Angular

Part 1
Mobile Apps!

- Introduction
- Why hybrid application?
- Some drawbacks
- Architecture Angular/IONIC, NodeJS & MongoDB

Part 1 - Mobile Apps

Introduction

- Exists three types of mobile apps
  - Native application
  - **Hybrid application**
  - Progressive Web Application (PWA)

Native applications

Part 1 - Mobile Apps

Introduction - Native application

What is it?

- Specific to a platform or/and operating system (os)
- Usually written in languages that the platform accepts
  - Swift or Objective C for **iOS** applications,
  - Java for Android applications,
  - C# or VB.NET for **Windows** applications.
- Built using the specific IDE
  - Android Studio,
  - XCode,
  - etc?

Part 1 - Mobile Apps

Introduction - Native application

Some examples

- Instagram,
- Facebook,
- VLC media player,
- 2048 game,
- etc.

Part 1 - Mobile Apps

Introduction - Native application

Pros

- Very fast & response as built for the specific platform
- They have the best performances
- They are more smooth to use and give a better user experience

Cons

- More expensive to develop
- One application per platform
- Hard to maintain and push through all specific stores

Hybrid applications

Part 1 - Mobile Apps

Introduction - Hybrid application

What is it?

- Developed to be used across multiple platforms
- Usually written in HTML, CSS & JavaScript
- Target a WebView (not a browser) to allows us to use hardware capabilities
- Most of hybrid app leverage **Apache Cordova**

Part 1 - Mobile Apps

Introduction - Hybrid application

Popular Framework

- **Ionic**,
- PhoneGap,
- Sencha touch,
- Framework7,
- etc.

Some examples

- MarketWatch,
- Untappd,
- Fanreact,
- Tripcase,
- etc.

Part 1 - Mobile Apps

Introduction - Native application

Pros

- Adaptable: one code for multiple platforms
- Reduce developing and deployment time

Cons

- Slower app as the hybrid framework act like a bridge
- Applications with heavy animations and sound effects doesn’t feet well

Progressive web applications (PWAs)

Part 1 - Mobile Apps

Introduction - Progressive web application

What is it?

- Web app using latest web APIs to act like an app without install the app
- Usually written in HTML, CSS & JavaScript
- Access to the app through a browser using the URL

Part 1 - Mobile Apps

Introduction - Progressive web application

Some examples

- AliExpress,
- Financial Times,
- Nasa,
- PayStack,
- etc.

Part 1 - Mobile Apps

Introduction - Progressive web application

Pros

- Easy to build
- Easy to maintain
- Available offline and perform well on low-quality network
- Build one app for all platform

Cons

- Need a browser to run
- Less interactive than intuitive than native applications

Part 1 - Mobile Apps

Why hybrid apps

- Single code base for all platforms,
- Same development team can deliver the app for any platform,
- Hybrid app are based on web technologies, so it can be easily turn into a Progressive Web App,
- Can do the same hardware-based performance acceleration than native apps,
- Can ensure same and consistent user experience across all platforms

Part 1 - Mobile Apps

Some drawbacks

- For most application, performance is same as native app but 3D, HD games, etc might not go well
- Hardware capabilities depends on plugin to make them available
- Dependencies on a framework have to be in sync with latest platform version changes and releases

Part 1 - Mobile Apps

Architecture Angular/IONIC, NodeJS & MongoDB

For this course

- **Cordova:** open-source mobile development framework. Cordova plugins helps to access device features.
- **Ionic:** app platform for web developers. It’s based on the framework Angular and allow to use all Angular ecosystem.

Part 1 - Mobile Apps

Architecture Angular/IONIC, NodeJS & MongoDB