
97   )
98   )
99   )
100  )
101  )
102  )
103  )
104  )
105  )
106  )
107  )
108  )
109  )
110  )
111  )
112  )
113  )
114  )
115  )
116  )
117  )
118  )
119  )
120  )
121  )
122  )
123  )
124  )
125  )
126  )
127  )
128  )
129  )
130  )
131  )
132  )
133  )
134  )
135  )
136  )
137  )
138  )
139  )
140  )
141  )
142  )
143  )
144  )
145  )
146  )
147  )
148  )
149  )
150  )
151  )
152  )
153  )
154  )
155  )
156  )
157  )
158  )
159  )
160  )
161  )
162  )
163  )
164  )
165  )
166  )
167  )
168  )
169  )
170  )
171  )
172  )
173  )
174  )
175  )
176  )
177  )
178  )
179  )
180  )
181  )
182  )
183  )
184  )
185  )
186  )
187  )
188  )
189  )
190  )
191  )
192  )
193  )
194  )
195  )
196  )
197  )
198  )
199  )
200  )
201  )
202  )
203  )
204  )
205  )
206  )
207  )
208  )
209  )
210  )
211  )
212  )
213  )
214  )
215  )
216  )
217  )
218  )
219  )
220  )
221  )
222  )
223  )
224  )
225  )
226  )
227  )
228  )
229  )
230  )
231  )
232  )
233  )
234  )
235  )
236  )
237  )
238  )
239  )
240  )
241  )
242  )
243  )
244  )
245  )
246  )
247  )
248  )
249  )
250  )
251  )
252  )
253  )
254  )
255  )
256  )
257  )
258  )
259  )
259  )
260  )
261  )
262  )
263  )
264  )
265  )
266  )
267  )
268  )
269  )
269  )
270  )
271  )
272  )
273  )
274  )
275  )
276  )
277  )
278  )
279  )
279  )
280  )
281  )
282  )
283  )
284  )
285  )
286  )
287  )
287  )
288  )
289  )
289  )
290  )
291  )
292  )
293  )
294  )
295  )
296  )
297  )
298  )
299  )
299  )
300  )
301  )
302  )
303  )
304  )
305  )
306  )
307  )
308  )
309  )
309  )
310  )
311  )
312  )
313  )
314  )
315  )
316  )
317  )
318  )
319  )
319  )
320  )
321  )
322  )
323  )
324  )
325  )
326  )
327  )
328  )
329  )
329  )
330  )
331  )
332  )
333  )
334  )
335  )
336  )
337  )
338  )
339  )
339  )
340  )
341  )
342  )
343  )
344  )
345  )
346  )
347  )
348  )
349  )
349  )
350  )
351  )
352  )
353  )
354  )
355  )
356  )
357  )
358  )
359  )
359  )
360  )
361  )
362  )
363  )
364  )
365  )
366  )
367  )
368  )
369  )
369  )
370  )
371  )
372  )
373  )
374  )
375  )
376  )
377  )
378  )
379  )
379  )
380  )
381  )
382  )
383  )
384  )
385  )
386  )
387  )
388  )
389  )
389  )
390  )
391  )
392  )
393  )
394  )
395  )
396  )
397  )
398  )
399  )
399  )
400  )
401  )
402  )
403  )
404  )
405  )
406  )
407  )
408  )
409  )
409  )
410  )
411  )
412  )
413  )
414  )
415  )
416  )
417  )
418  )
419  )
419  )
420  )
421  )
422  )
423  )
424  )
425  )
426  )
427  )
428  )
429  )
429  )
430  )
431  )
432  )
433  )
434  )
435  )
436  )
437  )
438  )
439  )
439  )
440  )
441  )
442  )
443  )
444  )
445  )
446  )
447  )
448  )
449  )
449  )
450  )
451  )
452  )
453  )
454  )
455  )
456  )
457  )
458  )
459  )
459  )
460  )
461  )
462  )
463  )
464  )
465  )
466  )
467  )
468  )
469  )
469  )
470  )
471  )
472  )
473  )
474  )
475  )
476  )
477  )
478  )
479  )
479  )
480  )
481  )
482  )
483  )
484  )
485  )
486  )
487  )
488  )
489  )
489  )
490  )
491  )
492  )
493  )
494  )
495  )
496  )
497  )
498  )
499  )
499  )
500  )
501  )
502  )
503  )
504  )
505  )
506  )
507  )
508  )
509  )
509  )
510  )
511  )
512  )
513  )
514  )
515  )
516  )
517  )
518  )
519  )
519  )
520  )
521  )
522  )
523  )
524  )
525  )
526  )
527  )
528  )
529  )
529  )
530  )
531  )
532  )
533  )
534  )
535  )
536  )
537  )
538  )
539  )
539  )
540  )
541  )
542  )
543  )
544  )
545  )
546  )
547  )
548  )
549  )
549  )
550  )
551  )
552  )
553  )
554  )
555  )
556  )
557  )
558  )
559  )
559  )
560  )
561  )
562  )
563  )
564  )
565  )
566  )
567  )
568  )
569  )
569  )
570  )
571  )
572  )
573  )
574  )
575  )
576  )
577  )
578  )
579  )
579  )
580  )
581  )
582  )
583  )
584  )
585  )
586  )
587  )
588  )
589  )
589  )
590  )
591  )
592  )
593  )
594  )
595  )
596  )
597  )
598  )
599  )
599  )
600  )
601  )
602  )
603  )
604  )
605  )
606  )
607  )
608  )
609  )
609  )
610  )
611  )
612  )
613  )
614  )
615  )
616  )
617  )
618  )
619  )
619  )
620  )
621  )
622  )
623  )
624  )
625  )
626  )
627  )
628  )
629  )
629  )
630  )
631  )
632  )
633  )
634  )
635  )
636  )
637  )
638  )
639  )
639  )
640  )
641  )
642  )
643  )
644  )
645  )
646  )
647  )
648  )
649  )
649  )
650  )
651  )
652  )
653  )
654  )
655  )
656  )
657  )
658  )
659  )
659  )
660  )
661  )
662  )
663  )
664  )
665  )
666  )
667  )
668  )
669  )
669  )
670  )
671  )
672  )
673  )
674  )
675  )
676  )
677  )
678  )
679  )
679  )
680  )
681  )
682  )
683  )
684  )
685  )
686  )
687  )
688  )
689  )
689  )
690  )
691  )
692  )
693  )
694  )
695  )
696  )
697  )
698  )
698  )
699  )
700  )
701  )
702  )
703  )
704  )
705  )
706  )
707  )
708  )
709  )
709  )
710  )
711  )
712  )
713  )
714  )
715  )
716  )
717  )
718  )
719  )
719  )
720  )
721  )
722  )
723  )
724  )
725  )
726  )
727  )
728  )
729  )
729  )
730  )
731  )
732  )
733  )
734  )
735  )
736  )
737  )
738  )
739  )
739  )
740  )
741  )
742  )
743  )
744  )
745  )
746  )
747  )
748  )
749  )
749  )
750  )
751  )
752  )
753  )
754  )
755  )
756  )
757  )
758  )
759  )
759  )
760  )
761  )
762  )
763  )
764  )
765  )
766  )
767  )
768  )
769  )
769  )
770  )
771  )
772  )
773  )
774  )
775  )
776  )
777  )
778  )
779  )
779  )
780  )
781  )
782  )
783  )
784  )
785  )
786  )
787  )
788  )
789  )
789  )
790  )
791  )
792  )
793  )
794  )
795  )
796  )
797  )
798  )
798  )
799  )
800  )
801  )
802  )
803  )
804  )
805  )
806  )
807  )
808  )
809  )
809  )
810  )
811  )
812  )
813  )
814  )
815  )
816  )
817  )
818  )
819  )
819  )
820  )
821  )
822  )
823  )
824  )
825  )
826  )
827  )
828  )
829  )
829  )
830  )
831  )
832  )
833  )
834  )
835  )
836  )
837  )
838  )
839  )
839  )
840  )
841  )
842  )
843  )
844  )
845  )
846  )
847  )
848  )
849  )
849  )
850  )
851  )
852  )
853  )
854  )
855  )
856  )
857  )
858  )
859  )
859  )
860  )
861  )
862  )
863  )
864  )
865  )
866  )
867  )
868  )
869  )
869  )
870  )
871  )
872  )
873  )
874  )
875  )
876  )
877  )
878  )
879  )
879  )
880  )
881  )
882  )
883  )
884  )
885  )
886  )
887  )
888  )
889  )
889  )
890  )
891  )
892  )
893  )
894  )
895  )
896  )
897  )
898  )
898  )
899  )
900  )
901  )
902  )
903  )
904  )
905  )
906  )
907  )
908  )
909  )
909  )
910  )
911  )
912  )
913  )
914  )
915  )
916  )
917  )
918  )
919  )
919  )
920  )
921  )
922  )
923  )
924  )
925  )
926  )
927  )
928  )
929  )
929  )
930  )
931  )
932  )
933  )
934  )
935  )
936  )
937  )
938  )
939  )
939  )
940  )
941  )
942  )
943  )
944  )
945  )
946  )
947  )
948  )
949  )
949  )
950  )
951  )
952  )
953  )
954  )
955  )
956  )
957  )
958  )
959  )
959  )
960  )
961  )
962  )
963  )
964  )
965  )
966  )
967  )
968  )
969  )
969  )
970  )
971  )
972  )
973  )
974  )
975  )
976  )
977  )
978  )
979  )
979  )
980  )
981  )
982  )
983  )
984  )
985  )
986  )
987  )
988  )
988  )
989  )
990  )
991  )
992  )
993  )
994  )
995  )
996  )
997  )
998  )
999  )
999  )
1000  )
```

구독하는 프로세스

전 슬라이드의 코드는
3, 4, 5로직을 구현하는 코드

사용자의 컴퓨터/폰



Page

app/assets/javascripts/serviceworker-companion.js.erb 둘러보기

