



# Progressive Web Applications

Mobilné technológie a aplikácie  
FIIT STU v Bratislave

prof. Ing. Ivan Kotuliak, PhD.  
Ing. Marek Galinski

# Úvod



- Neexistuje oficiálna definícia PWA
- Pojem sa používa od cca roku 2015
- Väčšinou spĺňa nejaké generické kritéria

# Webstránka vs. PWA



**Web + HTTPS + JSON Web Manifest + Service Worker**

**=**

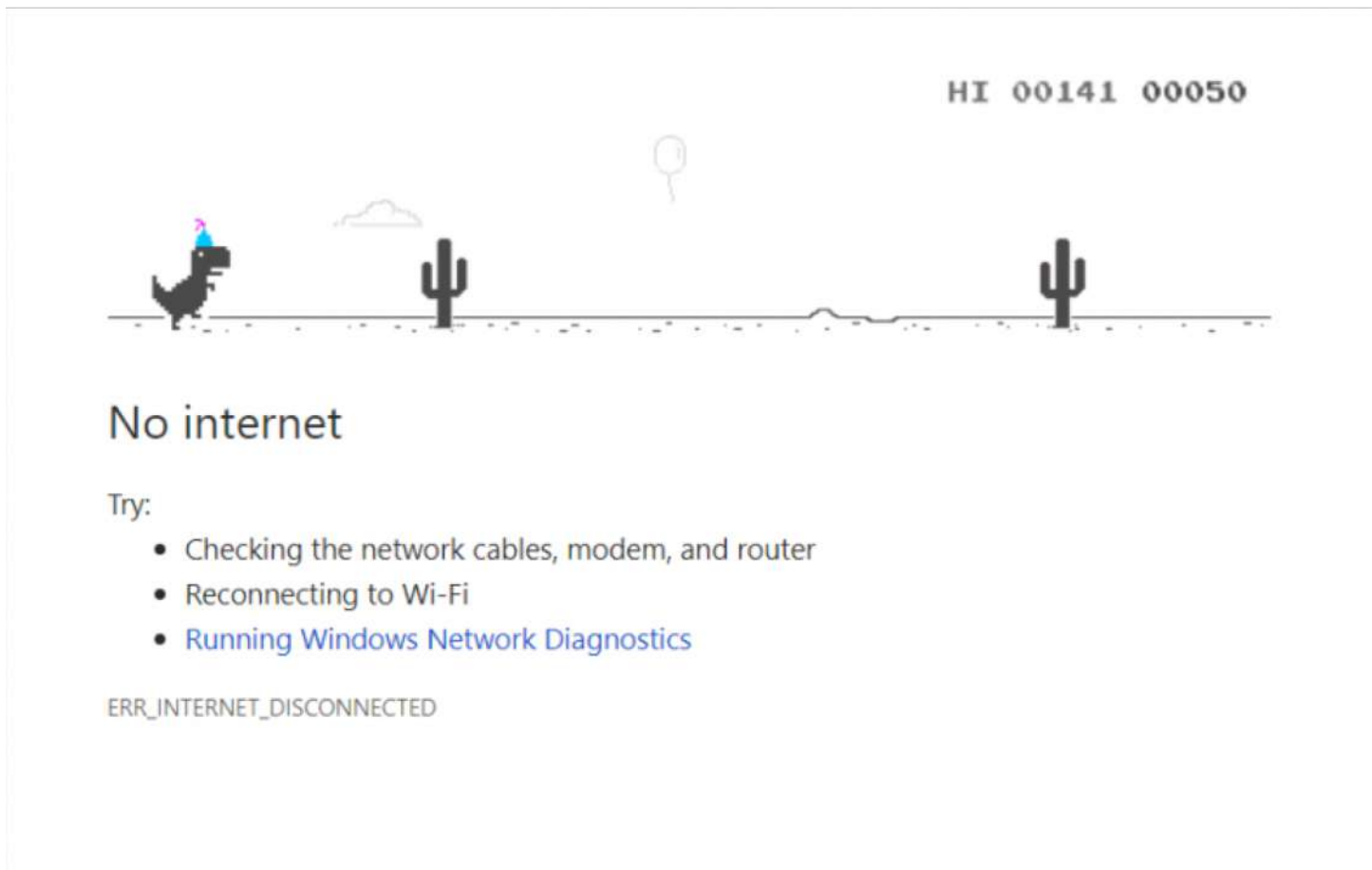
**Progressive Web Application**

# Čo hovorí Google?

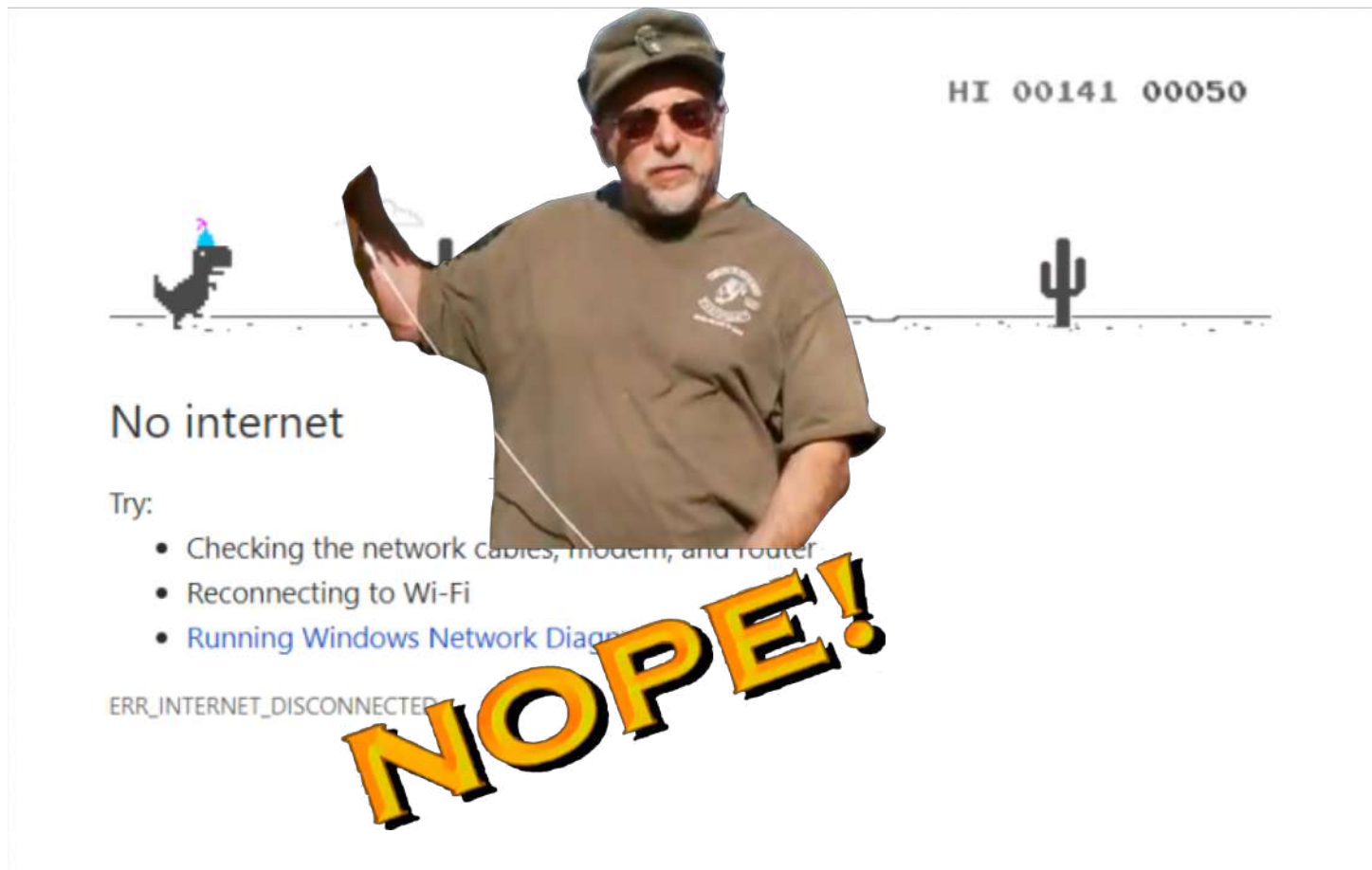


- PWA má spĺňať 3 základné atribúty
- 1. Spoľahlivá
- 2. Rýchla
- 3. Používateľský prívetivá

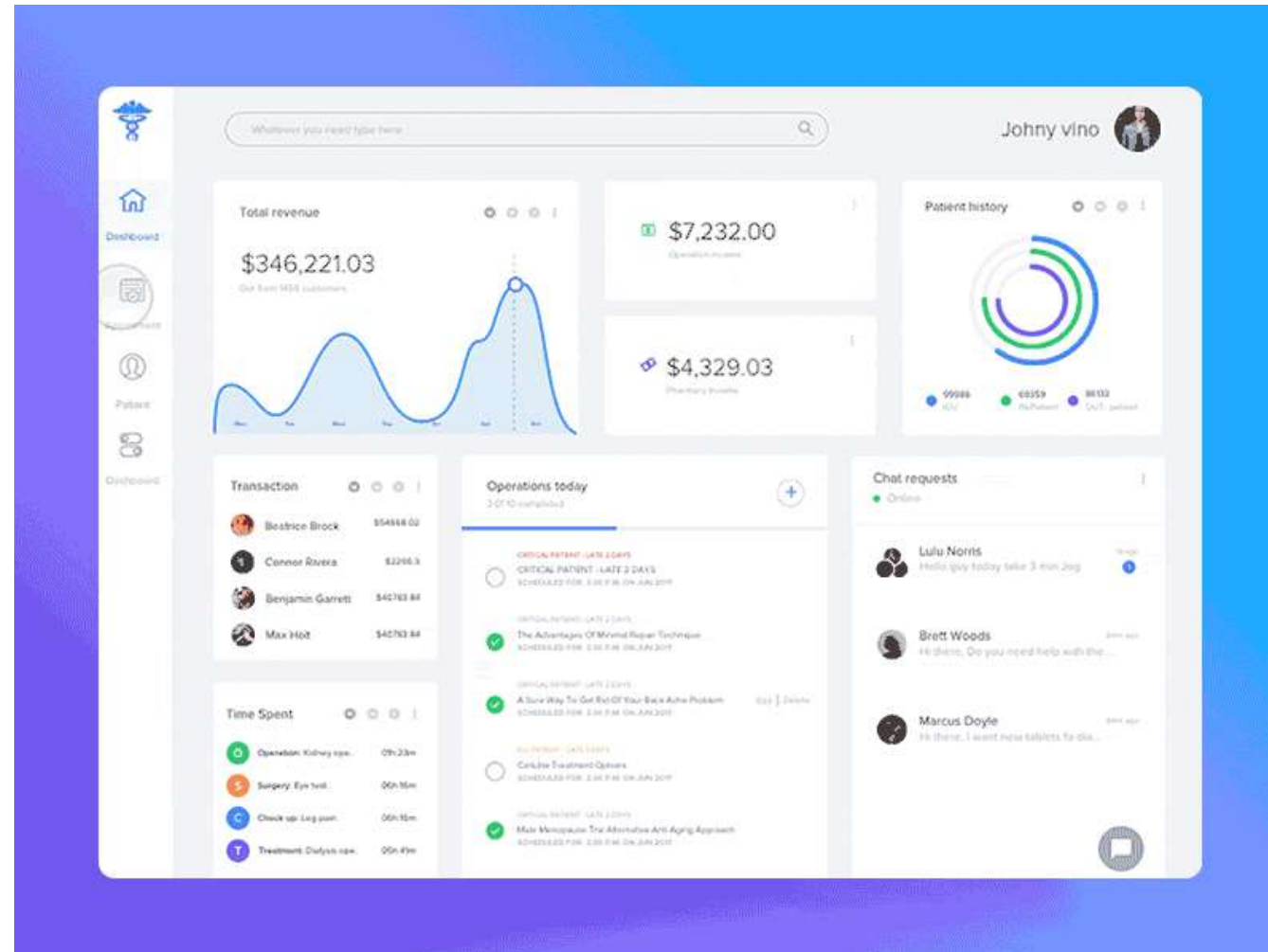
# Čo hovorí Google?



