

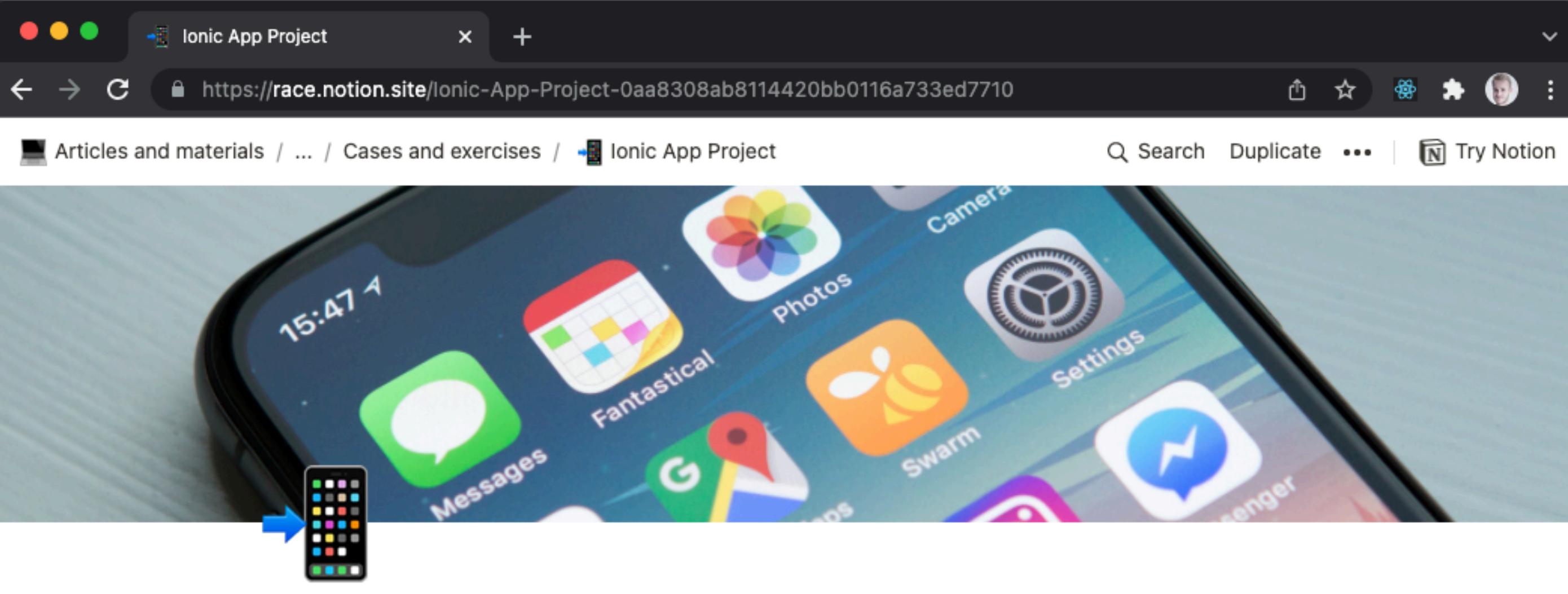
Firebase & App Dev

APP Development



Agenda

- Ionic App Project
- Ionic Later List App w/ Firebase REST
- Firebase, Databases & Storage
- Firebase Authentication
- Wrap up and next step



Ionic App Project

Develop a Mobile App based on themes, concept and terms from the first part of the Mobile App Dev Course.