```
⚠ 7  navigator.serviceWorker.ready  
  8  .then((swReg) => {  
  9    console.log("When SW is ready");  
 10   swReg.pushManager.getSubscription()  
 11     .then((subscription) => {  
 12       isSubscribed = !(subscription === null);  
 13     });  
 14   });  
 15 }  
 16  
 17 // https://github.com/GoogleChrome/samples/tree/gh-pages/service-worker/companion-script-example
```

subscription 에는 현재의 구독정보가 담기게 된다.

만일 구독을 하고있지않다면 *null*값이고

구독을 하고있다면 구독정보값을 갖고있다.

```
PushSubscription {endpoint: "https://fcm.googleapis.com/fcm/send/dn7R18MTlju:AP...gw  
▼ Dfuv7BdT7FJuuBQf0-leaqBRrzkWlLd0x800IGKATc1MXow", expirationTime: null, options: PushSubscriptionOptions} ⓘ  
  endpoint: "https://fcm.googleapis.com/fcm/send/dn7R18MTlju:APA91bE40Waai_0kxzxcW11u  
  expirationTime: null  
▼ options: PushSubscriptionOptions  
  ► applicationServerKey: ArrayBuffer(65) {}  
    userVisibleOnly: true  
  ► __proto__: PushSubscriptionOptions  
▼ __proto__: PushSubscription  
  endpoint: (...)  
  expirationTime: (...)  
  ► getKey: f getKey()  
    options: (...)  
  ► toJSON: f toJSON()  
  ► unsubscribe: f unsubscribe()  
  ► constructor: f PushSubscription()  
  Symbol(Symbol.toStringTag): "PushSubscription"
```

subscribe 메소드를 통해서 구독정보를 생성하고
서비스워커에 저장하는 과정을 진행할수있다.

app/assets/javascripts/serviceworker-companion.js.erb

```
20 // 아직 구독을 안하고 있는 경우!
21 // 그렇기 때문에 구독을 생성해줘야 한다.
22 console.log('User is NOT subscribed.');
23 return swReg.pushManager
24     .subscribe({
25         // 구독
26         userVisibleOnly: true,
27         applicationServerKey: window.vapidPublicKey
28     }).then(() => (swReg))
```

app/assets/javascripts/vapid.js.erb

```
1 <% @key = Base64.urlsafe_decode64(ENV['VAPID_PUBLIC_KEY']).bytes %>
2 window.vapidPublicKey = new Uint8Array(<%= @key %>)
```

방금 만든 키값을 정제해서 자바스크립트 변수에 저장한다.

app/assets/javascripts/serviceworker-companion.js.erb

```
28      }).then(() => (swReg))
29      .then((swReg) => {
30          swReg.pushManager.getSubscription()
31          .then((sub) => {
32              // 서버에도 구독 정보를 저장해야 하므로
33              let body = {
34                  subscription: sub.toJSON(),
35                  message: 'Hey!'서버에 구독정보를 저장하는 요청을 보낸다.
36              };
37
38      fetch('/push', {
39          method: 'POST',
40          body: JSON.stringify(body)
41      }
42      .then((res) => {
43          console.log(res)
44      })
45      .catch((err) => {
46          console.log(err)
47      })
48
49
50 })
```

```
    .then((res) => {
        console.log(res)
    })
    .catch((err) => {
        console.log(err)
    })
})
```

```
▼ {endpoint: "https://fcm.googleapis.com/fcm/send/dn7R18MTljU:AP...gWDfuv7BdT7FJuuBQf0
-leaqBRrzkWlLd0x800IGKATc1MXow", expirationTime: null, keys: {...}} ⓘ
  endpoint: "https://fcm.googleapis.com/fcm/send/dn7R18MTljU:APA91bE40Waai_0kxzxcW11uh0
  expirationTime: null
  ▼ keys:
    auth: "DHJcCh9qJjF7JIW27KDCmg=="
    p256dh: "BCADrFZMfm5kKHQyXYjvxaTB8gKIjH1UJvske_Rr-v861YBQhdN4RHJPQenxVYPaPvtKHaYPMn
  ▼ proto :
```

sub.toJSON()가 구독정보이다.

app/assets/javascripts/serviceworker-companion.js.erb

```
28      }).then(() => (swReg))
29      .then((swReg) => {
30          swReg.pushManager.getSubscription()
31          .then((sub) => {
32              // 서버에도 구독 정보를 저장해야 하므로
33              let body = {
34                  subscription: sub.toJSON(),
35                  message: 'Hey!'서버에 구독정보를 저장하는 요청을 보낸다.
36              };
37
38      })
39
40
41
42
43
44
45
46
47
48
```

```
        fetch('/push', {
            method: 'POST',
            body: JSON.stringify(body)
        })
        .then((res) => {
            console.log(res)
        })
        .catch((err) => {
            console.log(err)
        })
    
```

```
1 class NotificationsController < ApplicationController
2   skip_before_action :verify_authenticity_token
3
4   def push POST /push
5     jsonbody = JSON.parse request.body.read()
6     endpoint = jsonbody["subscription"]["endpoint"]
7     p256dh = jsonbody["subscription"]["keys"]["p256dh"]
8     auth = jsonbody["subscription"]["keys"]["auth"]
9     @notification = Notification.new(endpoint: endpoint, p256dh: p256dh, auth: auth)
10    @notification.ip = request.remote_ip
11    # 필요에 따라 user의 id도 저장할 수 있음.
12
13    @notification.save()
14
15    render json: jsonbody
16  end
```

app/assets/javascripts/serviceworker-companion.js.erb