# Čo hovorí Google?



# Čo hovorí Google?



# Čo hovorí Wikipedia?



- „ Progressive Web Apps (PWAs) are web applications that are regular web pages or websites, but can appear to the user like traditional applications or native mobile applications. The application type attempts to combine features offered by most modern browsers with the benefits of a mobile experience. “



# Čo hovorí MDN?



- „Progressive web apps use modern web APIs along with traditional progressive enhancement strategy to create cross-platform web applications.”
- „PWAs should be discoverable, installable, linkable, network independent, progressive, re-engageable, responsive, and safe.“

# Čo hovorí Alex Russel?



- **„ ...they're just websites that took all the right vitamins.”**

*Alex Russel, Google & ECMA*

# Prečo chceme vyvíjať PWAs

- Chceme byť nezávislí od platformy



# Prečo chceme vyvíjať PWAs

- Chceme byť nezávislí od zariadenia (jeho veku, jeho veľkosi displeja)



**PWA musí byť responzívna**

# Prečo chceme vyvíjať PWAs



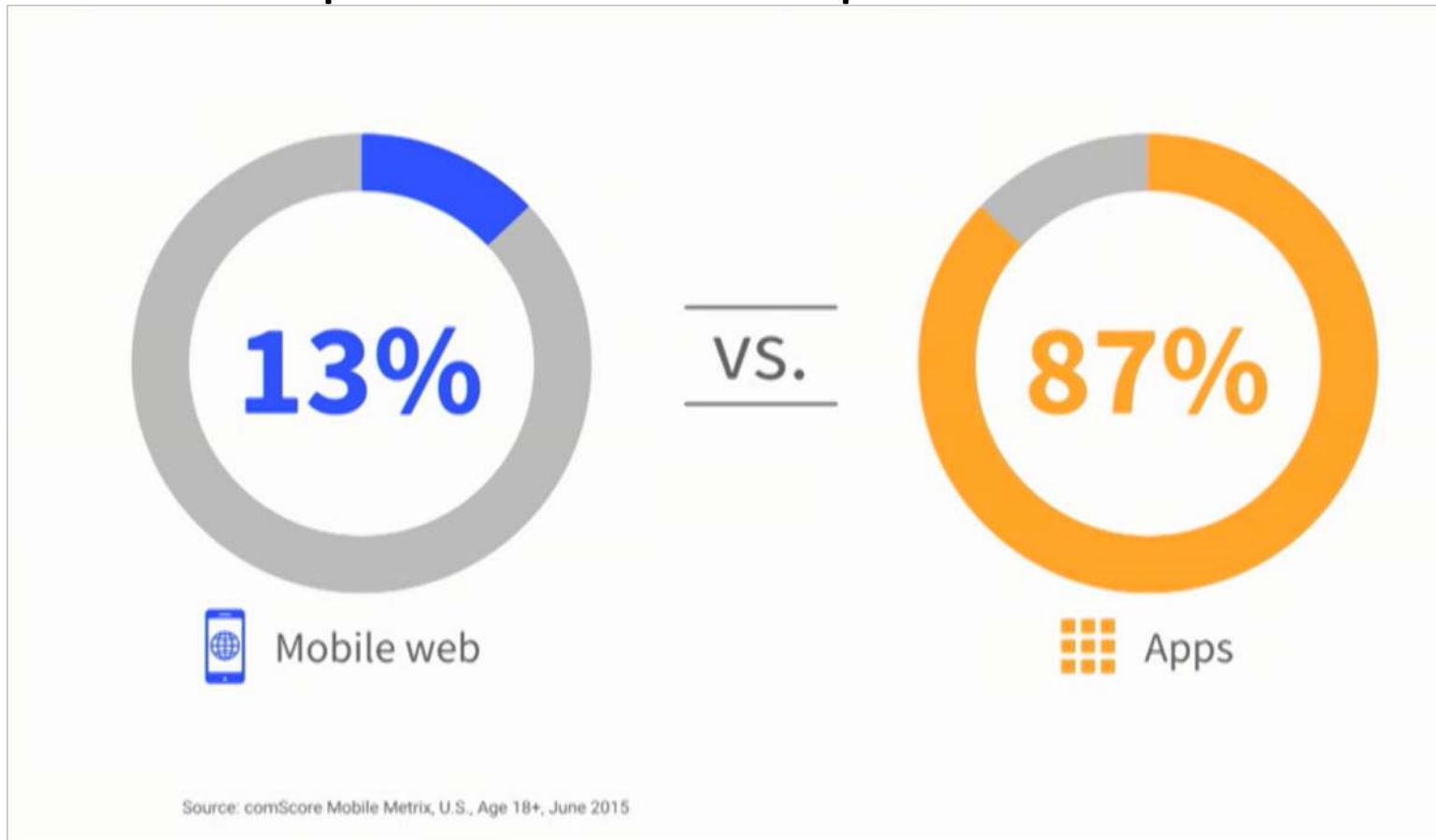
- Tradičný web má niekoľko problémov, ktorým dnes čelíme.
- Tradičné aplikácie majú niekoľko problémov, ktorým dnes čelíme.
- Používatelia sa nesprávajú úplne racionálne

# Prečo chceme vyvíjať PWAs

- Používatelia nechcú inštalovať aplikácie
  - **Priemerný používateľ inštaluje 0 aplikácií mesačne**  
*(freeCodeCamp)*

# Prečo chceme vyvíjať PWAs

- Používatelia chcú používať natívne aplikácie

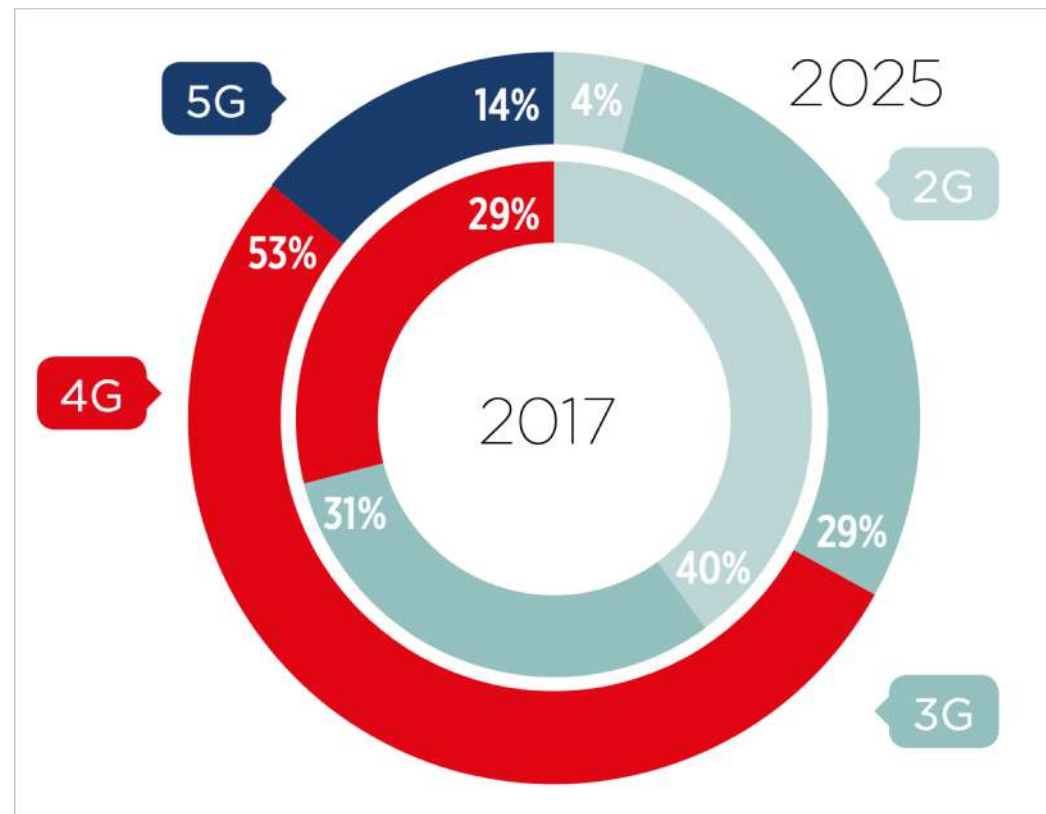




# Prečo chceme vyvíjať PWAs

- Nie všetci majú prístup k rýchlemu mobilnému internetu (2017)

- 2G - 40%
- 3G - 31%
- 4G - 29%



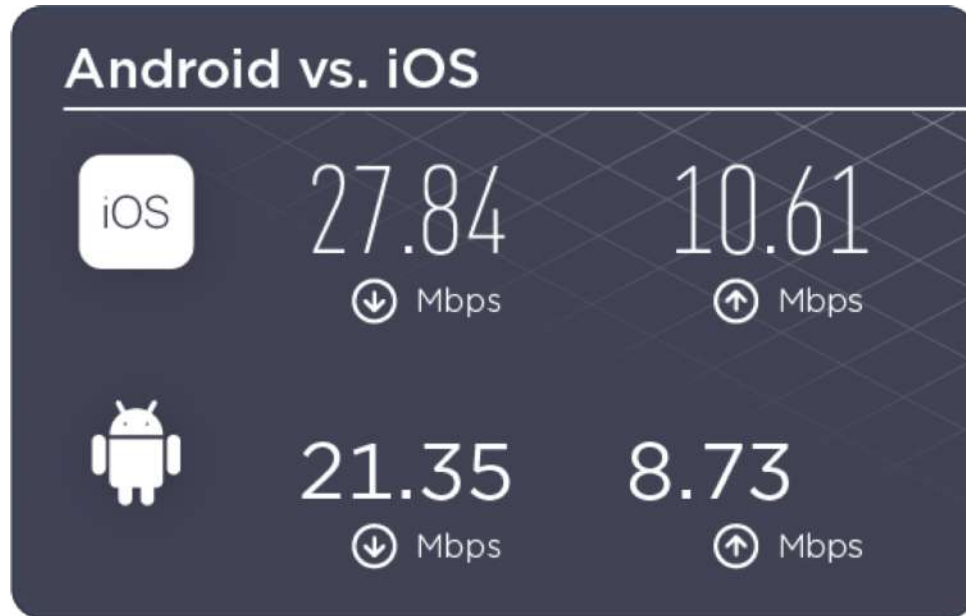
# Prečo chceme vyvíjať PWAs



- Nie všetci majú prístup k rýchlemu mobilnému internetu (2017)
- **Tradičný web kde ťaháme vždy UI aj dáta je príliš pomalý**