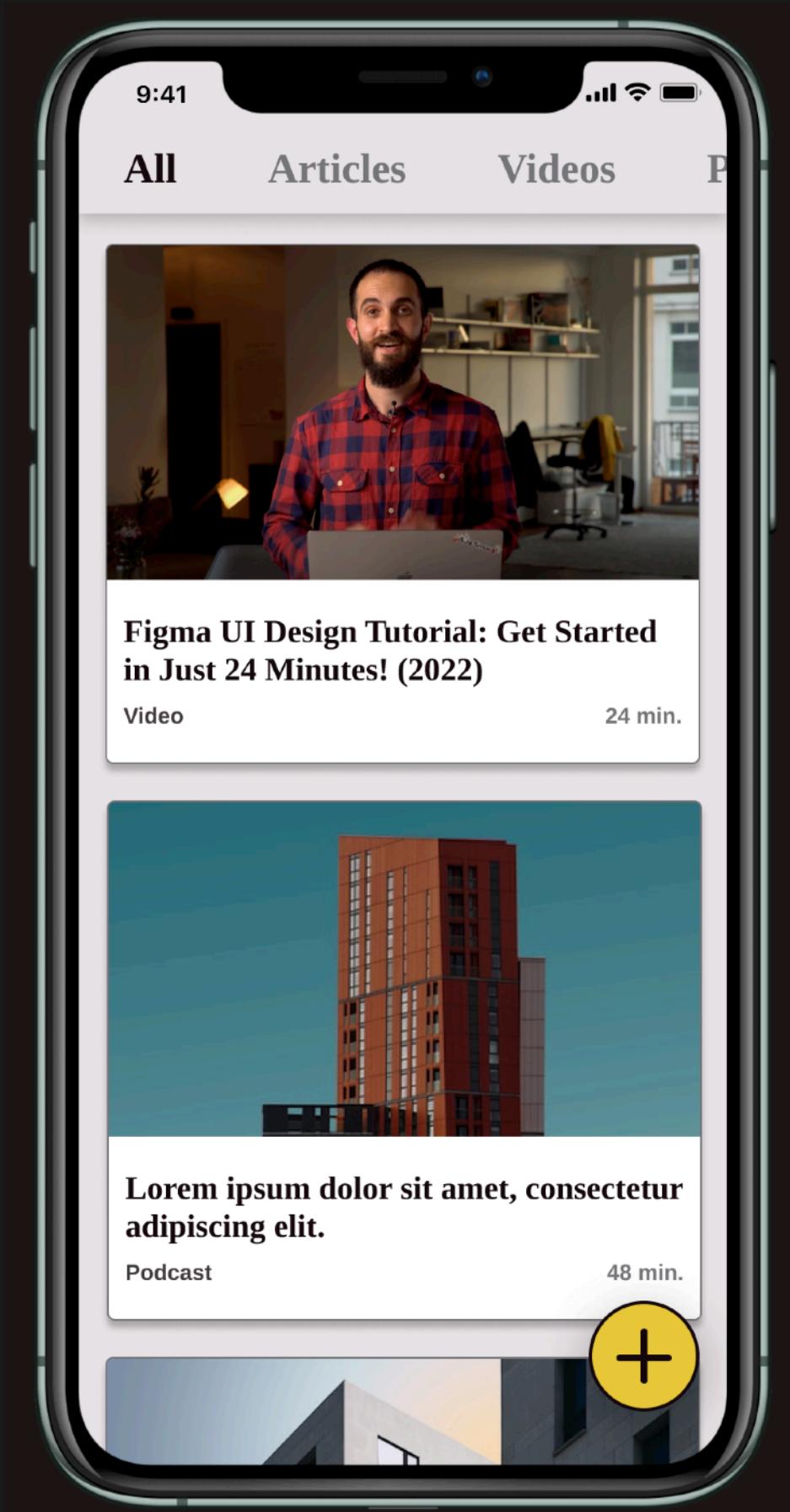
The Mobile App is developed in groups of 2-3 students. The project is based on a self-chosen theme and must meet the below technical requirements, demonstrating your learned skills and competencies from the course.

There are no requirements in addition to research and project management but feel free to use known project tools. However the main purpose is to practice programming and development of mobile apps with React, Ionic React and Capacitor JS.

The project is finalised by handing in a link to a GitHub repository including a README.md file with a short description of your app.