```
28      }).then(() => (swReg))
29      .then((swReg) => {
30          swReg.pushManager.getSubscription()
31          .then((sub) => {
32              // 서버에도 구독 정보를 저장해야 하므로
33              let body = {
34                  subscription: sub.toJSON(),
35                  message: 'Hey!'
36              };
37
38      fetch('/push', {
39          method: 'POST',
40          body: JSON.stringify(body)
41      }
42      .then((res) => {
43          console.log(res)
44      })
45      .catch((err) => {
46          console.log(err)
47      })
48  
```

서버에 구독정보를 저장하는 요청을 보낸다.

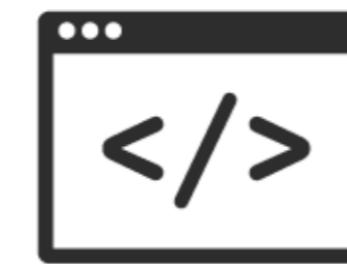
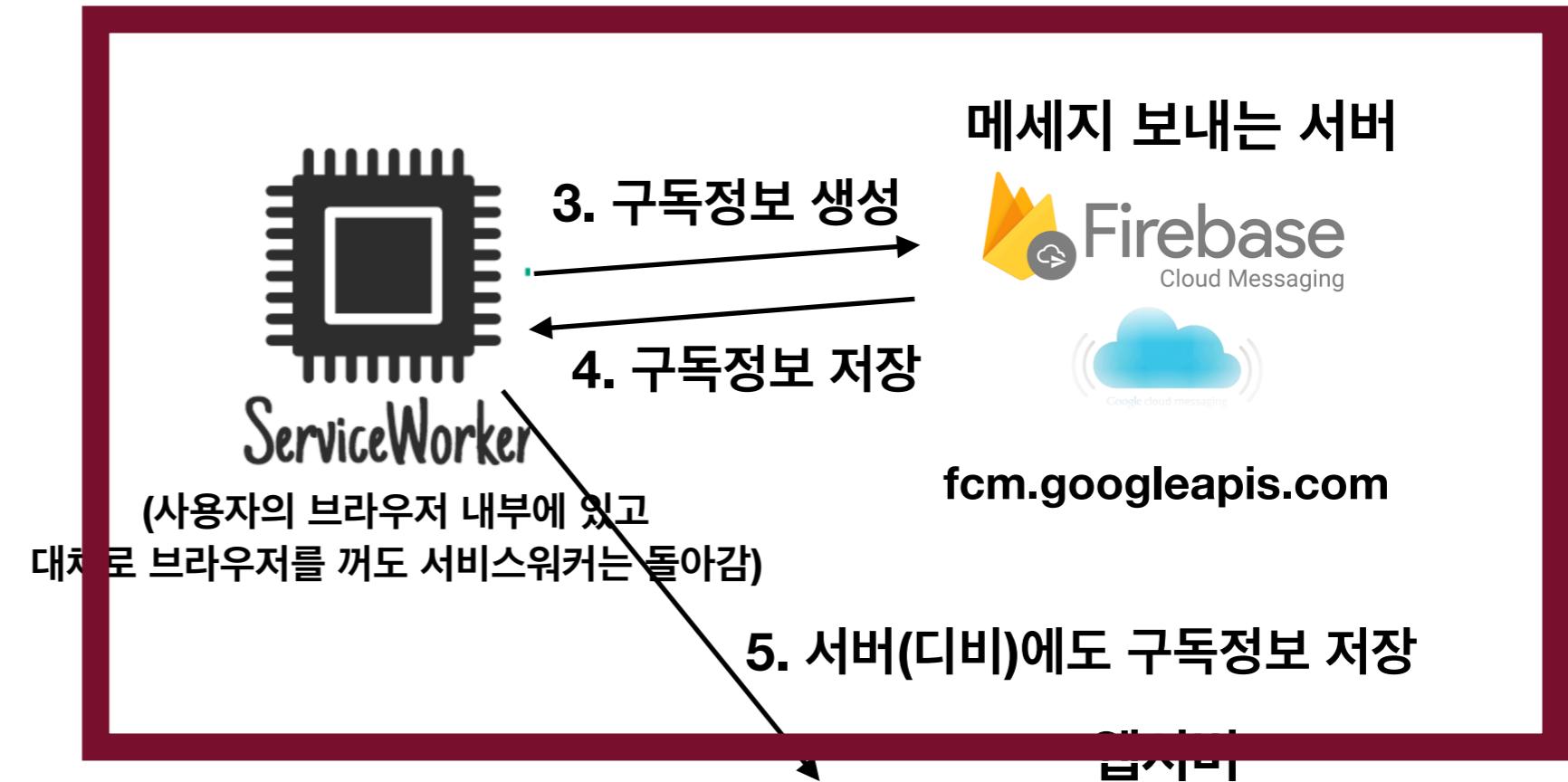
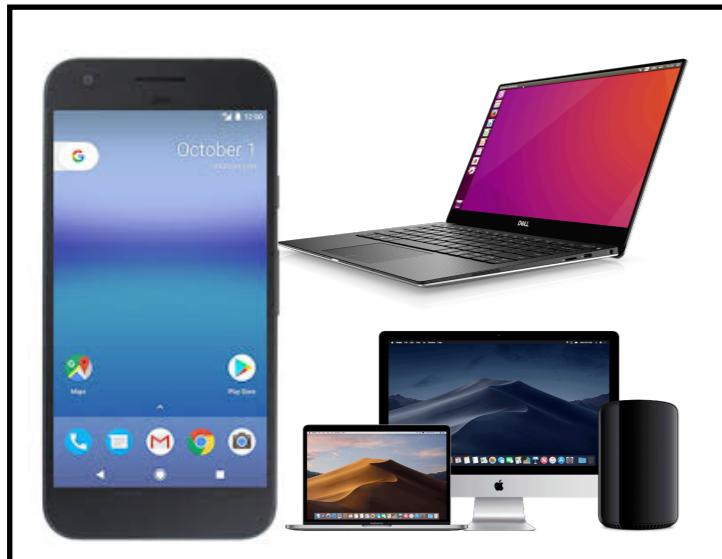
```
1 class NotificationsController < ApplicationController
2   skip_before_action :verify_authenticity_token
3
4   def push
5     jsonbody = JSON.parse request.body.read()
6     endpoint = jsonbody["subscription"]["endpoint"]
7     p256dh = jsonbody["subscription"]["keys"]["p256dh"]
8     auth = jsonbody["subscription"]["keys"]["auth"]
9     @notification = Notification.new(endpoint: endpoint, p256dh: p256dh, auth: auth)
10    @notification.ip = request.remote_ip
11    # 필요에 따라 user의 id도 저장할 수 있음.
12
13    @notification.save()
14
15    render json: jsonbody
16  end

```

Notification 디비에 구독정보를 저장함.

구독하는 프로세스

사용자의 컴퓨터/폰

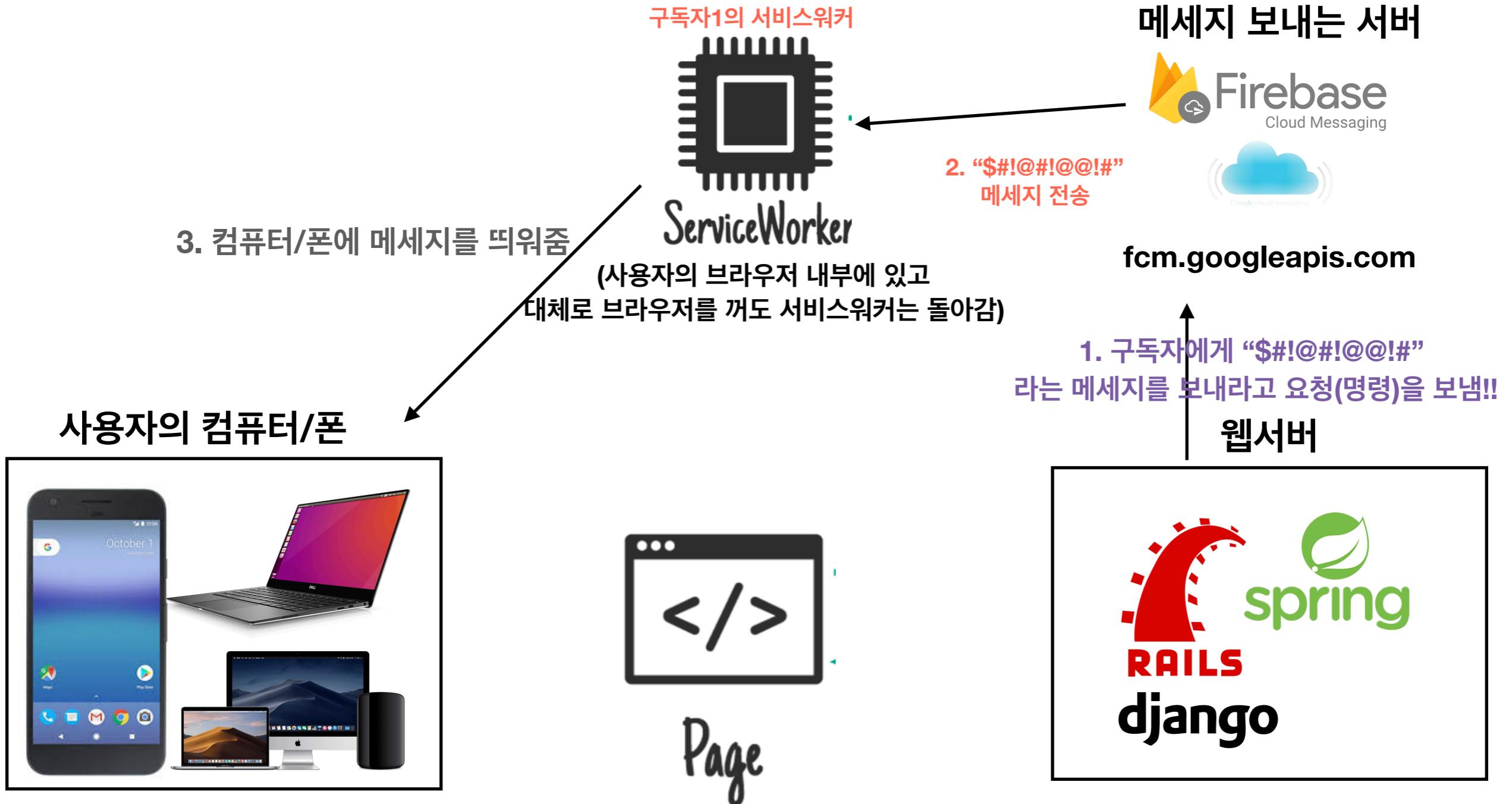


Page



3, 4, 5 완료.

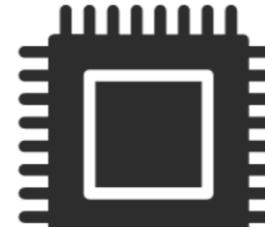
구독되어있는 상태에서 메세지 받는 프로세스



구독되어있는 상태에서 메세지 받는 프로세스