**Priemerná doba, kým používateľ  
zavrie tab prehliadača, ktorý sa  
nenačíta, je 3 sekundy.**

# Mimochodom, rok 2018



SPEEDTEST  
by OOKLA



SPEEDTEST  
by OOKLA

# Prečo chceme vyvíjať PWAs



- **Ako tieto problémy rieši prístup PWA?**

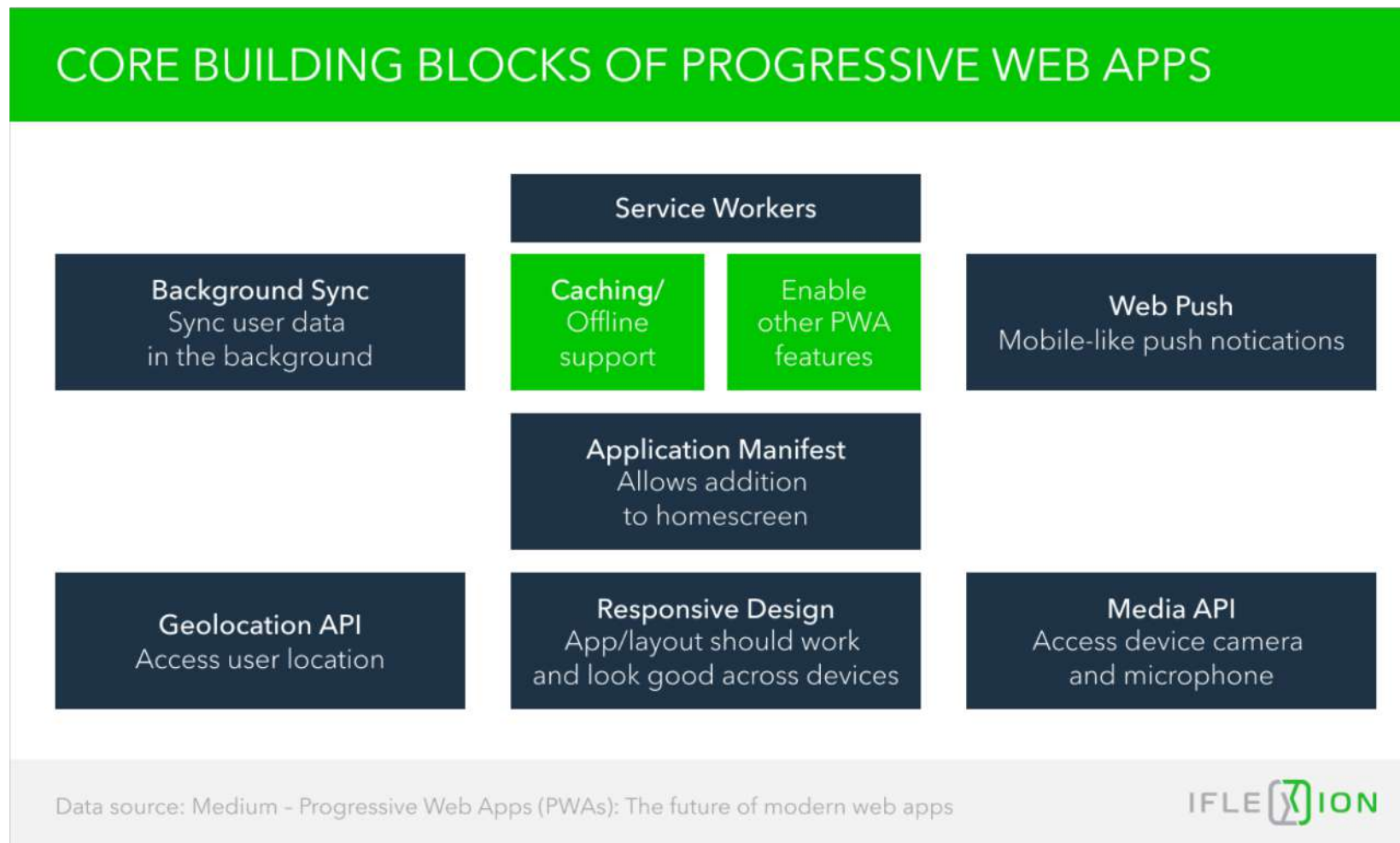
# Prečo chceme vyvíjať PWAs



- **Ako tieto problémy rieši prístup PWA?**

- Sú rýchlejšie – UI je súčasťou aplikácie, ktorú máme lokálne na zariadení, statické dáta sa cachuju, dáta v aplikácii ostávajú aj po odpojení
- Správajú sa natívne - interaktívne a rýchle UI, podpora push notifikácií
- Ošetrujú offline stavy – v prípade, že zariadenie stratí pripojenie, aplikácia vie vhodne zareagovať (service workery)

# Ako si môžeme predstaviť PWA



**PWA nemusí být nainštalovaná  
(ale může)**



# Cesta od webu k PWA



**Web + HTTPS + JSON Web Manifest + Service Worker**

**=**

**Progressive Web Application**

# Cesta od webu k PWA

- **Web, ktorý chceme považovať za PWA musí:**

- mať responzívny dizajn



- podporovať HTTPS



# Cesta od webu k PWA



- **Potrebujeme Web App Manifest (JSON)**, pre pridanie na homescreen

```
1  {
2    "name": "Trending Meme",
3    "short_name": "Meme",
4    "theme_color": "#0f1113",
5    "background_color": "#111316",
6    "display": "standalone",
7    "orientation": "portrait",
8    "Scope": "/",
9    "start_url": "/",
10   "icons": [
11     {
12       "src": "images/icons/icon-72x72.png",
13       "sizes": "72x72",
14       "type": "image/png"
15     },
16     {
17       "src": "images/icons/icon-96x96.png",
18       "sizes": "96x96",
19       "type": "image/png"
20     },
21     {
22       "src": "images/icons/icon-128x128.png",
23       "sizes": "128x128",
24       "type": "image/png"
25     },
26     {
27       "src": "images/icons/icon-144x144.png",
28       "sizes": "144x144",
29       "type": "image/png"
30     }
31   ]
32 }
```

```
2 "name": "Trending Meme",
3 "short_name": "Meme",
4 "theme_color": "#0f1113",
5 "background_color": "#111316",
6 "display": "standalone",
7 "orientation": "portrait",
8 "Scope": "/",
9 "start_url": "/",
10 "icons": [
11   {
12     "src": "images/icons/icon-72x72.png",
13     "sizes": "72x72",
14     "type": "image/png"
15   },
16   {
17     "src": "images/icons/icon-96x96.png",
18     "sizes": "96x96",
19     "type": "image/png"
20   },
21   {
22     "src": "images/icons/icon-128x128.png",
23     "sizes": "128x128",
24     "type": "image/png"
25   },
26   {
27     "src": "images/icons/icon-144x144.png",
28     "sizes": "144x144",
```

# Cesta od webu k PWA



- **Potrebujeme využiť Service Workery**
- A service worker is an event-driven worker registered against an origin and a path. It takes the form of a JavaScript file that can control the web-page/site that it is associated with, intercepting and modifying navigation and resource requests, and caching resources in a very granular fashion to give you complete control over how your app behaves in certain situations (the most obvious one being when the network is not available).

# Cesta od webu k PWA



- **Potrebujeme využit' Service Workery**
- A service worker is an **event-driven worker** registered against an origin and a path. It takes the **form of a JavaScript file** that **can control the web-page** that it is associated with, intercepting and **modifying navigation and resource requests**, and **caching resources** in a very granular fashion to give you complete **control over how your app behaves in certain situations** (the most obvious one being when the network is not available).

# Cesta od webu k PWA



- **A čo je vlastne worker?**
- „Worker represents a **background task** that can be easily created and **can send messages back to its creator**”
- Beží v samostatnom threade ako JS kód na webe
  - Neblokuje, plne asynchrónny

# Cesta od webu k PWA



- **Príklad Service Workera – registrácia SW**

```
if ('serviceWorker' in navigator) {
  window.addEventListener('load', function() {
    navigator.serviceWorker.register('/sw.js').then(function(registration) {
      // Registration was successful
      console.log('ServiceWorker registration successful with scope: ', registration.scope);
    }, function(err) {
      // registration failed :(
      console.log('ServiceWorker registration failed: ', err);
    });
  });
}
```



# Cesta od webu k PWA

- **Príklad Service Workera – inštalácia SW**

```
var CACHE_NAME = 'my-site-cache-v1';
var urlsToCache = [
  '/',
  '/styles/main.css',
  '/script/main.js'
];

self.addEventListener('install', function(event) {
  // Perform install steps
  event.waitUntil(
    caches.open(CACHE_NAME)
      .then(function(cache) {
        console.log('Opened cache');
        return cache.addAll(urlsToCache);
      })
  );
});
```

# Cesta od webu k PWA



- **Príklad Service Workera – využitie SW**

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

# Cesta od webu k PWA



- **Čo je dobré vedieť o Service Workeroch?**
- Fungujú iba na HTTPS
- Potrebujú podporu prehliadača
  - Dnes to zvláda aj MS Edge a Safari
- Mimo cache – ak zmeníte kód SW, prehliadač si to skontroluje
- Nefungujú v “private / incognito” režime