[Ionic App Project Description](#)

Ionic Later List App



<https://race.notion.site/Ionic-Later-List-App-866a28f6bf284b109183863d41bdd0ee>

Individual

ionic-later-list-app

1. Run the App and check out the project and code.
2. Focus on /components/ and /pages/

In pairs

ionic-later-list-app

1. What are the differences from the project(s) we worked on last week?
2. How can you use this project in addition to the Ionic App Project?
2. In order to keep data synced and updated, what is the challenge when using a REST API?

Round The Table

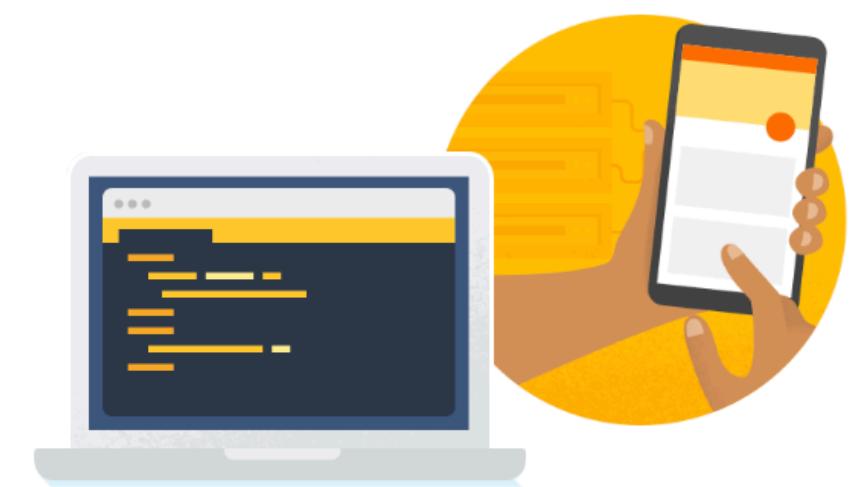
ionic-later-list-app

1. Discuss your findings.
2. Note questions & comments.



What is Firebase?