```
Webpush.payload_send(  
    message: { title: '제목!!', content: '내용' }.to_json,  
    endpoint: noti.endpoint,  
    p256dh: noti.p256dh,  
    auth: noti.auth,  
    ttl: 24 * 60 * 60,  
    vapid: {  
        subject: 'mailto:gooooo@gmail.com',  
        public_key: ENV['VAPID_PUBLIC_KEY'],  
        private_key: ENV['VAPID_PRIVATE_KEY']  
    }  
)
```

구독자1의 서비스워커



2. "\$#!@#!@@!#" 메세지 전송

메세지 보내는 서버



fcm.googleapis.com

1. 구독자에게 "\$#!@#!@@!#"라는 메세지를 보내라고 서버에 요청(명령)을 보냄!!

웹서버



noti = Notification.last와 같은
구독자 디비값



Page

Webpush.payload_send 메소드를 사용하면 1번, 2번 로직이 구현된다.
즉 구독자(의 서비스워커)에 메세지를 보내도록하는 과정이 구현된다.

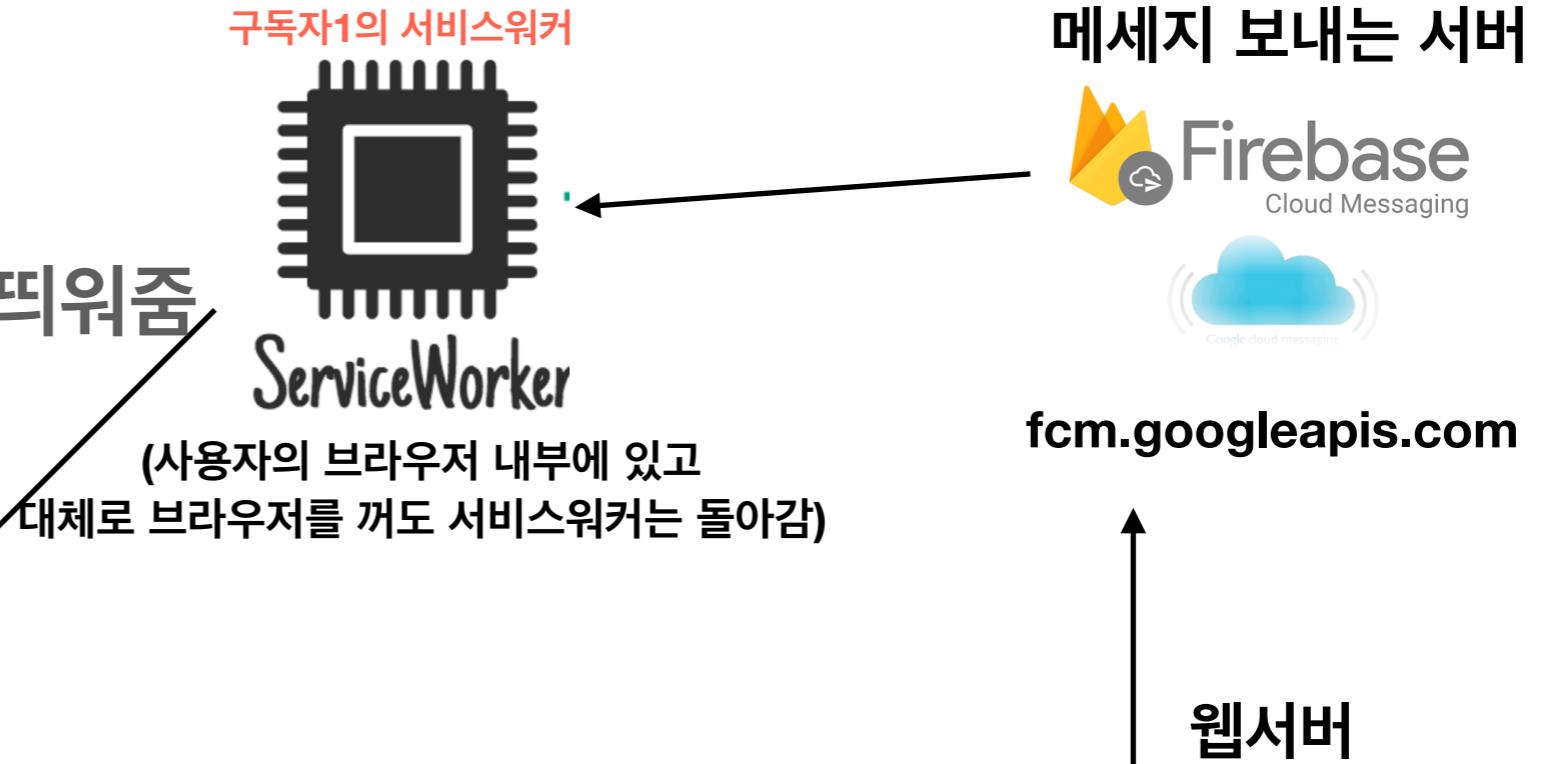
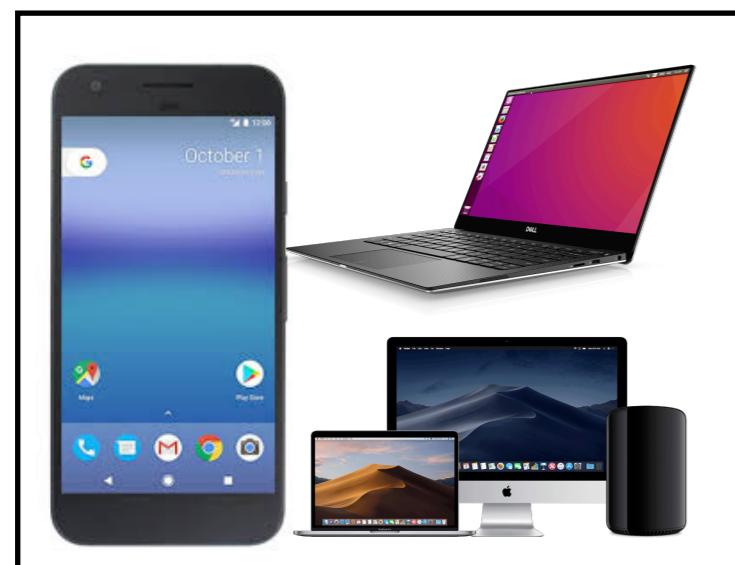
posts_controller