**Nie, Internet Explorer service  
workery nepodporuje.**



# Cesta od webu k PWA



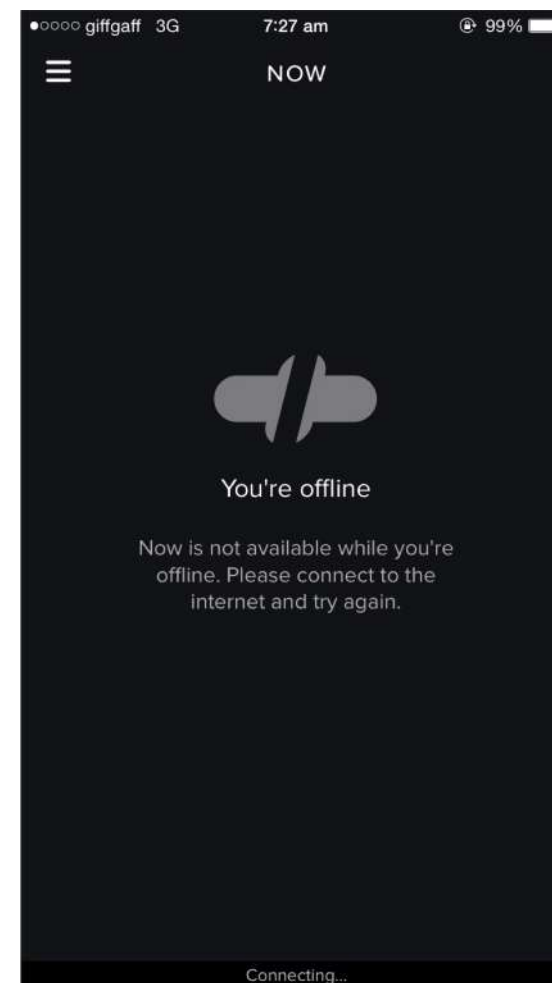
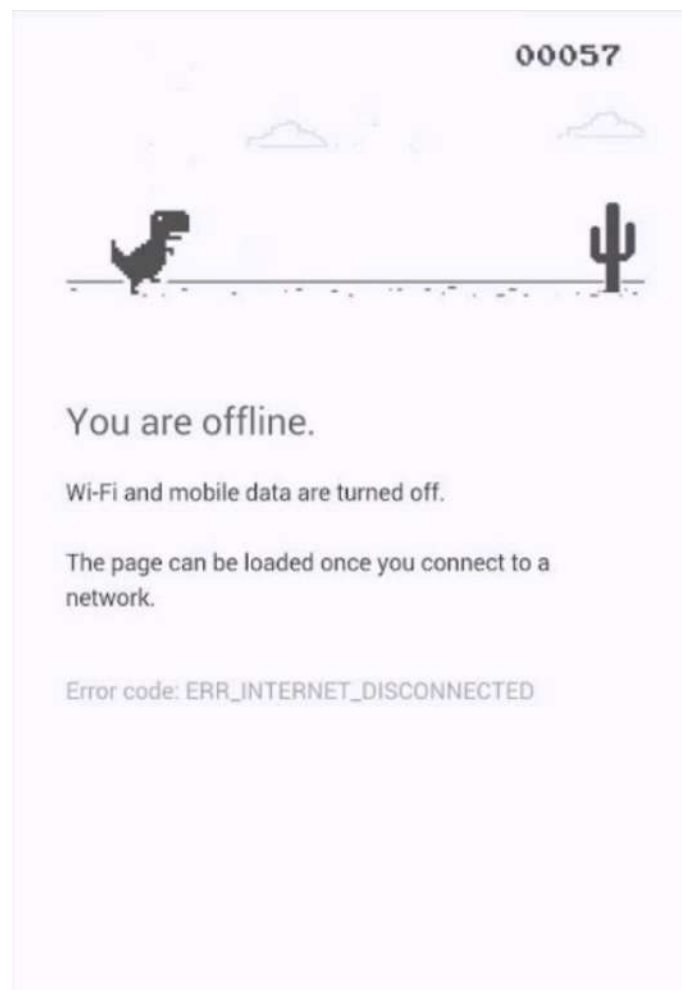
- **Príklad presmerovania na offline stránku**
  - Vytvoríme statickú cestu /offline.html
  - Nacachujeme /offline.html
  - Pri neúspešnom requeste presmerujeme na /offline.html

```
1  const CACHE_VERSION = 1;
2  ▼ let CURRENT_CACHES = {
3      offline: 'offline-v' + CACHE_VERSION
4  };
5  const OFFLINE_URL = 'offline.html';
6
7  ...
8
9  self.addEventListener('fetch', event => {
10     if (event.request.mode === 'navigate' ||
11         (event.request.method === 'GET' &&
12     ▼     event.request.headers.get('accept').includes('text/html'))) {
13         console.log('Handling fetch event for', event.request.url);
14         event.respondWith(
15     ▼     fetch(event.request).catch(error => {
16         console.log('Fetch failed; returning offline page instead.', error);
17         return caches.match(OFFLINE_URL);
18     })
19     );
20 }
```

# Cesta od webu k PWA



- **Výsledok?**



# Cesta od webu k PWA



- **Push notifikácie cez Service Workers**

- Nutná podpora prehliadača
- Worker sa zobudí na čas potrebný na zobrazenie notifikácie



# Cesta od webu k PWA

- **Push notifikácie cez Service Workers**
  - Používateľ musí povoliť

```
Notification.requestPermission(function(status) {  
    console.log('Notification permission status:', status);  
});
```

# Cesta od webu k PWA



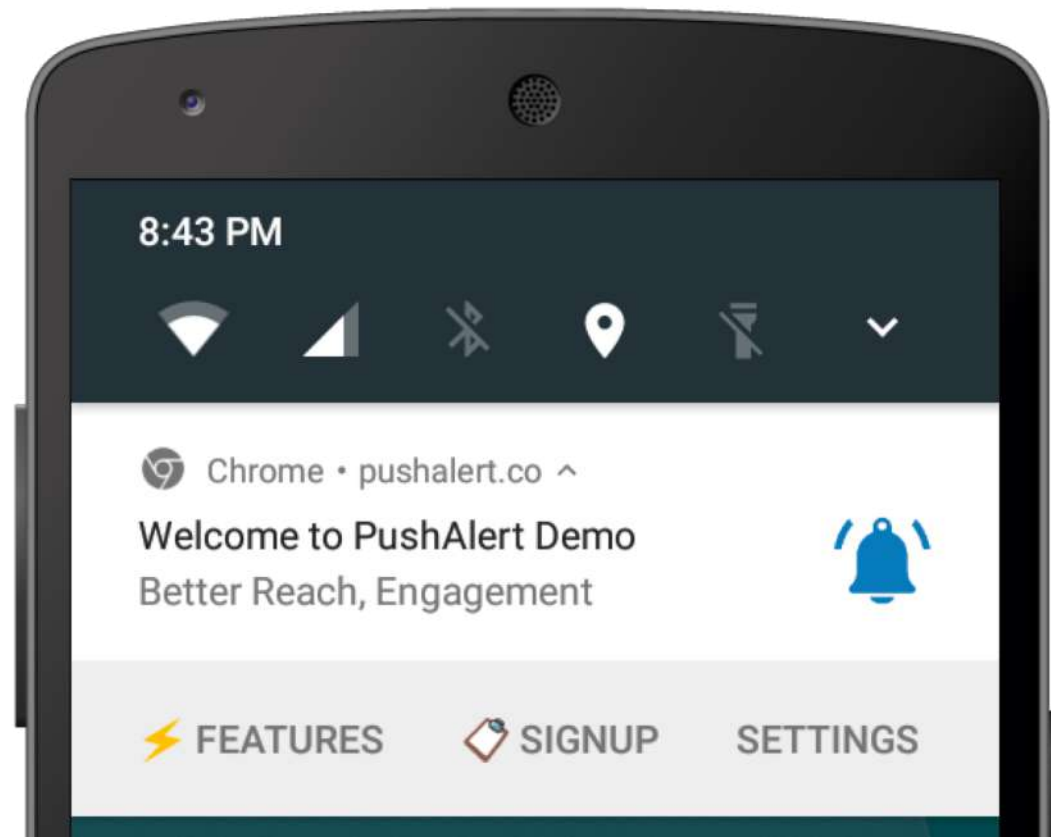
- **Push notifikácie cez Service Workers**
  - Zobrazenie notifikácie – základný kód

```
function displayNotification() {  
  if (Notification.permission == 'granted') {  
    navigator.serviceWorker.getRegistration().then(function(reg) {  
      reg.showNotification('Hello world!');  
    });  
  }  
}
```

# Cesta od webu k PWA



- **Push notifikácie cez Service Workers**
  - V notifikácii vidno zdroj



**Čo ďalej? Zrýchliť načítavanie!**

**Stránky, ktoré sa načítajú pod 5 sekúnd zarábajú 2-krát viac ako stránky, ktoré sa načítavajú 20 sekúnd. (*DoubleClick*)**

**Keď AutoAnything zrýchlil  
načítavanie o polovicu, predaje  
stúpili o 12%**



# Cesta od webu k PWA



- **Ako zrýchliť načítavanie webu?**

- Používať HTTP/2 – súbežné requesty, kompresia hlavičiek, ...
- Využívať **rel=preload** a **rel=preconnect**
- Minifikovať všetko – kódy, obrázky, uglifikáciu kódu
- Responzívne obrázky
- Neťaháť to, čo práve netreba

