

How to build a mobile app with no time, experience but at least knowing React

Robert Orliński

- **Front-end Developer**
- **Blogger**
- **Activist**
- **Love travelling, reading, and listening to audiobooks**

robertorlinski.pl



Extras:

robertorlinski.pl/mobile-apps



Close your eyes...

Don't peek!



**There is only 1
small problem**

Or maybe 3

1. You have to deliver it in less than 2 months (it would be hard even in the case of web app).

- 1. You have to deliver it in less than 2 months (it would be hard even in the case of web app).**
- 2. You have never built any mobile app and you don't even know where to start.**

- 1. You have to deliver it in less than 2 months (it would be hard even in the case of web app).**
- 2. You have never built any mobile app and you don't even know where to start.**
- 3. There is no one who can help you.**

What do you do?

You have 4 options:

- **2 useless options***

- **2 useless options***
- **2 useful options***

- **2 useless options***
- **2 useful options***

*** assuming you have 2 months to build an app.**

First "useless" option:

First "useless" option:

Building 2 native apps:

- for iOS
- and for Android



 **Kotlin**

The best performance
The biggest control



The best performance

The biggest control

**There is no way to learn needed
technologies and build the app in
2 months**

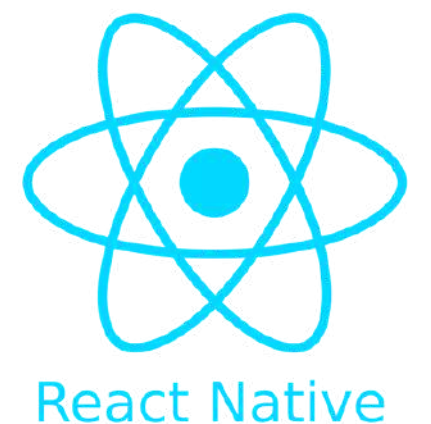


 **Kotlin**

Second "useless" option:

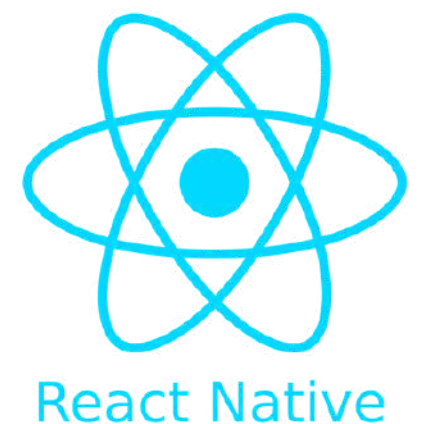
Second "useless" option:

Building 1 compiled app



1 codebase

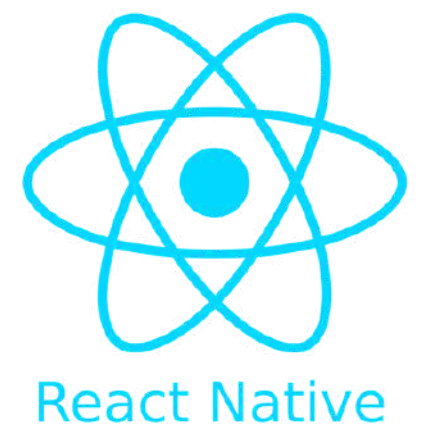
**Code syntax very familiar to
front-end developers**



1 codebase

Code syntax very familiar to front-end developers

Still it will be hard to build the mobile app in 2 months using these technologies



First "useful" option:

First "useful" option:

Building 1 hybrid app in Capacitor + Ionic



1 codebase

React syntax

(or Angular/Vue/vanilla JS)

**Useful plugins and components
made for mobile apps**



1 codebase

React syntax...

Useful plugins and components...

**Not much to learn and create on
your own. You only need to
assemble your mobile app**



Second "useful" option:*

Second "useful" option:*

***but probably not the option in general..**

Second "useful" option:*

***but probably not the option in general..**

**Progressive Web App
(PWA)**



Any front-end syntax you want
Very easy to make and deploy
If you can make web app in
time, you can also make PWA



