Progressive Web Apps

The web is ready, but what about the users?

Your Speaker

Loris Bachert

Trainer, Consultant, Developer
This talk is now available on every device...

... just kidding, that’s just what our users expect

Installing a native app is painful!

Open AppStore → Find app in AppStore → Click „Install“ → Accept permissions → Wait for the download... → Start the app

During each step an **average of 20%** of the users are lost

1000 ➔ 800 ➔ 640 ➔ 512 ➔ 409 ➔ 327
Visitors per month
(top 500 websites vs top 500 apps)

15.7 million vs 6 million

Source: comScore Mobile Metrix, U.S, 2017

Time spent per month
(top 500 websites vs top 500 apps)

11 minutes vs 187 minutes

Source: comScore Mobile Metrix, U.S, 2017
Talking about user expectations...
Users expect your app to...

...work while being online

Users expect your app to...

...work while commuting
Users expect your app to...

...work while traveling

...work at any time
Users expect your app to...

...inform about news

...show new things
Users expect your app to...

...look familiar

Users expect your app to...

...be in the app drawer
Lots of expectations for a native app!

You are offline.

Your device is offline.

Try:
- Turning off airplane mode
- Turning on mobile data or Wi-Fi
- Checking the signal in your area
The users need to be taught...

...about PWAs

Progressive Web Apps...

... are a modern type of web applications
... are combining the advantages of native and web
... are websites which fullfill user expectations of native apps
„Progressive Web Apps are just websites that took all the right vitamins.“
– Alex Russel
About Progress

Installability
App Install Banner

- Proactive information that the app can be installed
- Replacement for:

![Google Play](image)
![App Store](image)
Web App Manifest

- JSON file
- Contains metadata about the app
- Defines design and behaviour of the installed app

theme_color

icon

name

Facebook

background_color
Offline Availability

Service Worker
- Background thread
- Can be used as a proxy for network requests
- Can use browser cache and database
PWAs – What about the users?

Website

Webserver

Request data

Cache

Website

Webserver

Cache

return

service worker
PWAs – What about the users?

Webserver

Service Worker

.prepare service worker installation

Website

Cache

Webserver

request data

Website

Cache
PWAs – What about the users?

© Orientation in Objects GmbH

Website
Service Worker
Webserver

return data

Cache
cache
data

Website
Service Worker
Webserver

cache data

Cache
PWAs – What about the users?

© Orientation in Objects GmbH
PWAs – What about the users?

© Orientation in Objects GmbH

Service Worker Webserver

Cache

return data

Website

Webserver

Browser Tabs

Service Worker

Webserver

© Orientation in Objects GmbH
Tell users about...

...the network state

Tell users about...

...the offline availability
Background Sync
Background Sync

- Queue tasks while offline
- Execute tasks when online again
- New API, only supported in Chrome yet
Tell users about...

...the synchronization state

Tell users about...

...conflicts
Push Notifications

Diagram:
- **Push Server**
- **Web Server**
- **Service Worker**
- **Website**

Push Server:
- **register at push server**

Diagram illustrates the flow of push notifications from the website through the service worker to the push server and then to the web server.
PWAs – What about the users?

Push Server

create subscription and send it to server

Website

Service Worker

Web Server

Push Server

use subscription and send message to push server

Website

Service Worker

Web Server
Tell users about...

Hey

Hey
What’s up?
Want to go to the cinema?
I’m bringing another friend.
*friend

*friend

...push events
If you won’t tell them...

...you might get caught

User Interface
Designing a **PWA** is more like a **native app** than a **responsive website**

You should...

...use existing guidelines
You should...

…use default fonts
You should...

...keep your app smart
You should...

...use an app shell
You should…

...make your app fast

You should…

...avoid bouncing content
You should...

...use existing APIs
You should...

...use Google Lighthouse

Google Lighthouse

- Provides KPIs for different metrics
- Has a complete checklist how to be a PWA
- Helps you keep your app smart, clean and usable
To sum it up...

Be a teacher

− Teach users about (maybe) unexpected features
− Tell users why you need permissions
### Be available
- even while being offline
- Queue tasks done while being offline...
  ...and execute them when you are online again

### Be transparent
- Show connection state changes
- Inform about caching state
- Inform about data synchronisation
Be trustworthy

− Don’t abuse push notifications
− Notify the user about incoming push notifications
− Don’t spam!

Be familiar

− Use default components
− Use default fonts
− Use an app shell
**Be usable**
- Avoid bouncing content
- Keep the app small
- Start fast

**Success Story: Twitter**
- 75% more tweets
- Bounce rate reduced by 20%
- Only 3% as big as the native Android app

https://pwa.bar/
Success Story: AliExpress

- Increased conversion rate by 104%
- Users visit twice as much pages as before

https://pwa.bar/

Success Story: Tinder

- Startup time from 11.91 to 4.69 seconds reduced
- 90% smaller than the native Android App
- Users stay longer

https://pwa.bar/
Success Story: Uber

- Startup time in less than 3 seconds on a 2G network
- Only 50 kB

https://pwa.bar/

Let’s...

...talk about the future
„By 2020, Progressive Web Apps will have replaced 50% of general-purpose consumer facing apps.“
– Gartner Research

Users will...

...be even more tired of apps
Users will...

A good PWA

...learn to love it

Everyone...

...is on board
Questions?

@LorisBachert

Thank you for your attention!
Further reading

- https://techbeacon.com/5-ways-ensure-great-ux-progressive-web-apps
- https://pwa.bar/
- https://caniuse.com/#search=api
- https://jakearchibald.github.io/isserviceworkerready/
- https://material.io/design/
- https://console.cloud.google.com/apis/library

Image Sources

- #60
  - Left: https://www.android.com/
  - Middle: https://material.io/
  - Right: https://angular.io/
- #62
  - Left: https://www.heise.de/
  - Middle: https://t3n.de/
  - Right: https://deanhume.github.io/beer/
- #67: https://developers.google.com/pay/api/android/
- #83 - #86: https://pwa.bar/
- #87:
  - https://commons.wikimedia.org/wiki/File:Safari_browser_logo.svg

All other pictures inside this presentation originate from https://pixabay.com and are under CC0 Creative Commons license or were created by myself.