# Cesta od webu k PWA



- **Ako zrýchliť načítavanie webu?**

- rel=preload

- Poviete prehliadaču, ktoré assety sú kľúčové a majú sa načítať okamžite

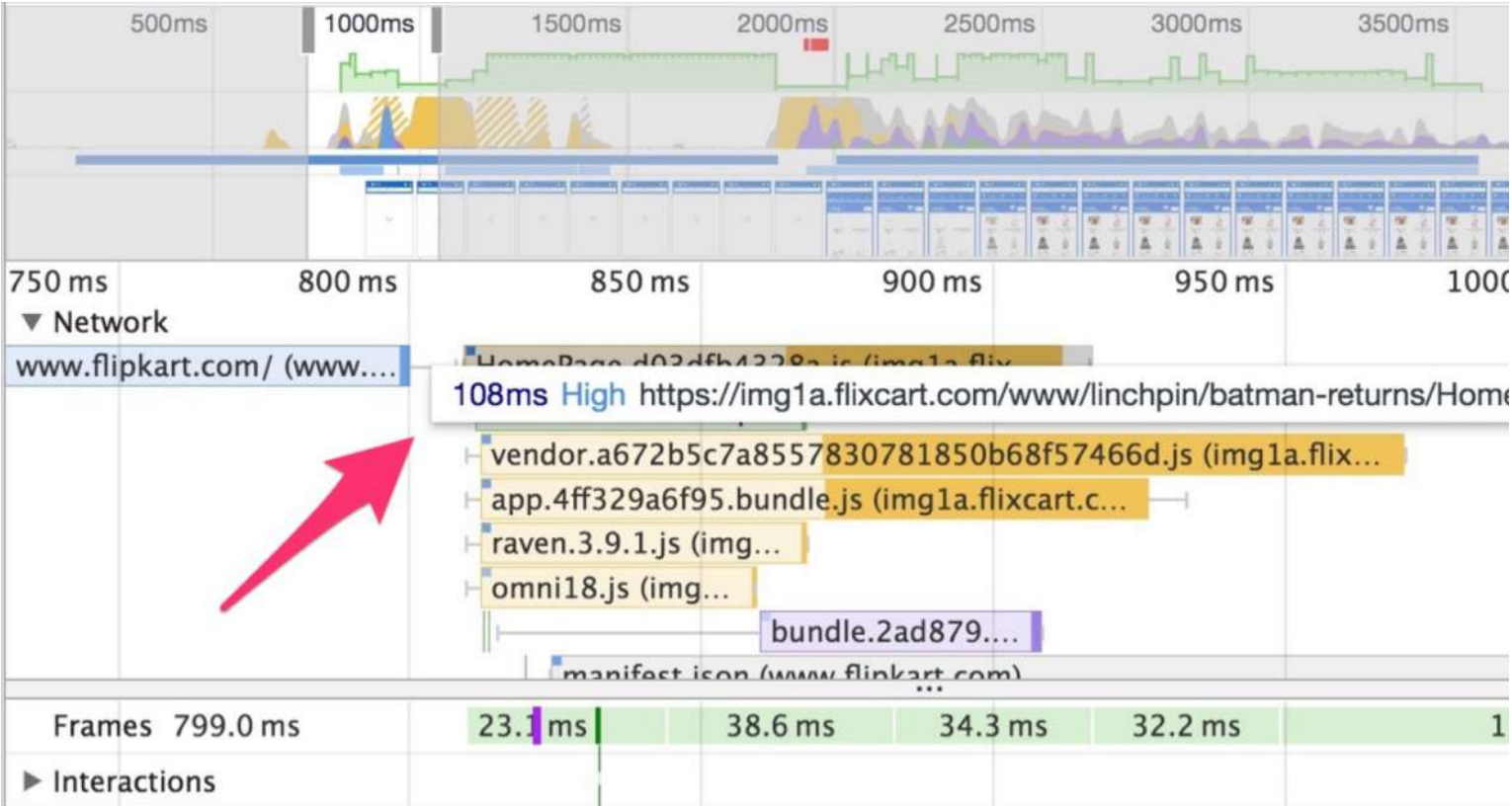
```
<link rel="preload" as="font" crossorigin="crossorigin" type="font/woff2" href=  
"https://cdn.shopify.com/shopify-marketing/assets/static/Brandon--light.woff2">  
<link rel="preload" as="font" crossorigin="crossorigin" type="font/woff2" href=  
"https://cdn.shopify.com/shopify-marketing/assets/static/Brandon--medium.woff2">  
<link rel="preload" as="font" crossorigin="crossorigin" type="font/woff2" href=  
"https://cdn.shopify.com/shopify-marketing/assets/static/Brandon--bold.woff2">
```



# Cesta od webu k PWA

- Ako zrýchliť načítavanie webu?

- rel=preload



# Cesta od webu k PWA



- **Ako zrýchliť načítavanie webu?**

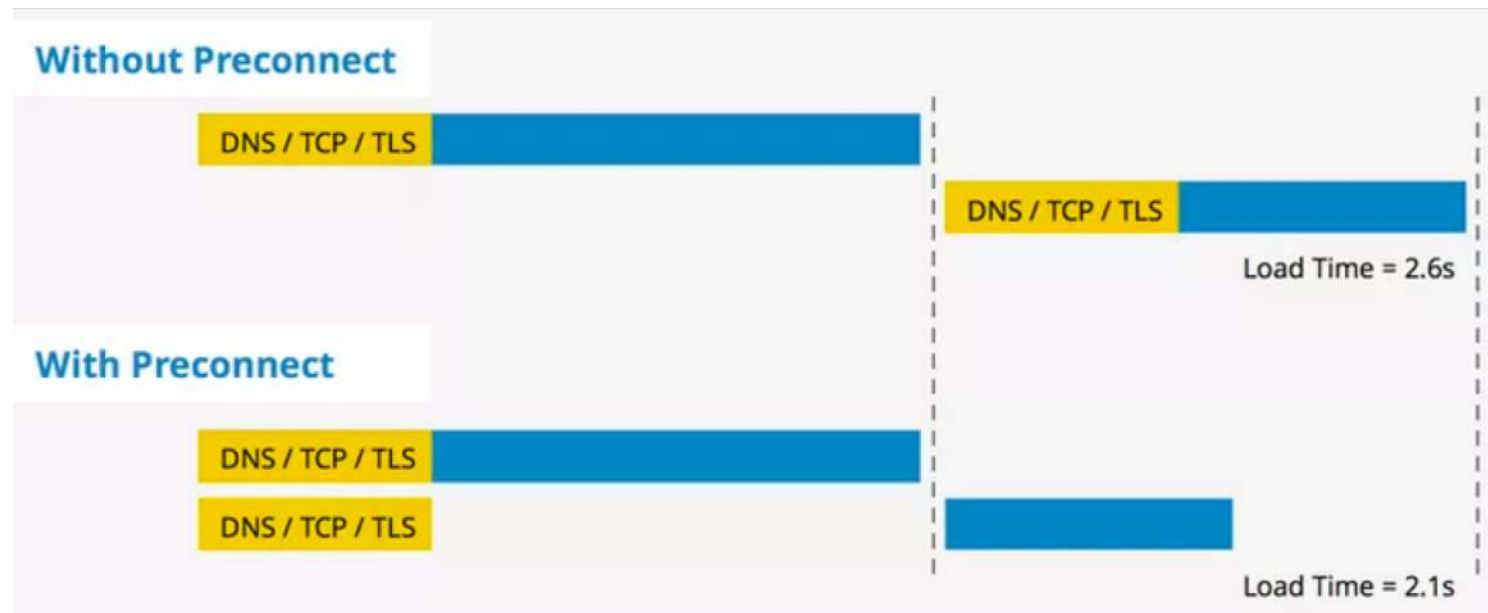
- rel=preconnect

- Poviete prehliadaču, aby nadviazal spojenie s konkrétnym webovým serverom predtým, než pošle samotnú požiadavku.

```
<link href="https://cdn.domain.com" rel="preconnect" crossorigin>
```

# Cesta od webu k PWA

- Ako zrýchliť načítavanie webu?
  - rel=preconnect



**Nie, Internet Explorer  
nepodporuje preconnect.**

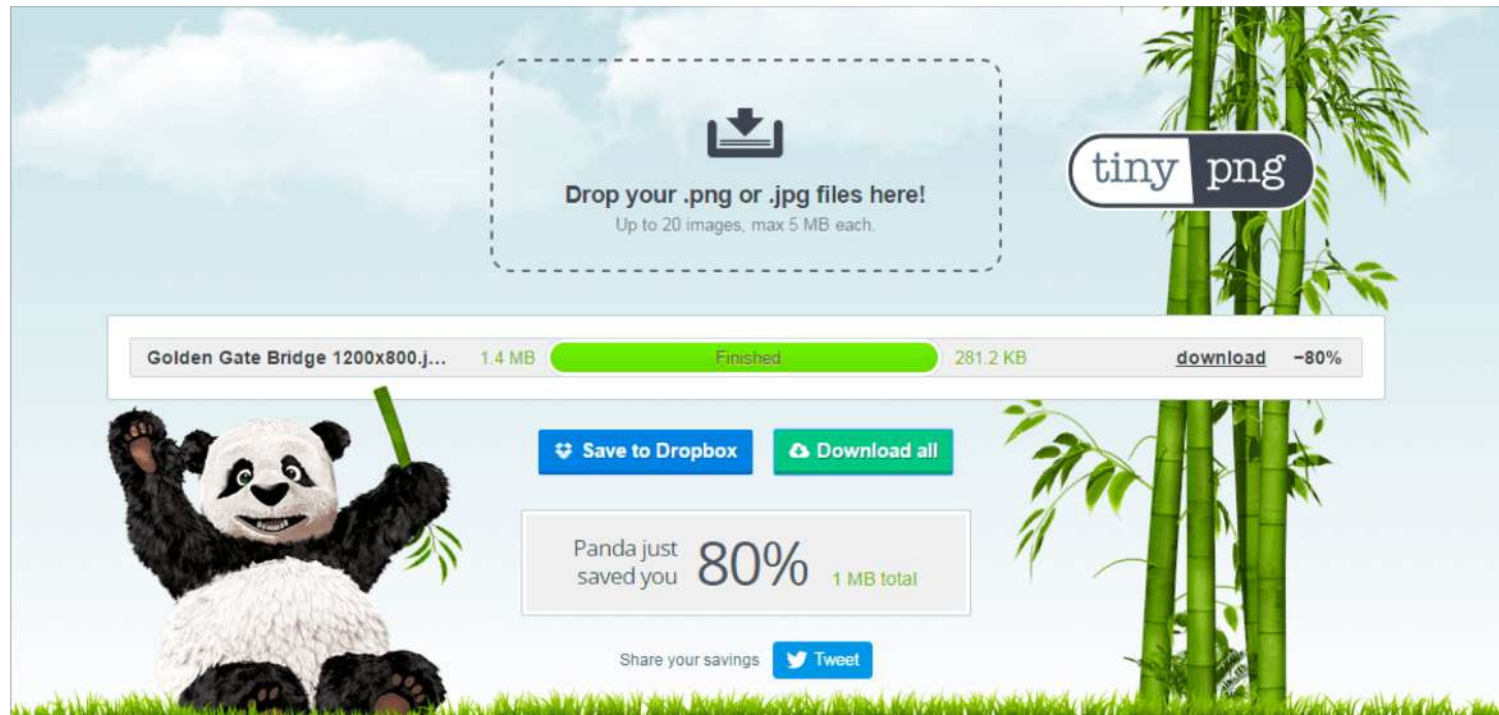
**A Opera Mini tiež nie.**

**MS Edge čiastočne.**



# Cesta od webu k PWA

- **Ako zrýchliť načítavanie webu?**
  - Minifikovať všetko – kódy, obrázky, uglifikáciu kódu



## Before

```
1 $(document).ready(function () {
2
3     // COLLAPSABLE WIDGETS (on Dashboard? & Profile pages)
4     // toggle widget box contents
5     $('a.toggle_box_contents').bind('click', toggleContent);
6
7     // WIDGET GALLERY EDIT PANEL
8     // Sortable widgets
9     var els = ['#leftcolumn_widgets', '#middlecolumn_widgets', '#rightcolumn_widgets', '#widget_picker_gallery' ];
10    var $els = $(els.toString());
11
12    $els.sortable({
13        items: '.draggable_widget',
14        handle: '.drag_handle',
15        forcePlaceholderSize: true,
16        placeholder: 'ui-state-highlight'.
```

