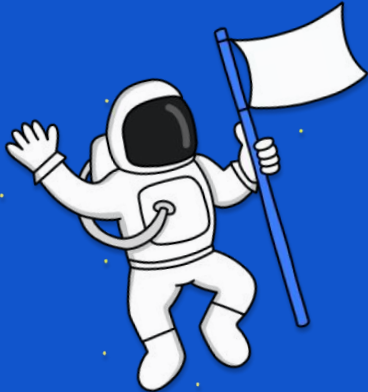


Introduction to Realtime Web Applications

Day 2 - 22 January 2021



Who are we



Low En
(@lowenhere)

SUTD ISTD Senior

Does software tings



Daryll Wong
(@daryllman)

SUTD ISTD Senior

Does *tings*

Logistics

Setup

~~For Day 1, make sure you have the following installed on your laptop:~~

- ~~- Git~~
- ~~- Node.js, along with a package manager (npm or yarn)~~
- ~~- A text editor / JavaScript IDE that you are comfortable with~~

For Day 2, you will need the following:

- A heroku account, and the heroku CLI installed
- Docker (optional)

Course Resources

Course Webpage (along with notes)

<https://realtime-apps-iap.github.io>

Github Repository

<https://github.com/realtime-apps-iap>

Other Resources / References

Here are a few resources that have been heavily referenced in the development of the workshop, mostly because they're really comprehensive and all around great:

Mozilla Developer Network (MDN)

<https://developer.mozilla.org/>

fireship.io YouTube channel

<https://www.youtube.com/channel/UCsBjURrPoezykLs9EqgamOA>

Agenda for Day 2

Introduction

Web APIs

Tutorial: Web API Experiments

Tutorial: GPS Location Sharing Application

Introduction to Progressive Web Applications (PWA)

Tutorial: Creating a PWA

Tutorial: Deploying projects on Heroku

Introduction to Docker

Q&A

Introduction

Introduction to Realtime Web Applications

The real-time web is a network web using technologies and practices that enable users to receive information as soon as it is published by its authors, rather than requiring that they or their software check a source periodically for updates.

-Wikipedia, Real-time Web

Web APIs

<https://realtime-apps-iap.github.io/docs/browserapi/main>

Tutorial: Web API Playground

<https://github.com/realtime-apps-iap/video-call-app>

Tutorial: GPS Location-Sharing Application

Introduction to Progressive Web Applications (PWA)

Progressive Web Applications (PWA)

Let's see what are some key phrases and sentences others use to describe a PWA:

(Taken from multiple online sources)

- built and enhanced with **modern APIs** to deliver enhanced **capabilities, reliability, and installability** while reaching anyone, anywhere, on any device with a **single codebase**.
- designed so they are **capable, reliable, and installable**
- feels like a **platform-specific** application
- intended to **work on any platform** that uses a standards-compliant browser, including both desktop and mobile devices
- a type of application software delivered through the web, built using **common web technologies** including HTML, CSS and JavaScript
- provide a **better online experience** for mobile and **eliminate the need to download apps** from an app store
- the concept of creating a flexible, **adaptable app** using **only web technologies**
- web apps developed using a number of specific technologies and standard patterns to allow them to **take advantage of both web and native app features**

Progressive Web Applications (PWA)

Simply put,

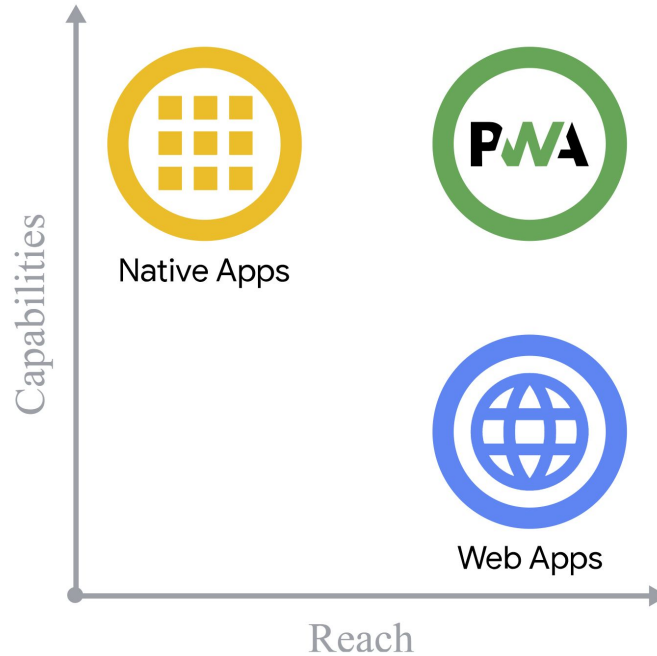
*PWAs are **websites** that provide a **native app experience** on a mobile phone*

Btw, what does ‘progressive’ mean:

- having a progressive enhancement strategy
- “as the user progressively builds a relationship with the app over time, it becomes more and more powerful” according to Pete LePage – Google Developer Advocate



Best of Both Worlds?