**Any front-end syntax you want
Very easy to make and deploy
If you can make web app in
time, you can also make PWA**

**It can't be downloaded from App
Store and Google Play**



**So this is probably not something
your customer/PM wanted**



**Hybrid looks like the
best option for our
needs**

**Then how can we
make a hybrid
mobile app?**

**As I mentioned, we
can use Capacitor and
Ionic for this job**





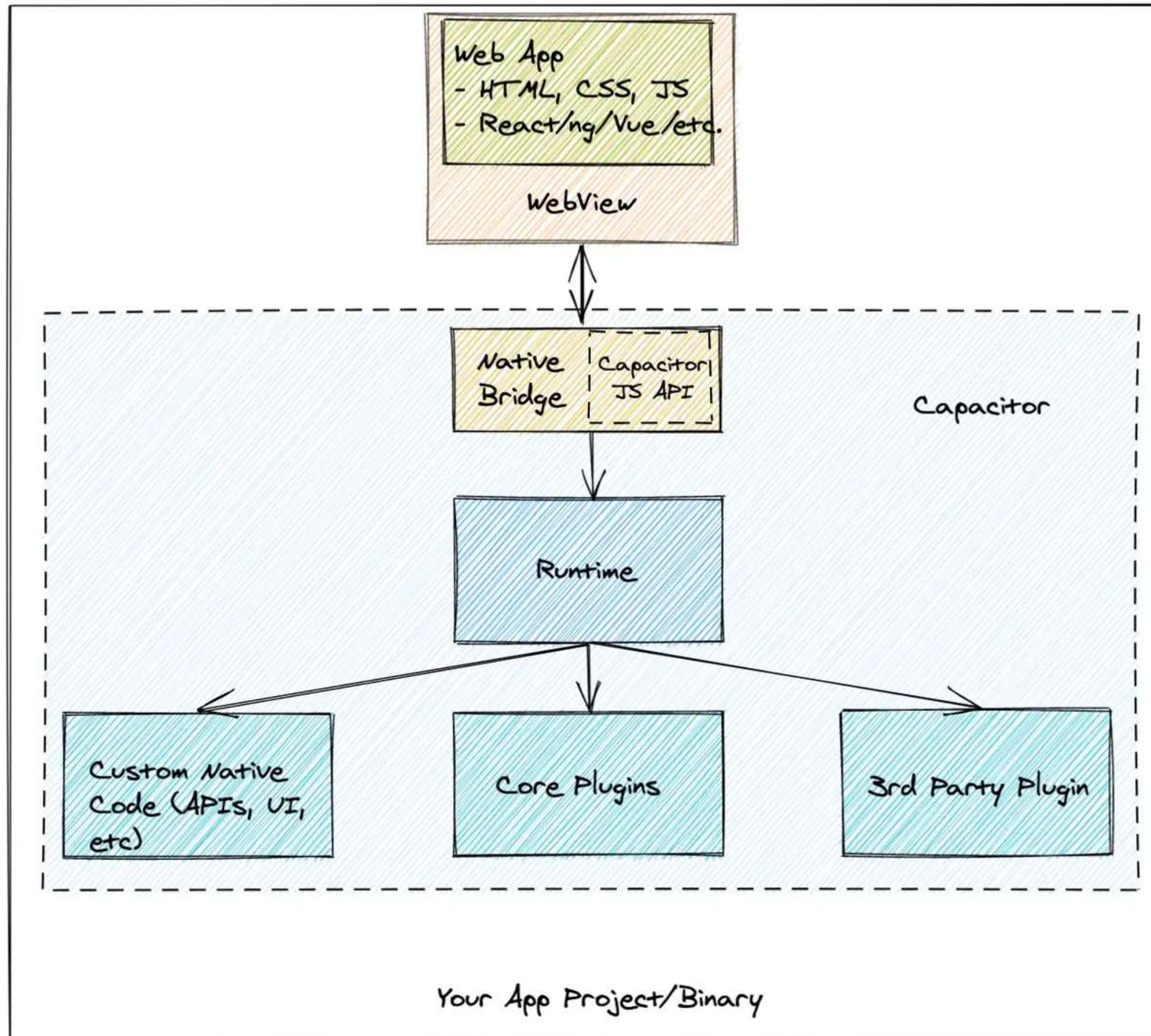
- **Runs our web app "inside of" the native app.**



- **Runs our web app "inside of" the native app.**
- **Provides plugins for handling native functionality and let us write some native code if we need it.**

[More: How Capacitor Works](#)





Source:

CapacitorJS: Turn Your Web App into a Mobile App







- **Provides routing, starting templates, theming, styles, etc...**



- **Provides routing, starting templates, theming, styles, etc...**
- **Works with React, Angular, Vue and vanilla JS**

More: [Ionic docs](#)