## After

```
1 $(document).ready(function(){ $('a.toggle_box_contents').bind('click',toggleContent);var els=['#leftcolumn_widgets','#middlec
2 function elgg_slide_toggle(activateLink,parentElement,toggleElement){$(activateLink).closest(parentElement).find(toggleEleme
3 function outputWidgetList(forElement){return $("input[name='handler'], input[name='guid']",forElement).makeDelimitedList("va
4 jQuery.fn.makeDelimitedList=function(elementAttribute){var delimitedListArray=new Array();var listDelimiter="::";this.each(f
5 function widget_state(forWidget){var thisWidgetState=$.cookie(forWidget);if(thisWidgetState=='collapsed'){forWidget="#" +forW
6 var toggleContent=function(e){var targetContent=$('div.collapsable_box_content',this.parentNode.parentNode);if(targetContent
7 return false;};function widget_moreinfo(){$("img.more_info").hover(function(e){var widgetdescription=$("input[name='descript
8 else{$("#widget_moreinfo").css("top",(e.pageY+10)+"px").css("left",(e.pageX-210)+"px").fadeIn("medium");}},function(){$("#wi
9 var expires='';if(options.expires&&(typeof options.expires=='number' ||options.expires.toUTCString)){var date;if(typeof optio
10 expires='; expires='+date.toUTCString();}
11 var path=options.path?'; path='+ (options.path):'';var domain=options.domain?'; domain'+(options.domain):'';var secure=optio
12 return cookieValue;}};$ .fn.elgg_dropdownmenu=function(options){options=$.extend({speed:350},options || {});this.each(function(
13 function getActuator(ele){if(ele.nodeName.toLowerCase()=='ul'){return $(ele).parents('li')[0];}else{return ele;}}
14 function hide(){var subnav=getSubnav(this);if(!subnav)return;$.data(subnav,'cancelHide',false);setTimeout(function(){if(!$.d
15 function show(){var subnav=getSubnav(this);if(!subnav)return;$.data(subnav,'cancelHide',true);$(subnav).css({?Index: ?Index++
```

# Cesta od webu k PWA

- **Ako zrýchliť načítavanie webu?**
  - Responzívne obrázky

## CSS

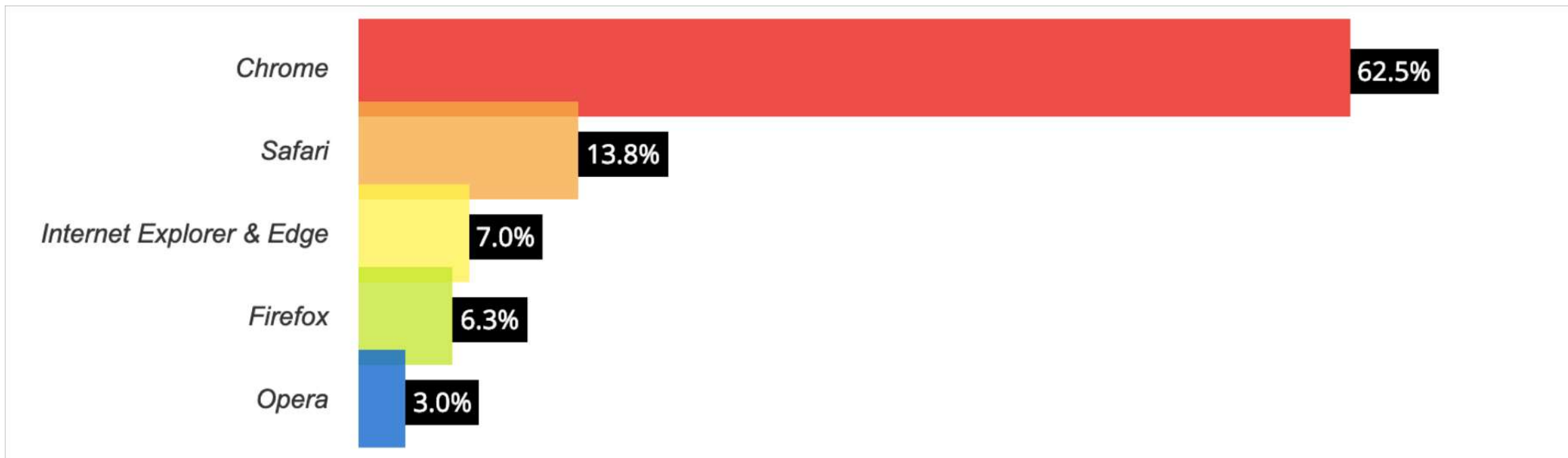
```
.img {  
  background-image: url(examples/images/image-384.jpg);  
}  
  
@media  
  (-webkit-min-device-pixel-ratio: 2),  
  (min-resolution: 192dpi) {  
  .img {  
    background-image: url(examples/images/image-768.jpg);  
  }  
}
```

**Podstatná vec! Podpora naprieč  
prehliadačmi ...**



# Cesta od webu k PWA

- Podiel používaných prehliadačov v roku 2018



# Cesta od webu k PWA

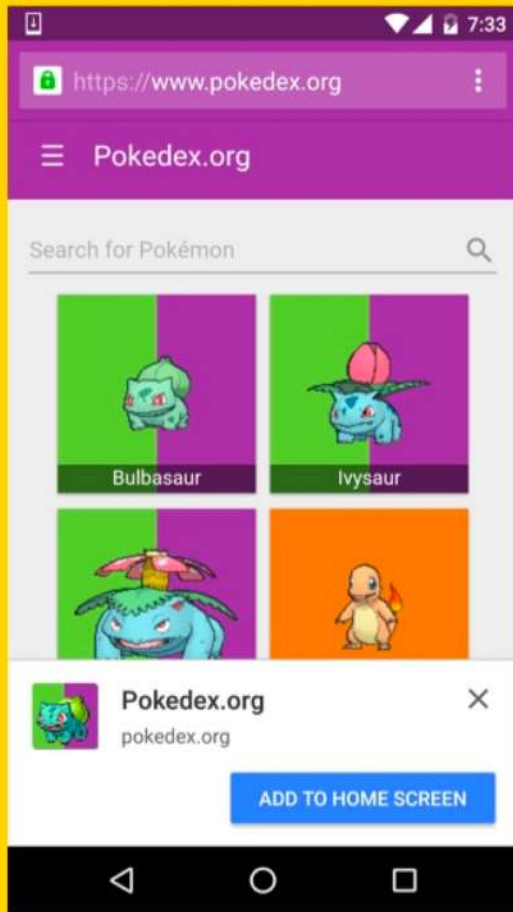


- **Ako otestovať podporu naprieč prehliadačmi?**
  - Najlepšie na konkrétnych zariadeniach
  - Super nástroj na testovanie podpory jednotlivých technológií:
    - **[caniuse.com](https://caniuse.com)**





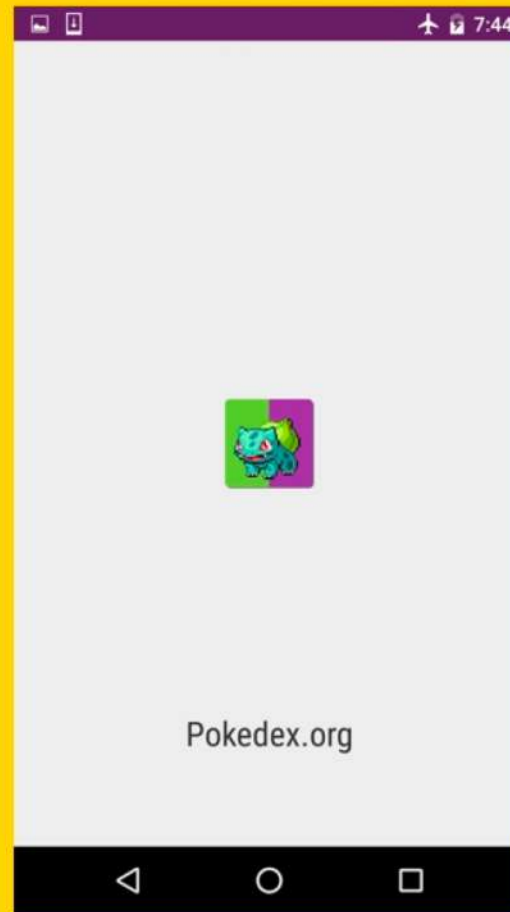
# Výsledok?



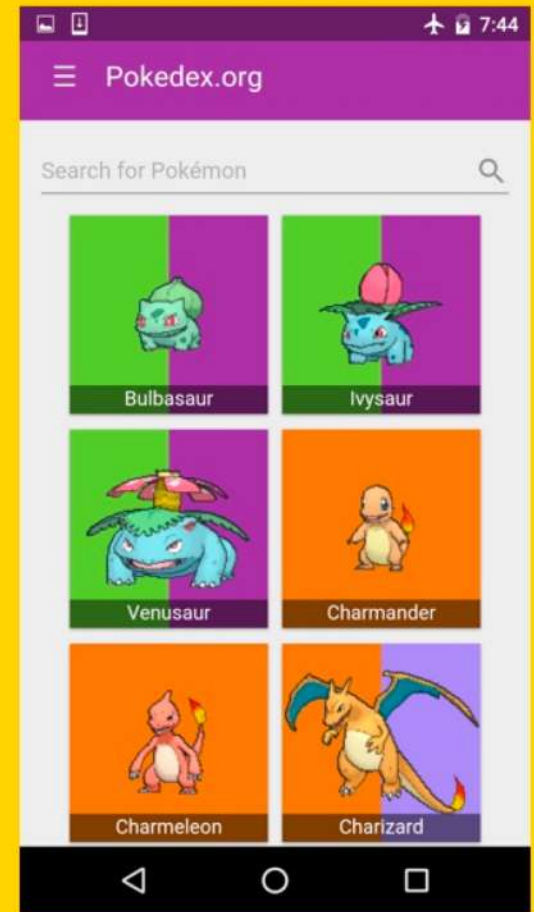
Web App install banner for engagement



Launch from user's home screen



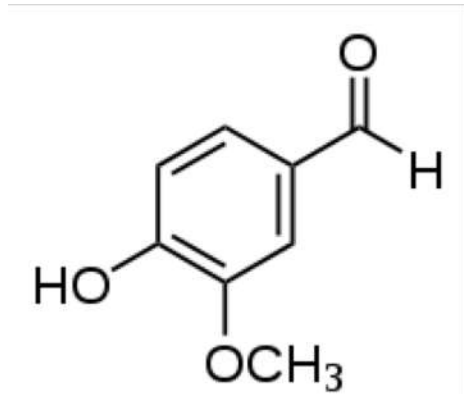
Splash screen (Chrome for Android 47+)



Works offline with Service Worker

# Existujúce frameworky?

# Frameworky pre PWA





# Frameworky pre PWA



**I Am Developer**  
@iamdeveloper

 Follow

“Maybe switching to [insert new JS framework] will compensate for my lack of actual JavaScript knowledge” - front-end developers in ~~2015~~. **2018**