Yes and No. Let's be critical and keep at the back of our minds as we go through this lesson: Why should we develop native apps or PWA or websites?

Characteristics of a Native Mobile Application

- **RELIABLE**
 - Loads instantly and responsively
- **FAST**
 - Responds quickly with smooth applications
- **Engaging**
 - Good UIUX that resembles that of an app rather than a website

(If a PWA has these characteristics, does it mean it *behaves like a native application*?)

PWA Checklist

1. Starts Fast, Stays Fast
2. Works in Any Browser
3. Responsive to Any Screen Size
4. Provides a Custom Offline Page
5. Is Installable
6. Provides an Offline Experience
7. Is Fully Accessible
8. Can be discovered through Search
9. Provides context for permission requests

[PWA Standards by Google](#)

PWA Checklist

1. Starts Fast, Stays Fast

- a. Fast loading of the websites and all components within. Some metrics will be measured by Lighthouse. Optimised for user-centric performance metrics.
Increases User Retention.

2. Works in Any Browser

- a. The website should still work properly on all kinds of browsers, across a spectrum of devices and browsers, even with older browsers.

3. Responsive to Any Screen Size

- a. The layout of a website on a browser and mobile phone should be quite different. It should look more like an app when viewed on a mobile phone.

PWA Checklist

4. Provides a Custom Offline Page

- a. When there is no connection, it should not show the typical ‘No internet’ dinosaur page. It should show a cached custom offline page.

5. Is Installable

- a. Users are able to ‘install’ your PWA — afterwhich the PWA will have an App Icon on the home screen of their phones/computers, and seems like any other native app.

6. Provides an Offline Experience

- a. User should be able to navigate and interact with the application even without connection.

PWA Checklist

7. Is Fully Accessible

- a. Pass WCAG2.0 accessibility requirements - standards for creating accessible websites

8. Can be discovered through search

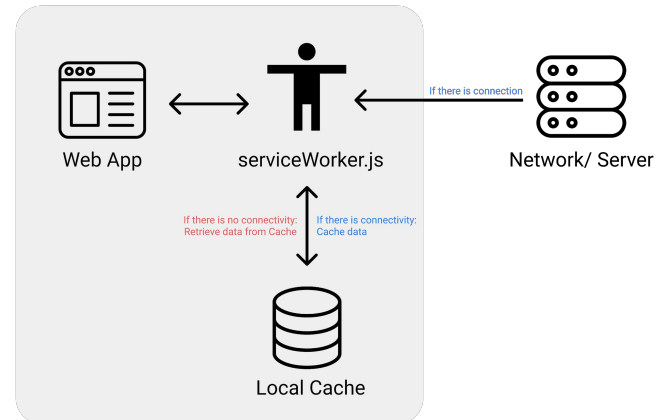
- a. Ability for the PWA website to be discovered organically through search engines e.g. Google. SEO discoverable.

9. Provides context for permission requests

- a. If you do have to use APIs (like push notifications, geolocation, credentials), it is important to ask for permission with additional context explaining the use of it.

serviceWorker.js

- You design this worker script to decide what to show when there is connectivity or not, etc.
 - E.g. If there is no network, maybe we could return cached web pages
 -
- This is a 'Butler' that helps you retrieve the information that you need - though this is something you code to define its behaviour.



manifest.json

```
{
  "short_name": "Weather",
  "name": "Weather: Do I need an umbrella?",
  "description": "Weather forecast information",
  "icons": [
    {
      "src": "/images/icons-192.png",
      "type": "image/png",
      "sizes": "192x192"
    },
    {
      "src": "/images/icons-512.png",
      "type": "image/png",
      "sizes": "512x512"
    }
  ],
  "start_url": "?source=pwa",
  "background_color": "#3367D6",
  "display": "standalone",
  "scope": "/",
  "theme_color": "#3367D6",
  "shortcuts": [
    {
      "name": "How's weather today?",
      "short_name": "Today",
      "description": "View weather information for today",
      "url": "/today?source=pwa",
      "icons": [{ "src": "/images/today.png", "sizes": "192x192" }]
    },
    {
      "name": "How's weather tomorrow?",
      "short_name": "Tomorrow",
      "description": "View weather information for tomorrow",
      "url": "/tomorrow?source=pwa",
      "icons": [{ "src": "/images/tomorrow.png", "sizes": "192x192" }]
    }
  ]
}
```

- Basic information needed to create a PWA
 - Name
 - Description
 - Icons
 - Background colors
 - Shortcuts
 - Etc.
- Fulfill the basic requirements for this to be detected as a PWA by your browser.
- Tip: Use Google Lighthouse

Tutorial: Creating a React PWA

<https://github.com/realtime-apps-iap/react-pwa>

Tutorial: Deploying applications to Heroku

<https://github.com/realtime-apps-iap/mushroom-classifier-app>

Brief Introduction to Docker

Course Feedback



Competition Signups



More Information at:
<https://realtime-apps-iap.github.io/competition>

Main Course Website

<https://realtime-apps-iap.github.io>