```
26 def create
27   @post = Post.new(post_params)
28
29   Notification.all.each do |notif|
30     # begin rescue end 구문은 예외처리를 위해 서임.
31     begin
32       Webpush.payload_send(
33         message: {"title": "새로운 게시글이 생성되었습니다.", "content": params[:title].to_json,
34         endpoint: notif.endpoint,
35         p256dh: notif.p256dh,
36         auth: notif.auth,
37         ttl: 24 * 60 * 60,
38         vapid: {
39           subject: 'mailto: hello@gmail.com',
40           public_key: ENV['VAPID_PUBLIC_KEY'],
41           private_key: ENV['VAPID_PRIVATE_KEY']
42         }
43       )
44     rescue
45     end
46   end
47
48   respond_to do |format|
49     if @post.save
50       format.html { redirect_to @post, notice: 'Post was successfully created.' }
51       format.json { render :show, status: :created, location: @post }
52     else
53       format.html { render :new }
54       format.json { render json: @post.errors, status: :unprocessable_entity }
55     end
56   end
57 end
```

새로운 게시글이 생성되었을때 알림을 보내주고싶다면 이런식으로 create액션에 Webpush.payload_send 를 실행하게 작성하면됨.

구독되어있는 상태에서 메세지 받는 프로세스

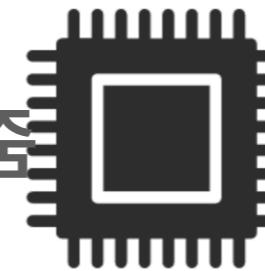
3. 컴퓨터/폰에 메세지를 띄워줌



구독되어있는 상태에서 메세지 받는 프로세스

구독자1의 서비스워커

3. 컴퓨터/폰에 메세지를 띄워줌



메세지 보내는 서버