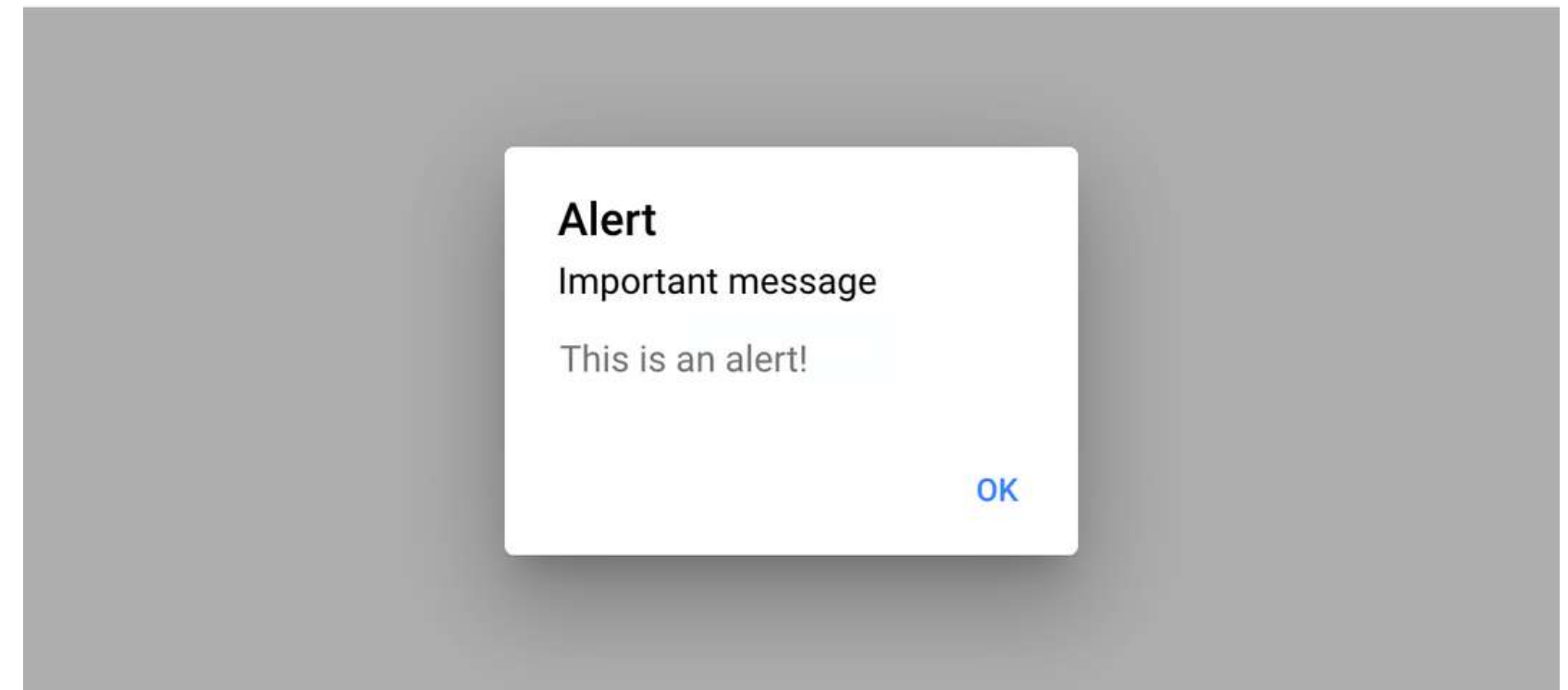
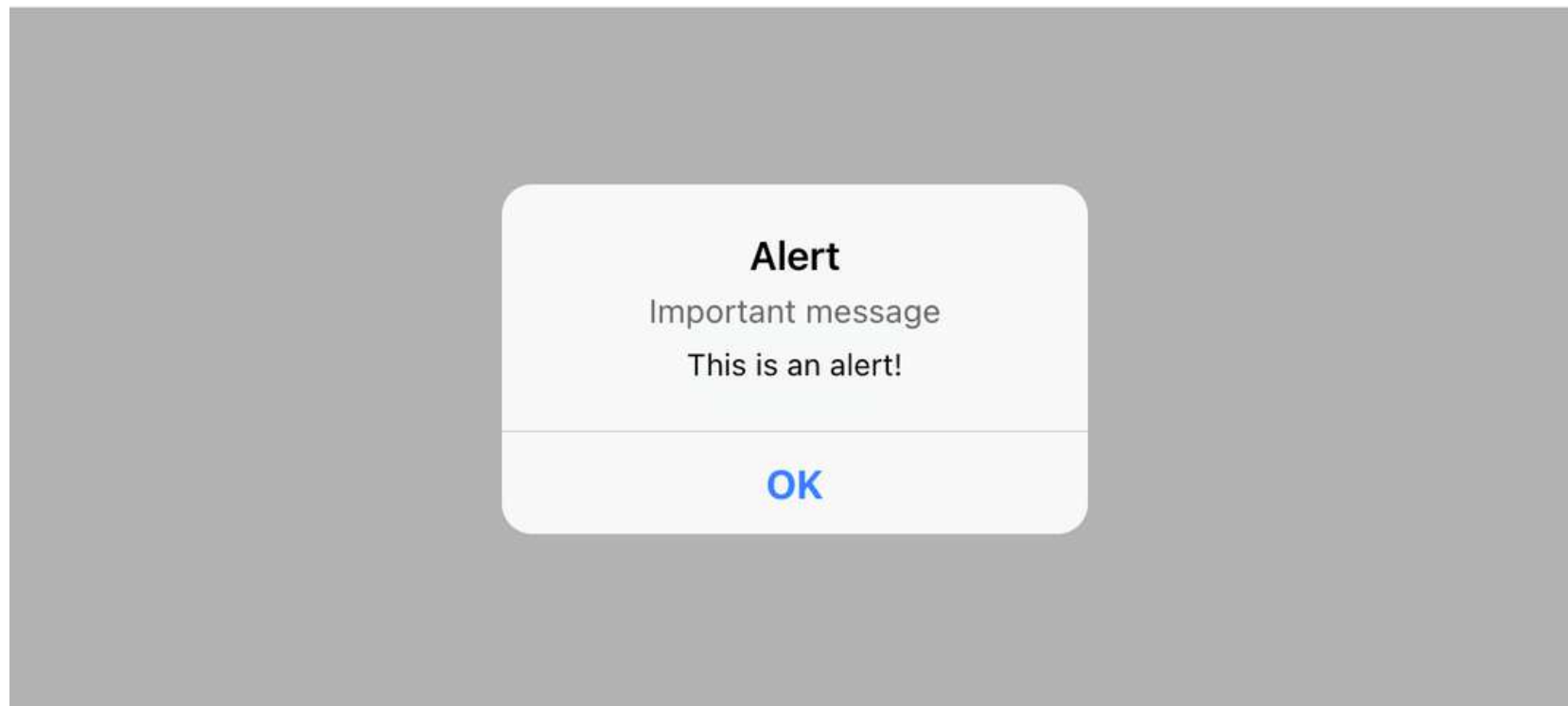