RETWEETS  
**1,273**

FAVORITES  
**1,066**



2:07 PM - 14 Jun 2015



「(ツ)」

**Next step: Web -> Desktop**

# Čo majú spoločné tieto aplikácie?



Visual Studio Code



Kitematic



Slack

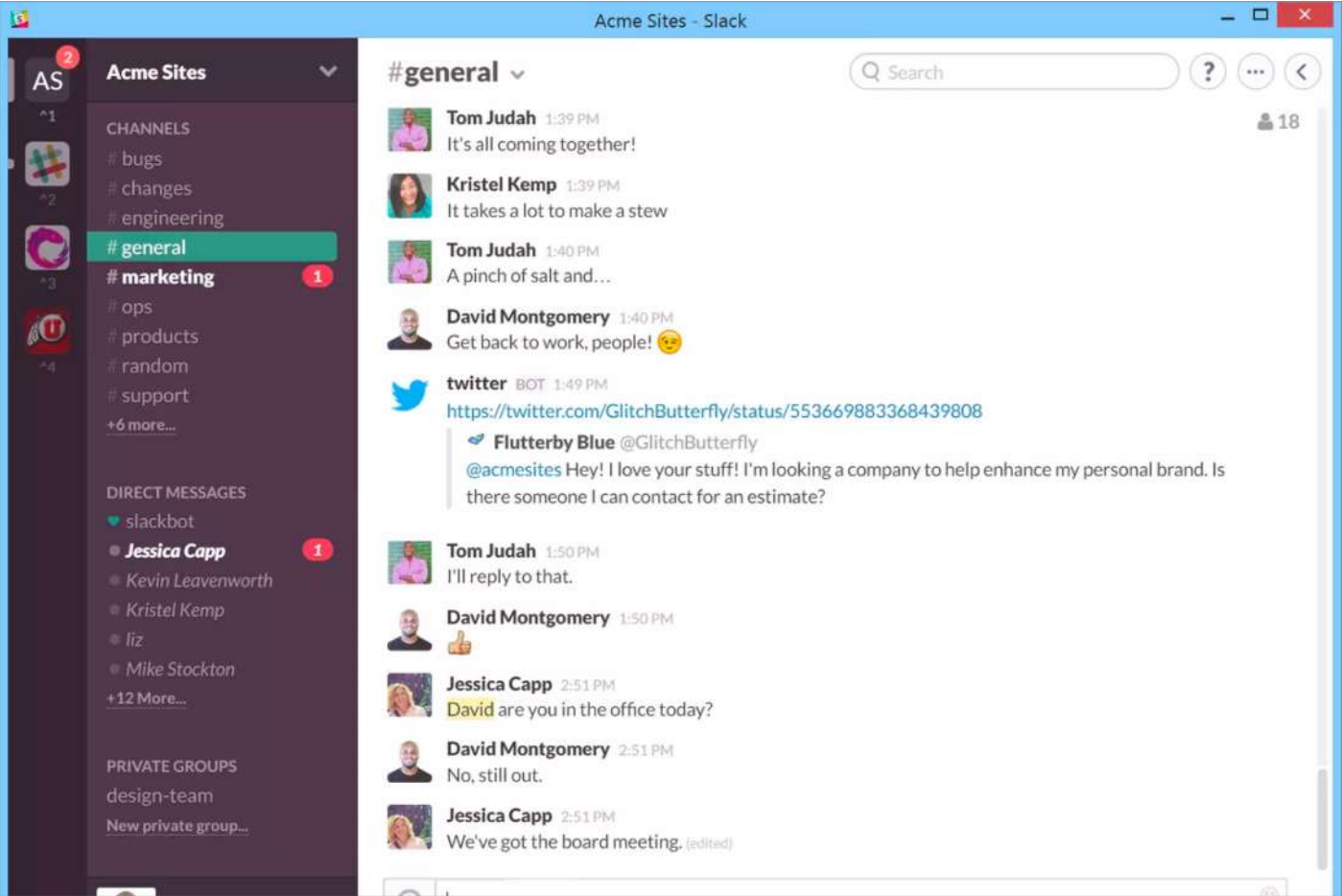


Atom



Skype

# Electron



# Electron



- Umožňuje spúšťať webové aplikácie v okne operačného systému
- Aplikácia sa tvári ako natívna
- Používateľovi viete dať ikonku na plochu

# Electron



- **Ako na to? (veľmi zjednodušené)**

```
your-app/  
├─ package.json  
├─ main.js  
└─ index.html
```

# Electron



- **Ako na to? (veľmi zjednodušené)**

```
{  
  "name": "your-app",  
  "version": "0.1.0",  
  "main": "main.js",  
  "scripts": {  
    "start": "node ."  
  }  
}
```

# Electron



- Ako na to? (veľmi zjednodušené)

```
const { app, BrowserWindow } = require('electron')

function createWindow () {
  // Create the browser window.
  let win = new BrowserWindow({ width: 800, height: 600 })

  // and load the index.html of the app.
  win.loadFile('index.html')
}

app.on('ready', createWindow)
```