```
125 self.addEventListener('push', function (event) {  
126     // 새로운 알림이 떴을 때 이 이벤트가 작동한다.  
127     console.log(event)  
128     console.log("Go")  
129     console.log(event.data)  
130     var jsonbody = JSON.parse(event.data && event.data.text());  
131     console.log(jsonbody)  
132     var title = jsonbody.title || "New Blog Post";  
133     var body = jsonbody.content || "Come to my Site";  
134     var dir = 'ltr';  
135     var tag = "push-simple-demo-tag";  
136     var data = jsonbody.link || "/";  
137     var icon = "/images/icons/icon.png";  
138     var renotify = true;  
139     var actions = [{ action: 'confirm', title: 'Visit', icon: '/images/icons/icon.png' }];  
140  
141     event.waitUntil(self.registration.showNotification(title, { body: body, icon: icon, dir: dir, tag: tag, renotti  
142 });  
143     self.addEventListener('notificationclick', function (event) {  
144         // 알림을 클릭하였을 때 이 이벤트가 작동함.  
145         var notification = event.notification;  
146         var action = event.action;  
147         var link = notification.data;  
148         console.log(event);  
149         if (action === 'confirm') {  
150             console.log('Confirm was chosen');  
151             notification.close();  
152         } else {  
153             console.log(action);  
154             notification.close();  
155         }  
156     }  
157  
158     event.waitUntil(clients.matchAll({  
159         type: "window"
```