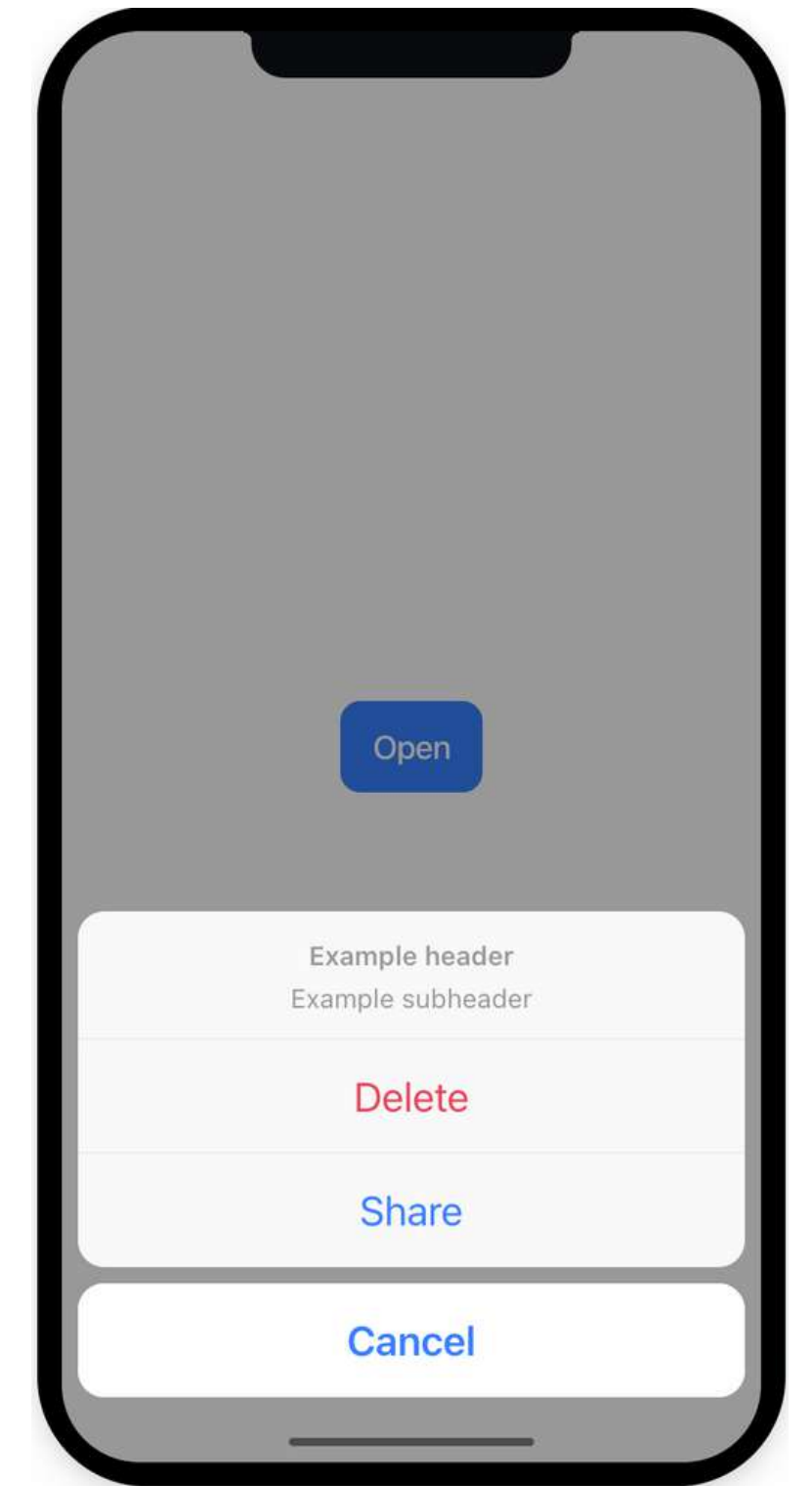
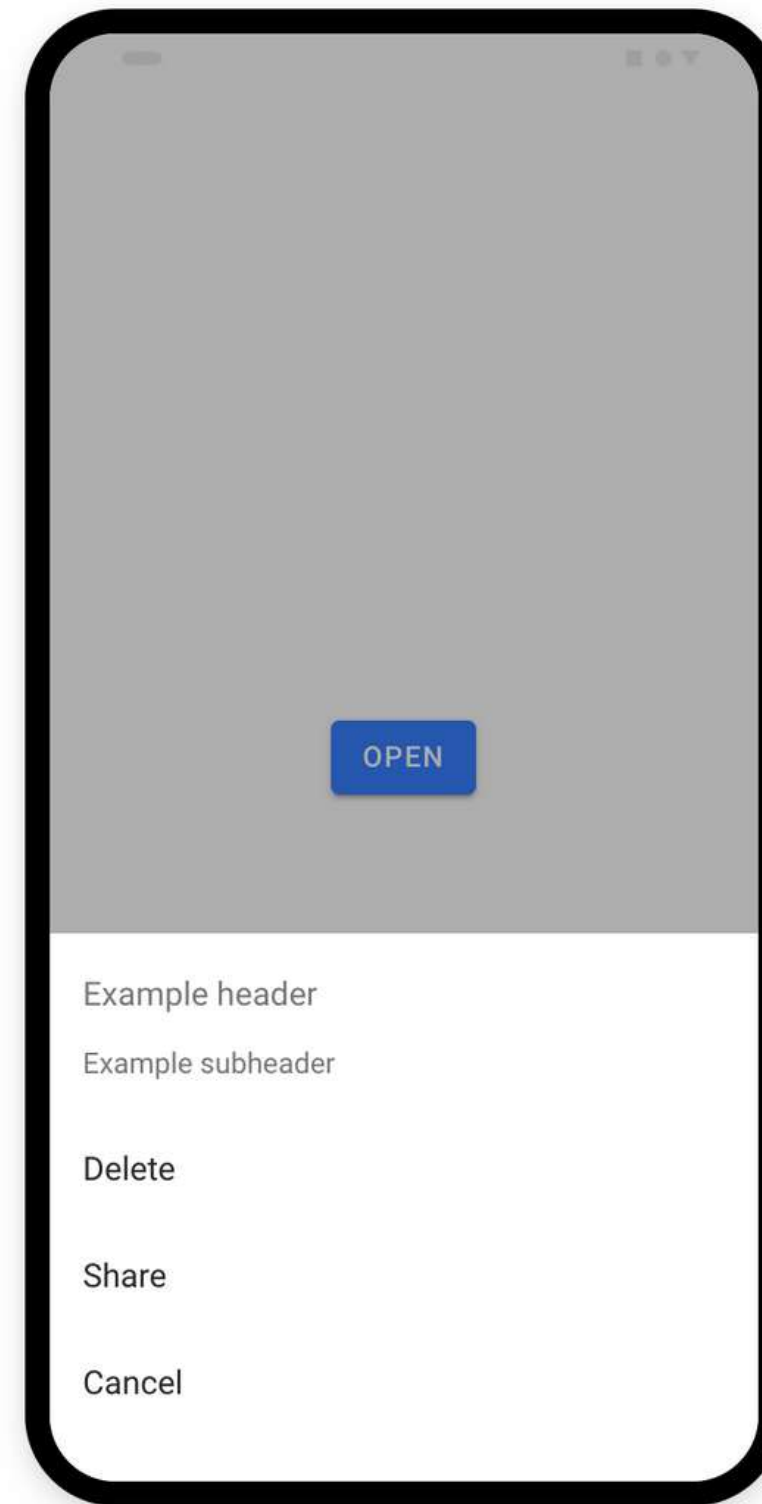
And gives us a lot of built-in components prepared for iOS and Android like:



Alerts:



Action sheets:



Toggles:

Default Toggle



Checked Toggle



Disabled Toggle



Disabled Checked Toggle



Default Toggle



Checked Toggle



Disabled Toggle



Disabled Checked Toggle





And many others:

- ▼ **List**
 - ion-list
 - ion-list-header
 - ion-virtual-scroll
- ▼ **Media**
 - ion-avatar
 - ion-icon
 - ion-img
 - ion-thumbnail
- ▼ **Menu**
 - ion-menu
 - ion-menu-button
 - ion-menu-toggle
 - ion-split-pane
- ▼ **Grid**
 - ion-grid
 - ion-col
 - ion-row
- ▼ **Infinite Scroll**
 - ion-infinite-scroll
 - ion-infinite-scroll-content
- ▼ **Icons**
 - ion-icon
- ▼ **Input**
 - ion-input
 - ion-textarea
- ▼ **Card**
 - ion-card
 - ion-card-content
 - ion-card-header
 - ion-card-subtitle
 - ion-card-title
- ▼ **Checkbox**
 - ion-checkbox
- ▼ **Chip**
 - ion-chip
- ▼ **Content**
 - ion-app
 - ion-content
- ▼ **Getting Started**
 - UI Components
- ▼ **Action Sheet**
 - ion-action-sheet
- ▼ **Accordion**
 - ion-accordion
 - ion-accordion-group
- ▼ **Alert**
 - ion-alert
- ▼ **Badge**
 - ion-badge
- ▼ **Breadcrumb**
 - ion-breadcrumb
 - ion-breadcrumbs
- ▼ **Button**
 - ion-button
 - ion-ripple-effect
- ▼ **Popover**
 - ion-popover
- ▼ **Progress Indicators**
 - ion-loading
 - ion-progress-bar
 - ion-skeleton-text
 - ion-spinner
- ▼ **Radio**
 - ion-radio
 - ion-radio-group
- ▼ **Range**
 - ion-range
- ▼ **Refresher**
 - ion-refresher
 - ion-refresher-content
- ▼ **Reorder**
 - ion-reorder
 - ion-reorder-group
- ▼ **Routing**
 - ion-router
 - ion-router-link
 - ion-router-outlet
 - ion-route
 - ion-route-redirect
- ▼ **Searchbar**
 - ion-searchbar
- ▼ **Segment**
 - ion-segment
 - ion-segment-button
- ▼ **Select**
 - ion-select
 - ion-select-option
- ▼ **Slides**
 - ion-slides
 - ion-slide
- ▼ **Tab**
 - ion-tabs
 - ion-tab
 - ion-tab-bar
 - ion-tab-button
- ▼ **Toast**
 - ion-toast
- ▼ **Toggle**
 - ion-toggle
- ▼ **Toolbar**
 - ion-toolbar
 - ion-header
 - ion-footer
 - ion-title
 - ion-buttons
 - ion-back-button
- ▼ **Typography**
 - ion-text
- ▼ **Resources**
 - API Index

We know the theory.

How we can create

our app then?

Let's look into 3 parts of app's creation:

Let's look into 3 parts of app's creation:

1. Setting it up.

Let's look into 3 parts of app's creation:

1. Setting it up.

2. Development.

Let's look into 3 parts of app's creation:

1. Setting it up.

2. Development.

3. Deployment.

**Let's see how the
setup can be done!**

Commands

I've used:



```
npm i -g @ionic/cli  
ionic start
```

```
ionic capacitor add ios  
ionic capacitor build ios
```

More on testing and debugging our code inside of simulators:

iOS Development

This guide covers how to run and debug Ionic apps on iOS simulators and devices using [Capacitor](#) or [Cordova](#). iOS apps can only be developed on macOS with Xcode installed.

There are two workflows for running Ionic apps on iOS:

- [Running with Xcode](#)
- [Running with the Ionic CLI](#)

More: [iOS Development](#) 

Android Development

This guide covers how to run and debug Ionic apps on Android emulators and devices using [Capacitor](#) or [Cordova](#). Android apps can be developed on Windows, macOS, and Linux.

Android Studio

[Android Studio](#) is the IDE for creating native Android apps. It includes the [Android SDK](#), which will need to be configured for use in the command line.

More: [Android Development](#) 



**The rest is classic web
app creation...**

**The rest is classic web
app creation...**

...with a bit of help from:

Ionic

components...

- ▼ **List**
 - ion-list
 - ion-list-header
 - ion-virtual-scroll
- ▼ **Media**
 - ion-avatar
 - ion-icon 
 - ion-img
 - ion-thumbnail
- ▼ **Menu**
 - ion-menu
 - ion-menu-button
 - ion-menu-toggle
 - ion-split-pane
- ▼ **Grid**
 - ion-grid
 - ion-col
 - ion-row
- ▼ **Infinite Scroll**
 - ion-infinite-scroll
 - ion-infinite-scroll-content
- ▼ **Icons**
 - ion-icon 
- ▼ **Input**
 - ion-input
 - ion-textarea
- ▼ **Card**
 - ion-card
 - ion-card-content
 - ion-card-header
 - ion-card-subtitle
 - ion-card-title
- ▼ **Checkbox**
 - ion-checkbox
- ▼ **Chip**
 - ion-chip
- ▼ **Content**
 - ion-app
 - ion-content
- ▼ **Getting Started**
 - UI Components
- ▼ **Action Sheet**
 - ion-action-sheet
- ▼ **Accordion**
 - ion-accordion
 - ion-accordion-group
- ▼ **Alert**
 - ion-alert
- ▼ **Badge**
 - ion-badge
- ▼ **Breadcrumb**
 - ion-breadcrumb
 - ion-breadcrumbs
- ▼ **Button**
 - ion-button
 - ion-ripple-effect
- ▼ **Popover**
 - ion-popover
- ▼ **Progress Indicators**
 - ion-loading
 - ion-progress-bar
 - ion-skeleton-text
 - ion-spinner
- ▼ **Radio**
 - ion-radio
 - ion-radio-group
- ▼ **Range**
 - ion-range
- ▼ **Refresher**
 - ion-refresher
 - ion-refresher-content
- ▼ **Reorder**
 - ion-reorder
 - ion-reorder-group
- ▼ **Routing**
 - ion-router
 - ion-router-link
 - ion-router-outlet
 - ion-route
 - ion-route-redirect
- ▼ **Searchbar**
 - ion-searchbar
- ▼ **Segment**
 - ion-segment
 - ion-segment-button
- ▼ **Select**
 - ion-select
 - ion-select-option
- ▼ **Slides**
 - ion-slides
 - ion-slide
- ▼ **Tabbar**
 - ion-tabs
 - ion-tab
 - ion-tab-bar
 - ion-tab-button
- ▼ **Toast**
 - ion-toast
- ▼ **Toggle**
 - ion-toggle
- ▼ **Toolbar**
 - ion-toolbar
 - ion-header
 - ion-footer
 - ion-title
 - ion-buttons
 - ion-back-button
- ▼ **Typography**
 - ion-text
- ▼ **Resources**
 - API Index