# Electron



- **Ako na to? (veľmi zjednodušené)**

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Hello World!</title>
  </head>
  <body>
    <h1>Hello World!</h1>
    We are using node <script>document.write(process.versions.node)</script>,
    Chrome <script>document.write(process.versions.chrome)</script>,
    and Electron <script>document.write(process.versions.electron)</script>.
  </body>
</html>
```

# Electron



- **Ako na to?**

- Oficiálny tutorial:

- <https://electronjs.org/docs/tutorial/first-app>

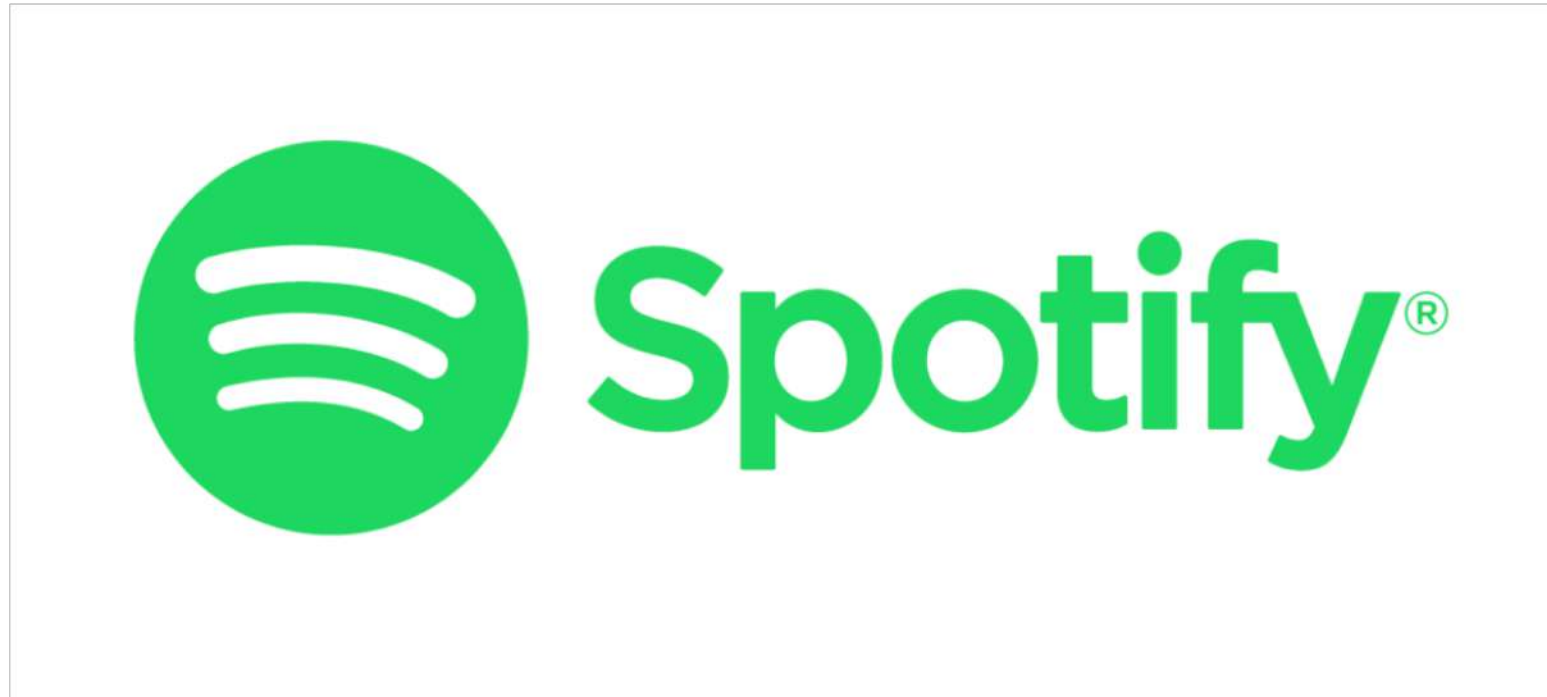
- <https://electronjs.org/docs/tutorial/quick-start>

# Chromium Embedded Framework



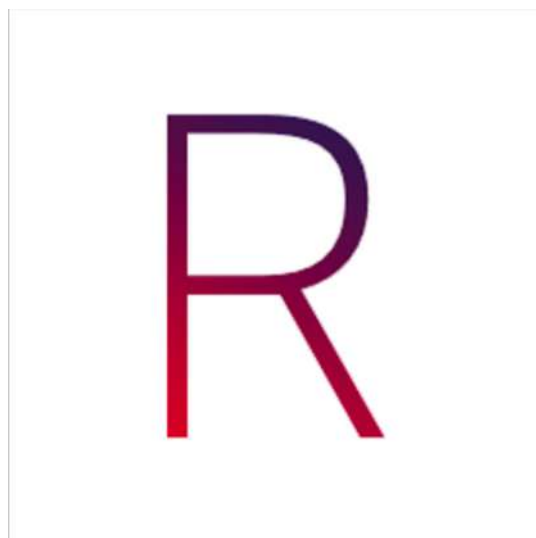
- „The Chromium Embedded Framework (CEF) is a simple framework for embedding Chromium-based browsers in other applications.”
  - Embedding an HTML5-compliant Web browser control in an existing native application.

# Chromium Embedded Framework



**Web -> Native mobile**

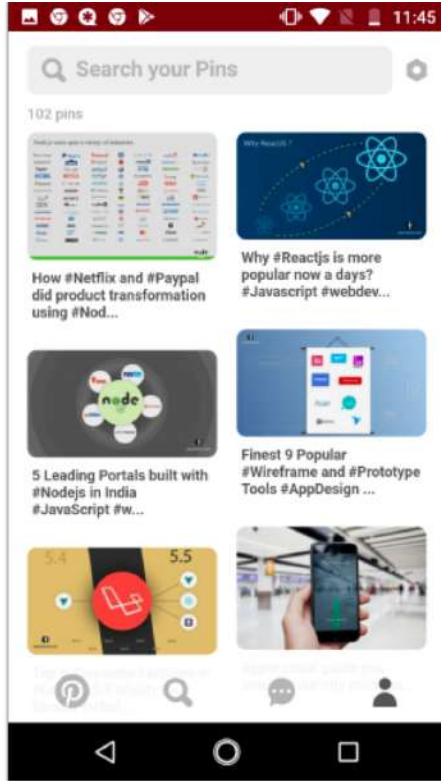
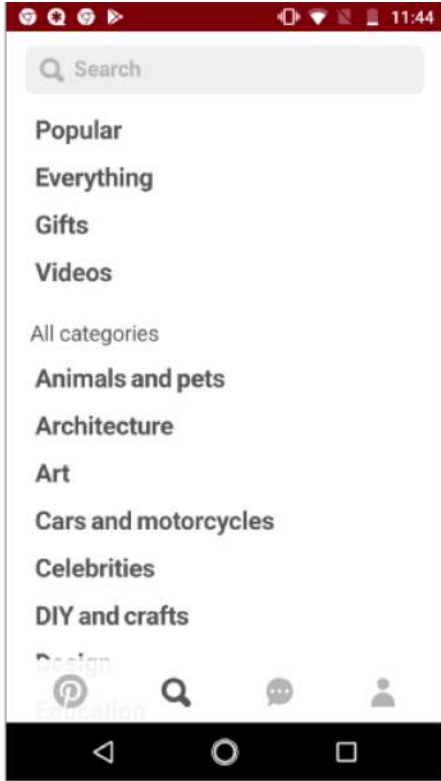
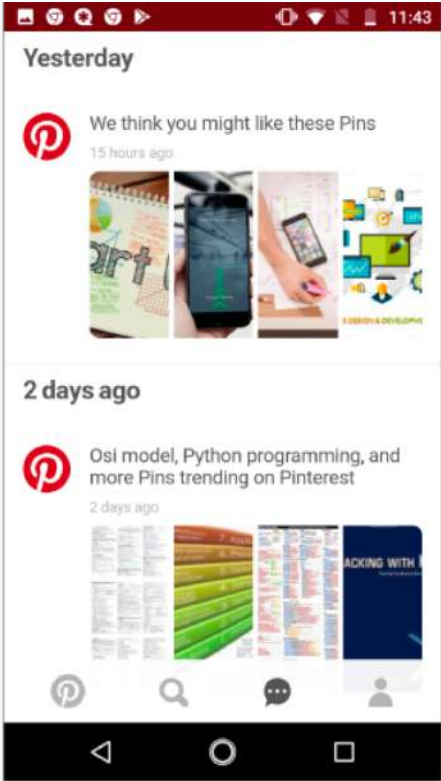
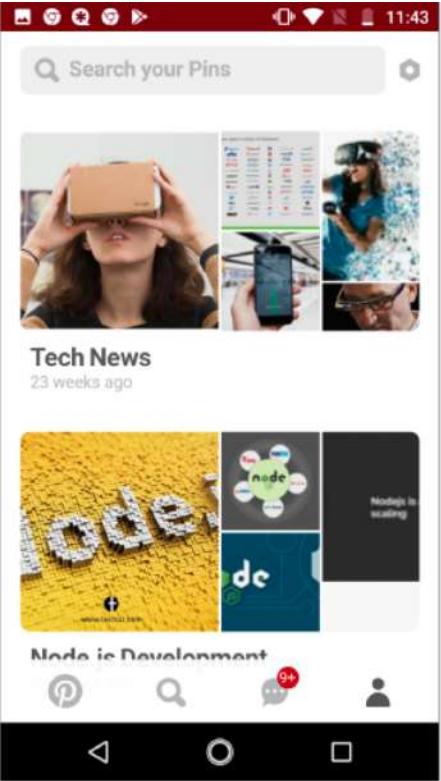
Mobilné Web -> Native frameworky



# Mobilné Web -> Native frameworky



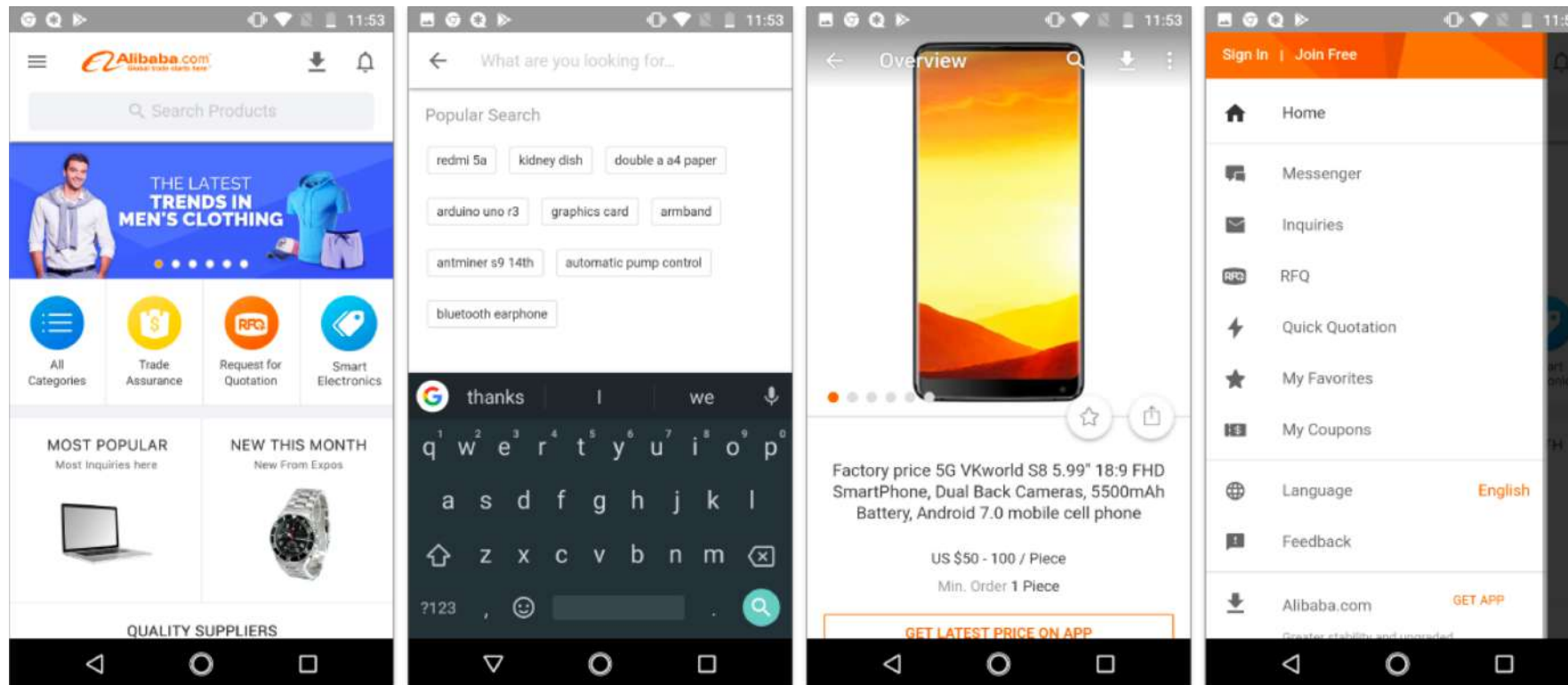
- **Pinterest**



# Mobilné Web -> Native frameworky



- **Alibaba / Aliexpress**





# Mobilné Web -> Native frameworky



- **Výhody**

- Stačí vedieť písať web (viac menej)
- Dá sa exportovať na obe platformy
- Dá sa ponúknuť cez Google Play / App Store
- Štandardný používateľ (BFU) nevie, že kliká web

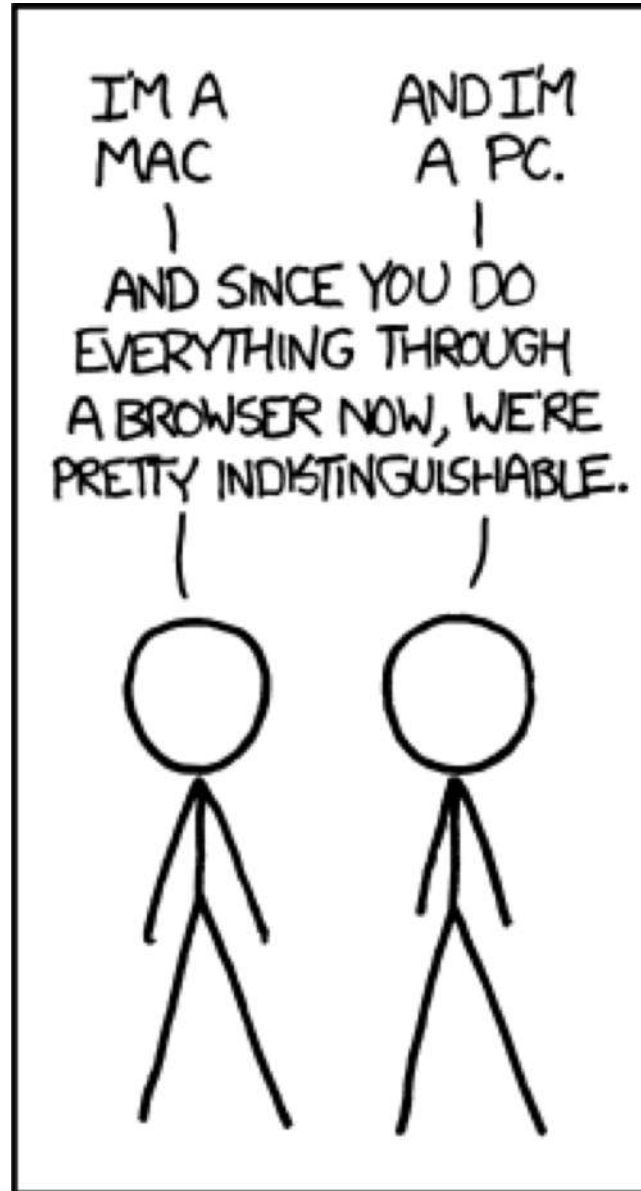
# Mobilné Web -> Native frameworky



- **Nevýhody**

- Výkon ... (spoločne aj pre desktop)
- Prispôsobiteľnosť aplikácie
- Úplný offline režim nedáva zmysel

# Zhrnutie



# Užitečné odkazy



- **PWA všeobecne**
  - <https://addyosmani.com/blog/getting-started-with-progressive-web-apps/>
  - <https://facebook.github.io/create-react-app/docs/making-a-progressive-web-app>
  - <https://developers.google.com/web/progressive-web-apps/checklist>
  - <https://developers.google.com/web/fundamentals/primers/service-workers/>

# Užitečné odkazy



- **Desktop**

- <https://www.qt.io/>
- <https://electronjs.org/>

# Užitečné odkazy



- **Mobil**

- <https://ionicframework.com/>
- <https://cordova.apache.org/>
- <https://framework7.io/>

# Bratislava OpenCamp 2019



## Zaujímajú ťa otvorené technológie?

Príď na Bratislava OpenCamp 2019 kde ťa čakajú prednášky a workshopy z oblasti otvorených technológií a otvorenej spoločnosti.

**Registrácia je bezplatná.**

**5 workshopov**

**25 prednášok**

HTTPS. Let's Encrypt. Python. Linux. Turrís. GDPR. Exploity. GIS. Bezpečnosť.



13. 4. 2019

[opencamp.sk](http://opencamp.sk)



# Bratislava OpenCamp 2019



**[opencamp.sk/program](https://opencamp.sk/program)**

**[opencamp.sk/registracia](https://opencamp.sk/registracia)**



# Ďakujem za pozornosť