serviceworker.js.erb

!!@!#"
!(명령)을 보냄!!



코드를 완전히 이해하지는 못해도 됨.

왜냐하면 코드들을 제대로 이해할려면 봐야
할 문서들이 많기에...

2시간만에 이해하기에는 내용자체가 너무 많음.

맨뒤에 도움될만한 링크들을 정리해놓았음.

처음에는 일단

“플로우”만 이해하면 된다.

그다음은 “플로우”를 이루는 각 단계를
위한 코드를 생각하면 되는것

구독하는 프로세스



메세지 보내는 서버



fcm.googleapis.com

사용자의 컴퓨터/폰



1. 사용자가 (서비스워커가 구축되었는)사이트 접속



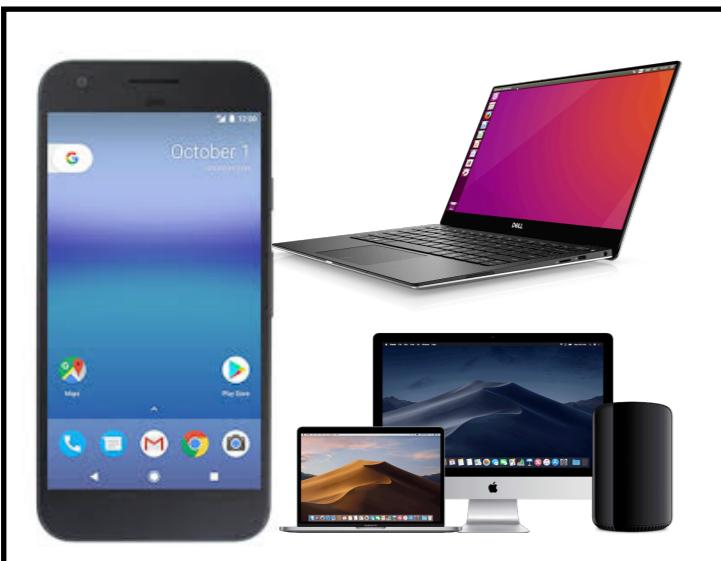
Page

웹서버



구독하는 프로세스

사용자의 컴퓨터/폰

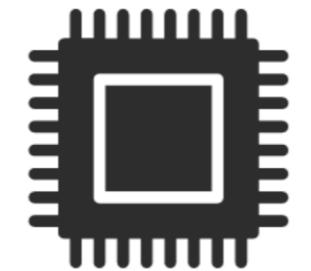


1. 사용자가 사이트 접속



2. 구독 버튼 클릭

Page



ServiceWorker

(사용자의 브라우저 내부에 있고
대체로 브라우저를 꺼도 서비스워커는 돌아감)

메세지 보내는 서버

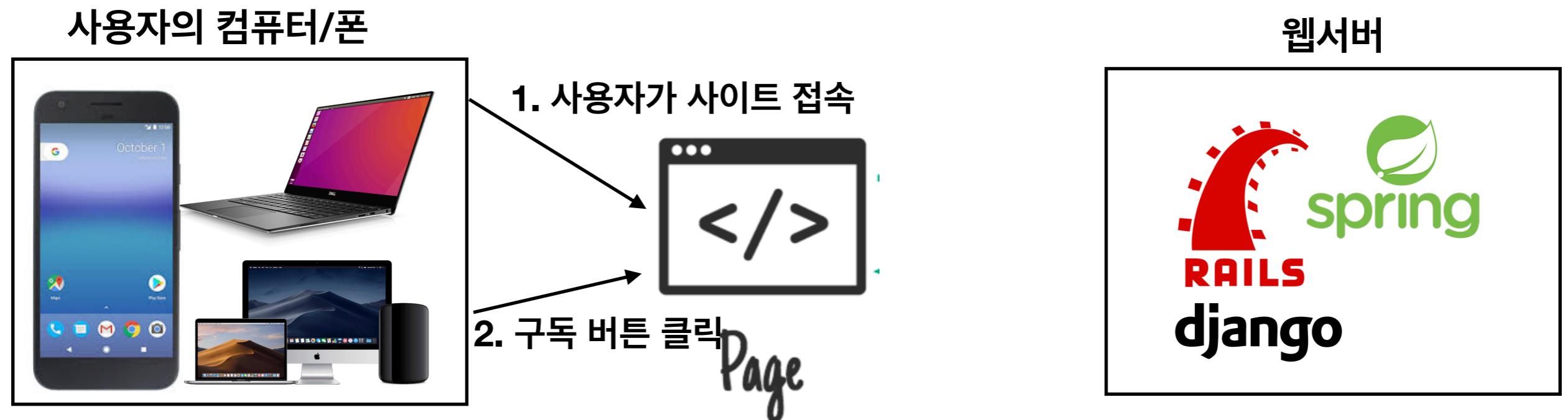
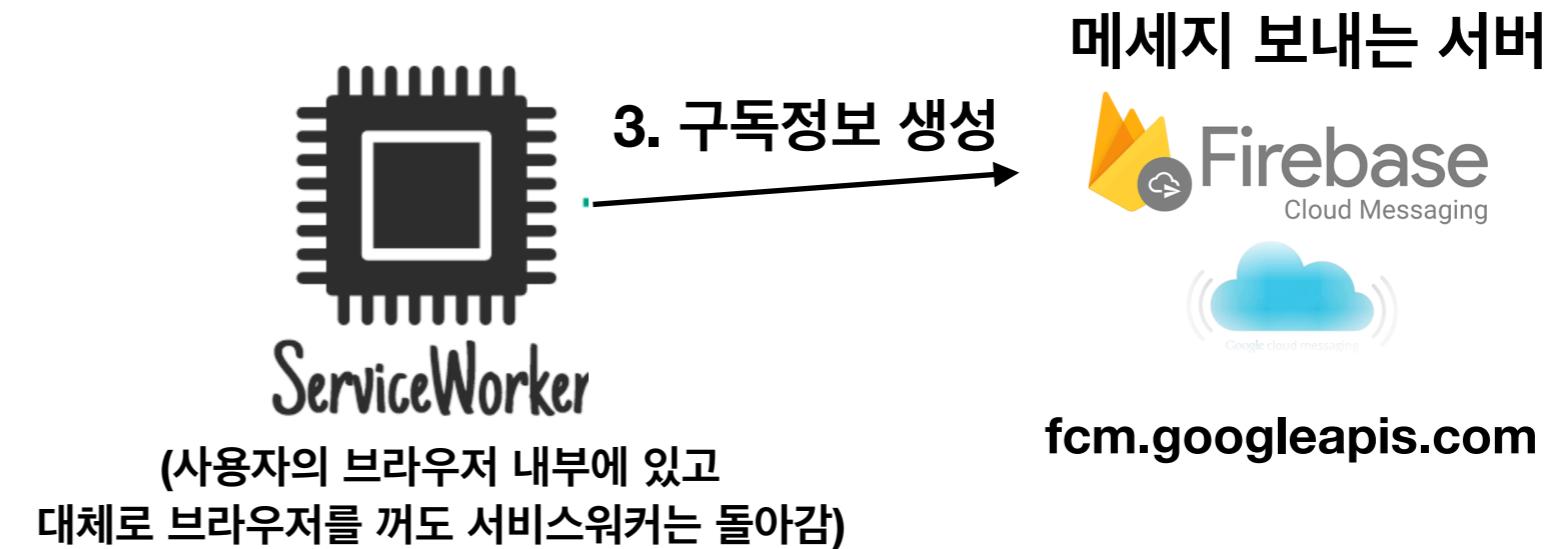


fcm.googleapis.com

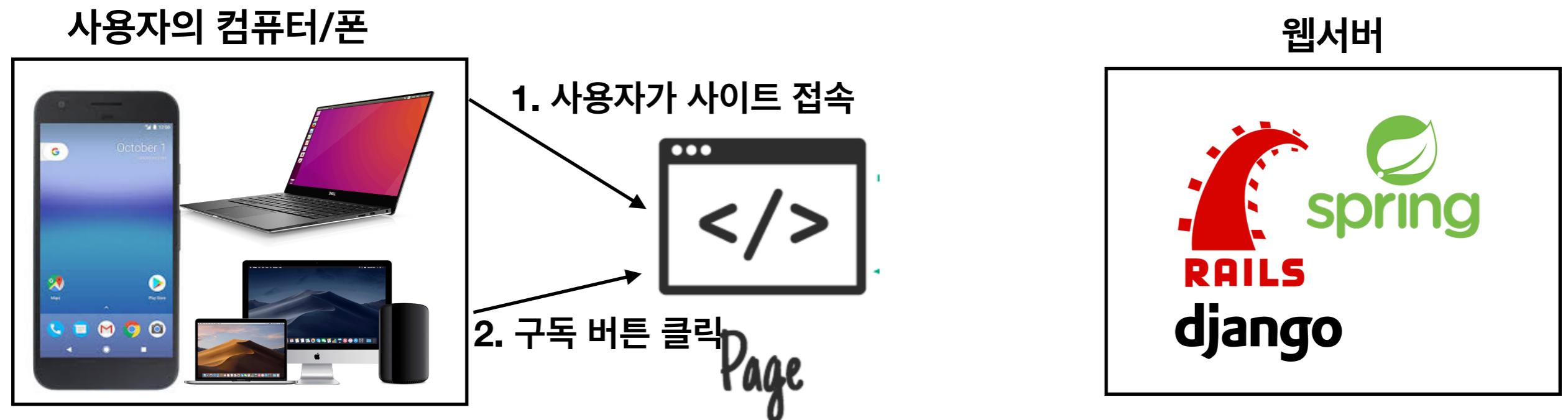
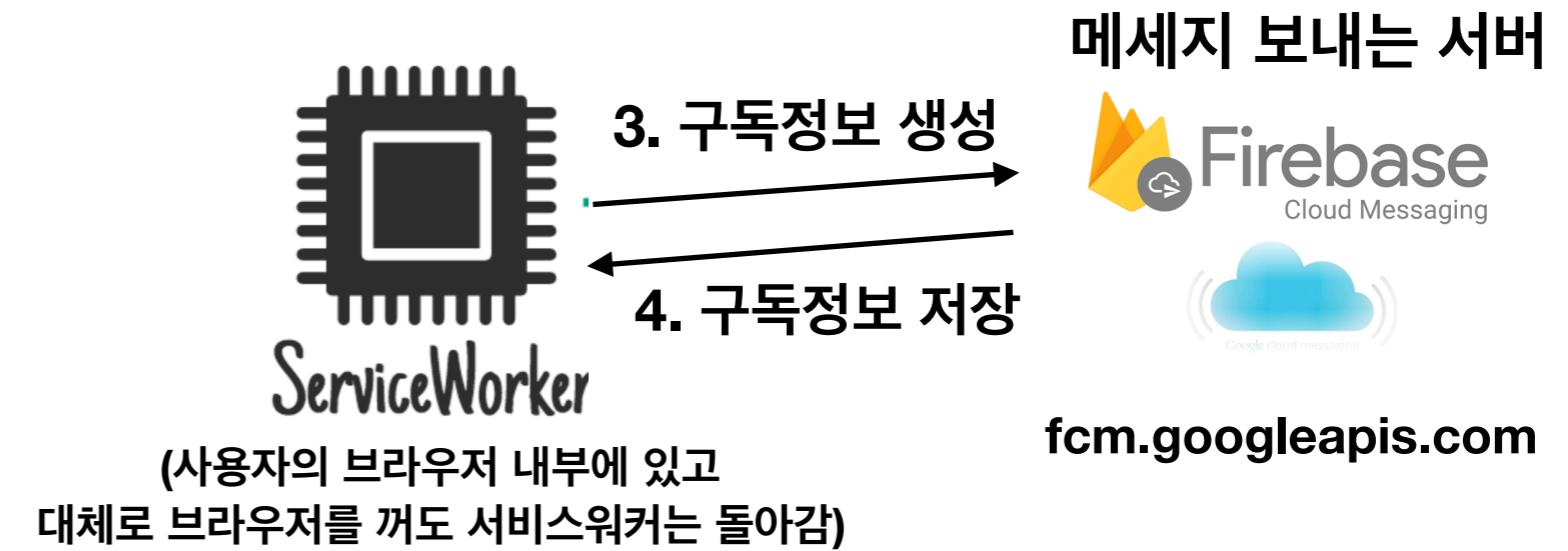
웹서버



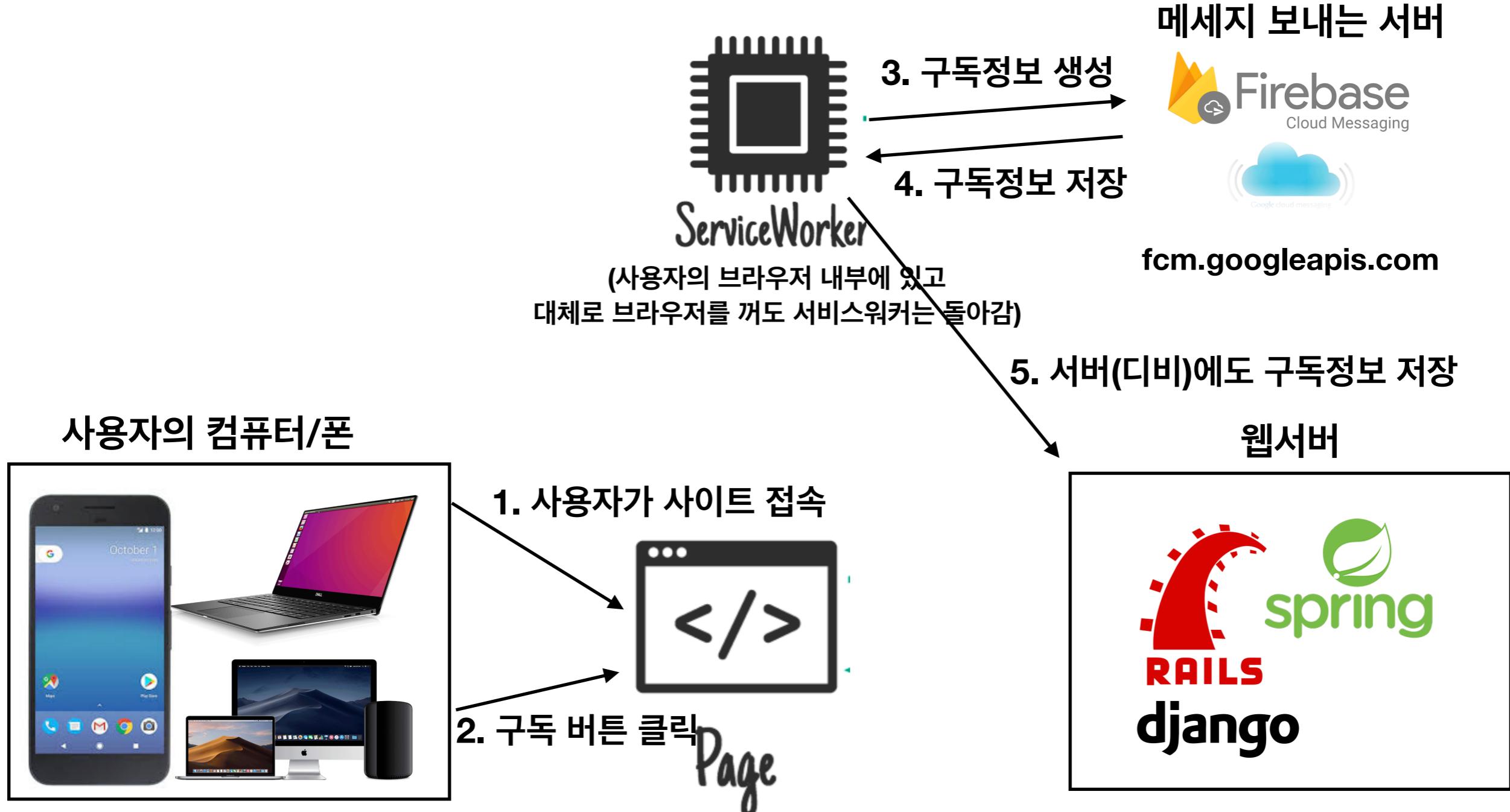
구독하는 프로세스



구독하는 프로세스



구독하는 프로세스

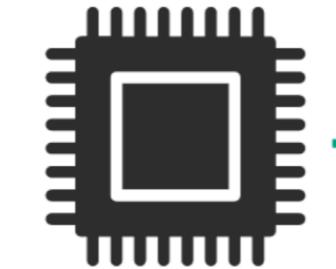


구독되어있는 상태에서 메세지 받는 프로세스

메세지 보내는 서버



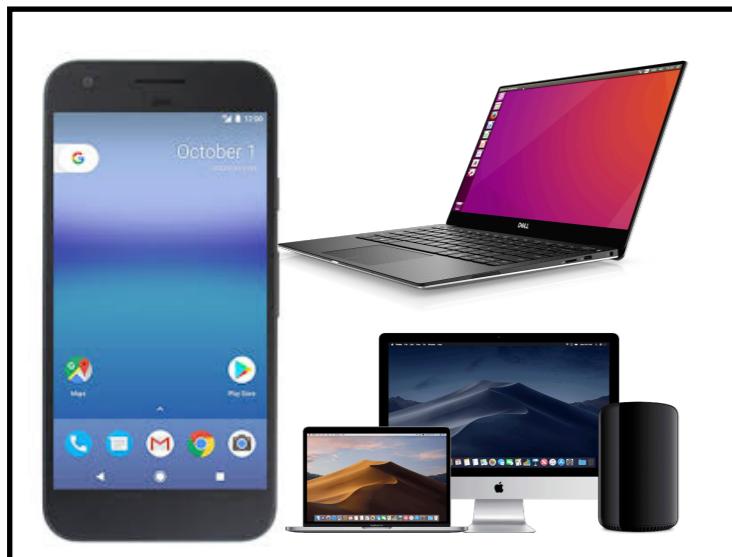
fcm.googleapis.com



ServiceWorker

(사용자의 브라우저 내부에 있고
대체로 브라우저를 꺼도 서비스워커는 돌아감)

사용자의 컴퓨터/폰



Page

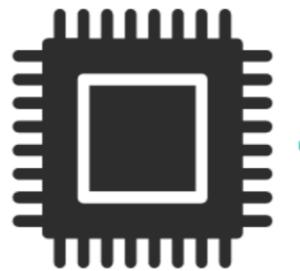
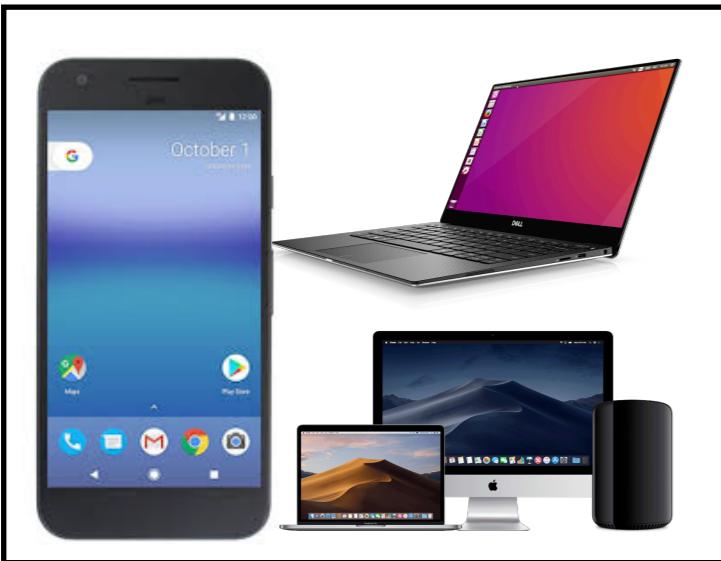
웹서버



1. 새로운 게시글이 생성됨

구독되어있는 상태에서 메세지 받는 프로세스

사용자의 컴퓨터/폰



ServiceWorker

(사용자의 브라우저 내부에 있고
대체로 브라우저를 꺼도 서비스워커는 돌아감)



Page

메세지 보내는 서버



fcm.googleapis.com

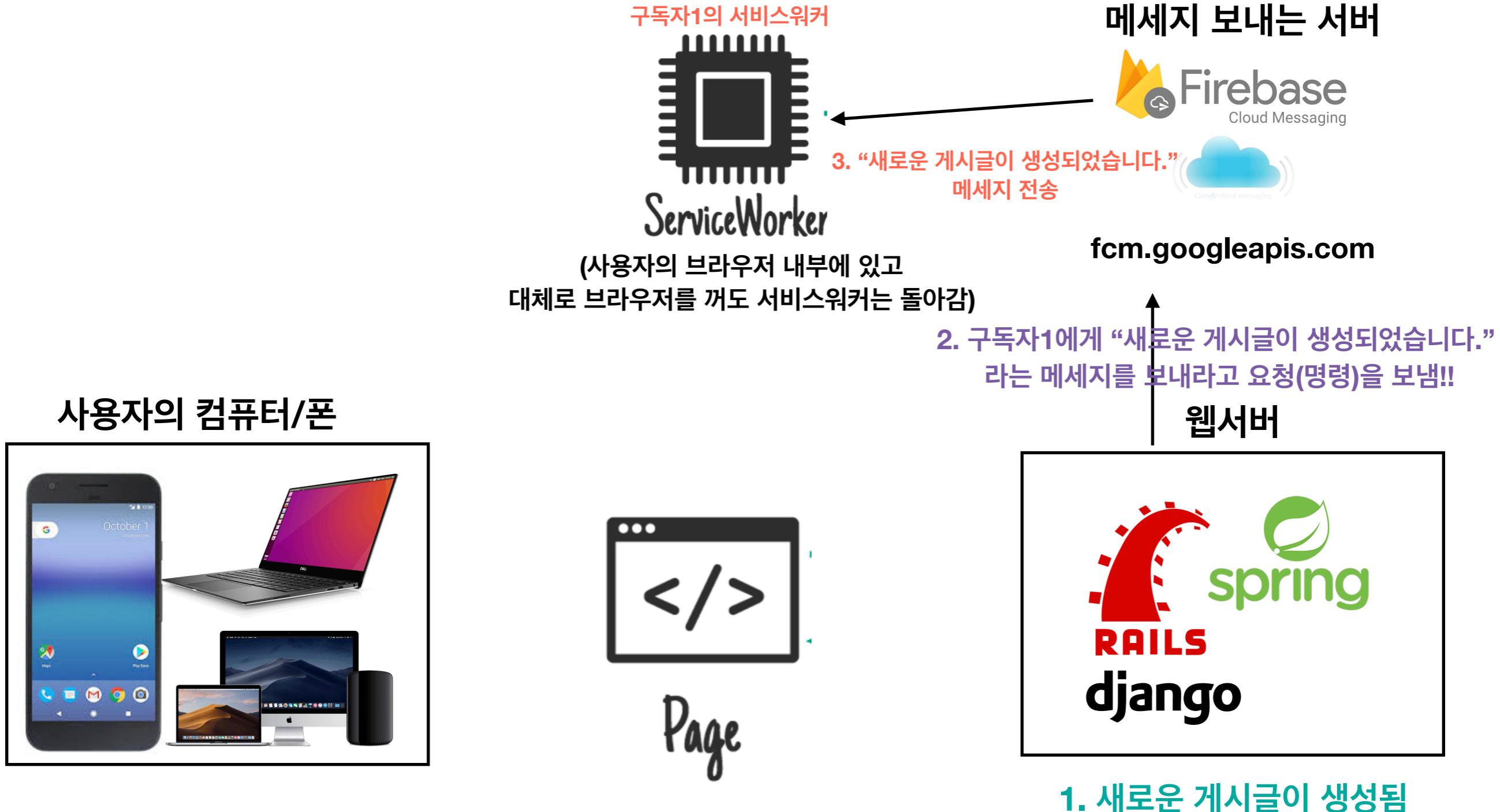
2. 구독자1에게 “새로운 게시글이 생성되었습니다.”
라는 메세지를 보내라고 요청(명령)을 보냄!!

웹서버

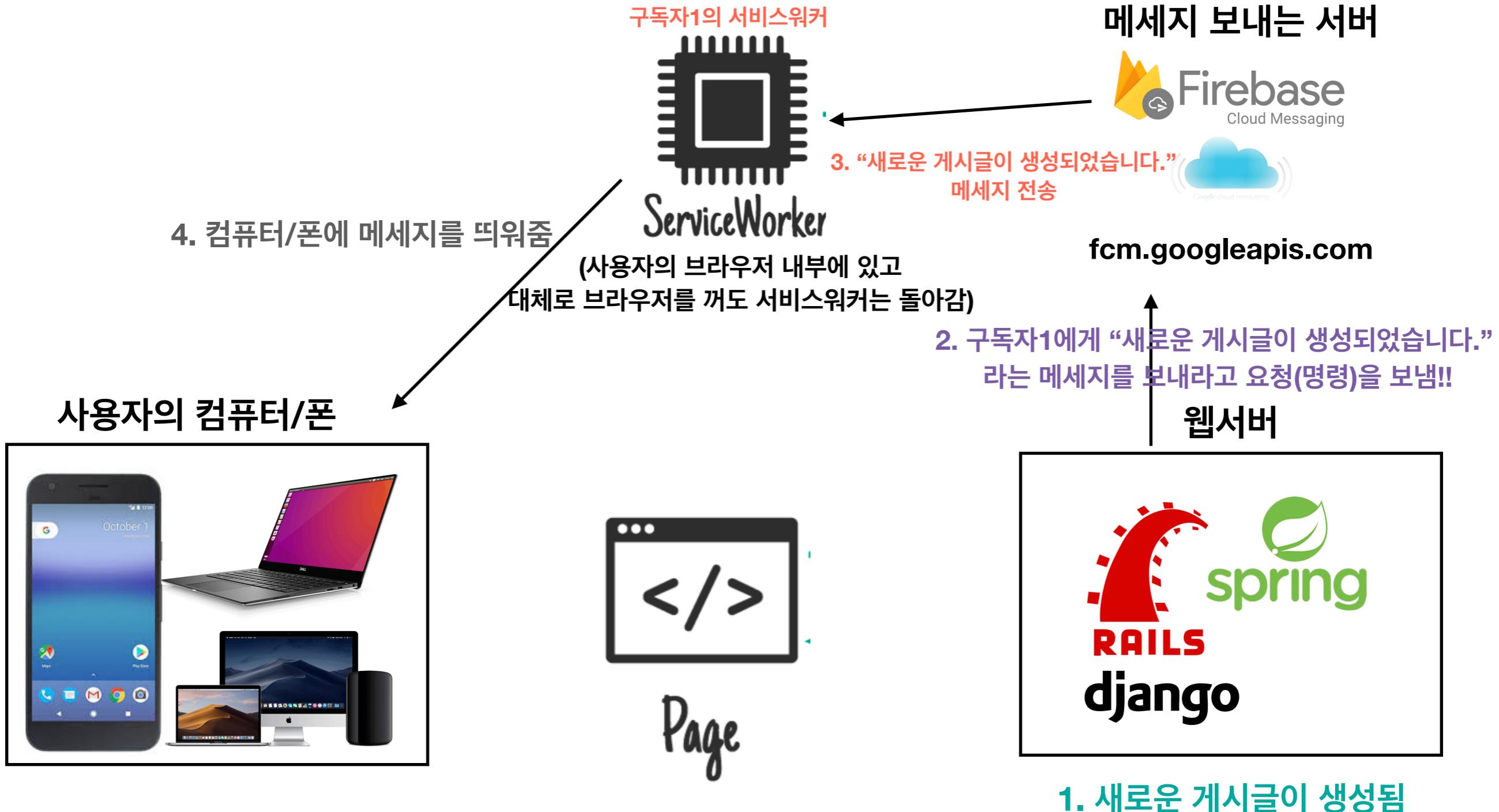


1. 새로운 게시글이 생성됨

구독되어있는 상태에서 메세지 받는 프로세스