...3 groups of Capacitor plugins that handle native functionality:

1. Official plugins (26):

- Action Sheet
- App
- App Launcher
- Browser
- Camera
- Clipboard
- Cookies
- Device
- Dialog
- Filesystem
- Geolocation
- Google Maps
- Haptics
- Http
- Keyboard
- Local Notifications
- Motion
- Network
- Preferences
- Push Notifications
- Screen Reader
- Share
- Splash Screen
- Status Bar
- Text Zoom
- Toast

2. Community plugins (55):

capacitor-community

Packages 55

@capacitor-community/admob

A native plugin for AdMob

rdlabo published 4.0.0 • 4 months ago

@capacitor-community/advertising-id

Allows access to the IDFA (iOS) and AAID (Android)

thomasvidas published 1.0.0 • 8 months ago

@capacitor-community/appcenter

Capacitor Plugin for Microsoft's App Center SDK.

johnborges published 3.0.0 • 4 months ago

@capacitor-community/appcenter-analytics

Capacitor plugin for AppCenter Analytics

@capacitor-community/appcenter-crashes

Capacitor plugin for Microsoft's App Center Crashes

johnborges published 3.0.1 • 3 months ago

@capacitor-community/app-icon

Capacitor community plugin for changing an iOS app icon.

johnborges published 3.1.0-beta.0 • 3 months ago

@capacitor-community/apple-sign-in

Capacitor Sign in with Apple

jcesarmobile published 4.0.0 • 3 months ago

@capacitor-community/app-react

React Hooks for Capacitor App Plugin

capcombot published 0.1.0 • a year ago

@capacitor-community/background-geolocation

Receive geolocation updates even while app is backgrounded.

diachedelic published 1.2.6 • 2 months ago

@capacitor-community/barcode-scanner

A fast and efficient (QR) barcode scanner for Capacitor

thegnuu published 3.0.1 • a month ago

3. Awesome Cordova Plugins - previously Ionic Native (229!):

awesome-cordova-plugins

Packages 229

@awesome-cordova-plugins/abbyy-rtr

Awesome Cordova Plugins - Native plugins for ionic apps

danielsogl published 6.2.0 • 2 months ago

@awesome-cordova-plugins/action-sheet

Awesome Cordova Plugins - Native plugins for ionic apps

danielsogl published 6.2.0 • 2 months ago

@awesome-cordova-plugins/adjust

Awesome Cordova Plugins - Native plugins for ionic apps

@awesome-cordova-plugins/admob-plus

Awesome Cordova Plugins - Native plugins for ionic apps

danielsogl published 6.2.0 • 2 months ago

@awesome-cordova-plugins/admob-pro

Awesome Cordova Plugins - Native plugins for ionic apps

danielsogl published 6.2.0 • 2 months ago

@awesome-cordova-plugins/aes-256

Awesome Cordova Plugins - Native plugins for ionic apps

danielsogl published 6.2.0 • 2 months ago

@awesome-cordova-plugins/all-in-one-sdk

Awesome Cordova Plugins - Native plugins for ionic apps

History: A New Chapter for @ionic-native



..and some native code if needed:

Custom Native iOS Code

With Capacitor, you are encouraged to write Swift or Objective-C code to implement the native features your app needs.

There may not be [a Capacitor plugin](#) for everything--and that's okay! It is possible to write WebView-accessible native code right in your app.

WebView-Accessible Native Code

The easiest way to communicate between JavaScript and native code is to build a custom Capacitor plugin that is local to your app.

[More: Custom Native iOS Code](#)



Custom Native Android Code

With Capacitor, you are encouraged to write Java or Kotlin code to implement the native features your app needs.

There may not be [a Capacitor plugin](#) for everything--and that's okay! It is possible to write WebView-accessible native code right in your app.

WebView-Accessible Native Code

The easiest way to communicate between JavaScript and native code is to build a custom Capacitor plugin that is local to your app.

[More: Custom Native Android Code](#)



**Then we do the
deployment!**

We build our app for iOS and Android:



```
ionic capacitor build ios --prod  
ionic capacitor build android --prod
```

...then upload them to App Store:

- 1. Set up Apple Developers account**
- 2. Generate signing certificates**
- 3. Build app archive ready to be sent to App Store**
- 4. Send app to App Store using Xcode Organizer**
- 5. Send app for approval to Apple**

More: [iOS App Store Deployment](#)



...and Play Store:

- 1. Set up Google Play Console account**
- 2. Generate upload key and keystore in Android Studio**
- 3. Sign the app**
- 4. Create a release in Play Store and upload signed app**

More:

- 1. [Android Play Store Deployment](#)**
- 2. [Publish your app \(Android Developers\)](#)**



**And then we have
our app!**


(hopefully on deadline)

**Couple of FYIs
at the end**



**You can easily add
Capacitor to the
existing project**

More: [Adding Capacitor to your existing web application](#) ✨



**You can use micro
frontends to skip app's
review (while updating
the app)**

**More: [How to bypass mobile app's
review thanks to micro frontends](#)**





**Your hybrid app can
easily become PWA**

For example, in case of Ionic and React, you only need to...