구독되어있는 상태에서 메세지 받는 프로세스



진짜 끝.

유용한 링크

Articles

[Introduction to Service Worker](#)

[Using Service Workers Today](#)

[JavaScript Promises: an Introduction](#)

[The offline cookbook](#)

[Using Service Workers \(MDN\)](#)

Sample Code & Recipes

[Prefetching Resources During Registration](#)

[Selective Read-through Caching](#)

[Returning Mock Responses](#)

[Returning Fallback Responses](#)

[...and more!](#)

Reference Apps

[Trained to Thrill](#)

[Chrome Dev Summit](#)

[SVGOMG](#)

[Google I/O 2015](#)

[Air Horn](#)

Documentation

[The Service Worker API \(draft\)](#)

[Service Worker Specification \(draft\)](#)

[Fetch API Specification \(living standard\)](#)

[Promises](#)

<https://jakearchibald.github.io/isserviceworkerready/resources.html>

유용한 링크

구글 디벨로퍼 사이트

MDN 사이트

<https://developers.google.com/web/fundamentals/primers/service-workers/>

<https://developers.google.com/web/fundamentals/codelabs/push-notifications/?hl=ko>

https://developer.mozilla.org/ko/docs/Web/API/Service_Worker_API/Using_Service_Workers

<https://developers.google.com/web/fundamentals/instant-and-offline/offline-cookbook/>

<https://www.netlify.com/blog/2017/10/31/service-workers-explained/>

<https://infrequently.org/2015/06/progressive-apps-escaping-tabs-without-losing-our-soul/>

<https://www.slideshare.net/jayjin0427/progressive-web-app-feat-react-django-82499585>

<https://www.slideshare.net/GihyoJoshuaJang/the-future-of-web-progressive-web-app>