...change one line of code:



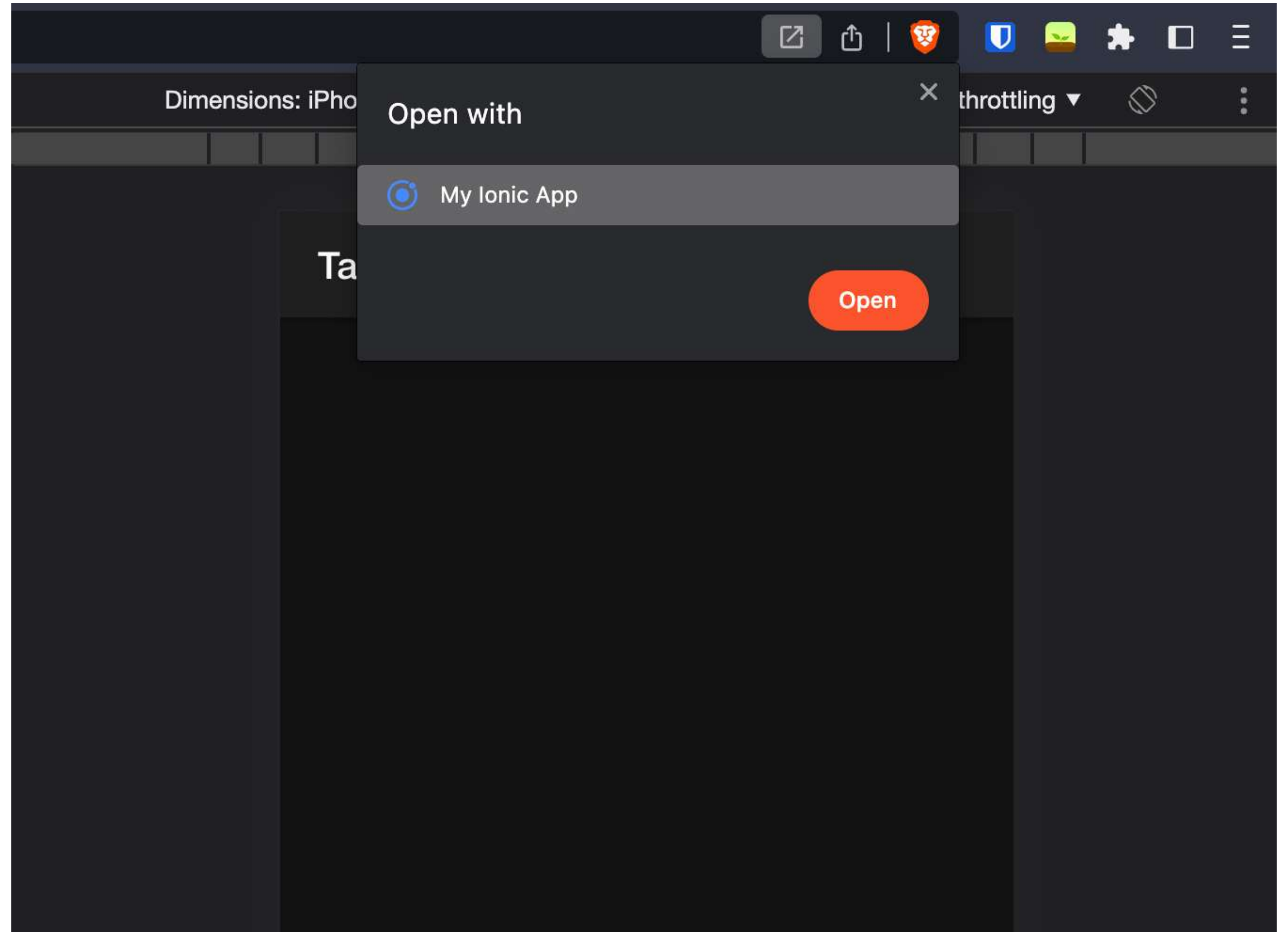
```
// src/index.tsx
```

```
serviceWorkerRegistration.unregister();
```

```
// to:
```

```
serviceWorkerRegistration.register();
```

And we have it!



More: [Progressive Web Apps in React](#)



**Extras once
again:**


robertorlinski.pl/mobile-apps



**Now a small
contest!**

**2 questions,
2 copies of this
book to get:**





**What are hybrid
mobile apps?**



**How does the
Capacitor work?**

Congratulations!

**And the last
part - Q&A**

Thank you!

**I will appreciate any
feedback:**