Platform & Backend-as-a-Service
for Web & App Development



Build better apps



Cloud Firestore

Store and sync app data at global scale



ML Kit BETA

Machine learning for mobile developers



Cloud Functions

Run mobile backend code without managing servers



Authentication

Authenticate users simply and securely



Hosting

Deliver web app assets with speed and security



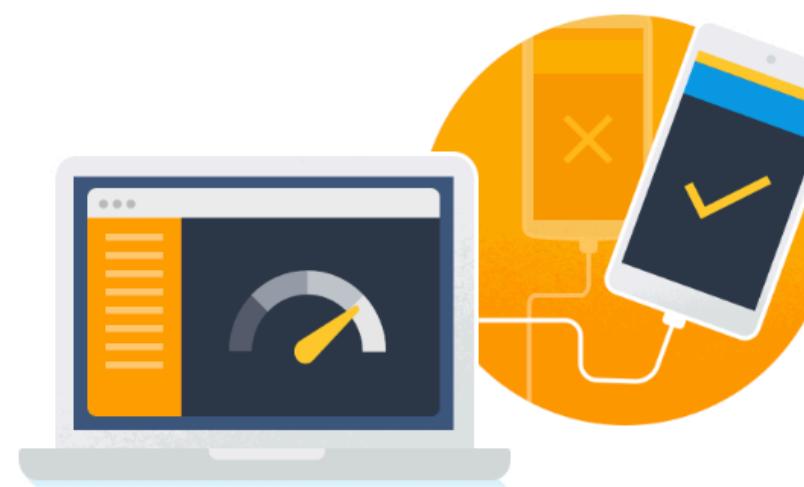
Cloud Storage

Store and serve files at Google scale



Realtime Database

Store and sync app data in milliseconds



Improve app quality



Crashlytics

Prioritize and fix issues with powerful, realtime crash reporting



Performance Monitoring

Gain insight into your app's performance



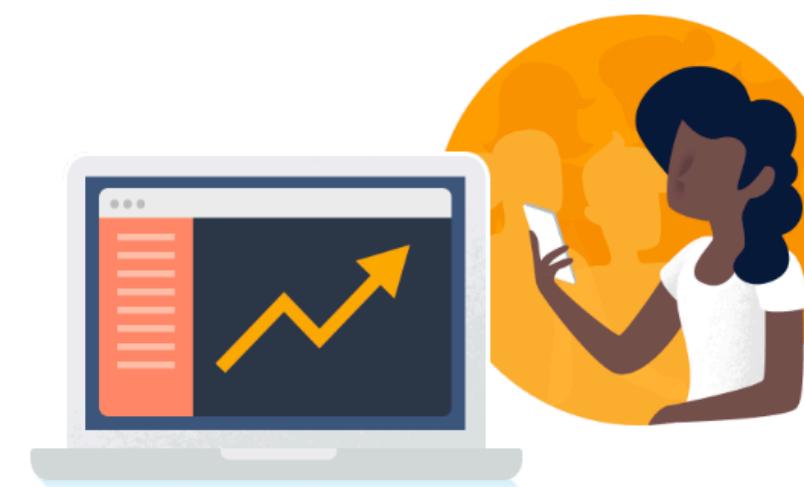
Test Lab

Test your app on devices hosted by Google



App Distribution BETA

Distribute pre-release versions of your app to your trusted testers



Grow your business



In-App Messaging BETA

Engage active app users with contextual messages



Google Analytics

Get free and unlimited app analytics



Predictions

Smart user segmentation based on predicted behavior



A/B Testing BETA

Optimize your app experience through experimentation



Cloud Messaging

Send targeted messages and notifications



Remote Config

Modify your app without deploying a new version



Dynamic Links

Drive growth by using deep links with attribution



Realtime Database & REST API

Store and sync data in real time
REST API or SDK

<https://firebase.google.com/products/realtime-database>

Realtime Database REST API

[https://firebase.google.com/docs/
database/rest/start](https://firebase.google.com/docs/database/rest/start)

```
export default function PostsPage() {
  const [posts, setPosts] = useState([]);
  const [showLoader, dismissLoader] = useIonLoading();

  async function getPosts() {
    const response = await fetch("https://race-rest-default-rtbd.firebaseio.com/posts.json");
    const data = await response.json();
    // map object into an array with objects
    const postsArray = Object.keys(data).map(key => ({ id: key, ...data[key] }));
    return postsArray;
  }

  async function getUsers() {
    const response = await fetch("https://race-rest-default-rtbd.firebaseio.com/users.json");
    const data = await response.json();
    // map object into an array with objects
    const users = Object.keys(data).map(key => ({ id: key, ...data[key] }));
    return users;
  }
}
```

Using the SDK

The screenshot shows a web browser window displaying the Firebase Realtime Database JS SDK documentation at <https://firebase.google.com/docs/database/web/start>. The page title is "Installation & Setup in JavaScript". The left sidebar contains navigation links for Overview, Emulator Suite, Authentication, Realtime Database (with sub-links for Introduction, Choose a Database, iOS+, Android, and Web), Get Started (with sub-links for Structure Data, Read and Write Data, Work with Lists of Data, Enable Offline Capabilities, Flutter, Admin, REST, C++, Unity, Security & Rules, Usage and Performance, Automated Backups, Extend with Cloud Functions, and Video series: Firebase for SQL Developers), and Cloud Firestore. The main content area is titled "Add the Realtime Database JS SDK and initialize Realtime Database". It instructs users to specify their Realtime Database URL when initializing the JavaScript SDK. It provides the URL format: `https://DATABASE_NAME.firebaseio.com` (for databases in `us-central1`) or `https://DATABASE_NAME.REGION.firebaseio.app` (for databases in all other locations). It also includes a code snippet for initializing the SDK:

```
Web version 9 (modular)    Web version 8 (namespaced)

★ Learn more about the tree-shakeable Web v9 modular SDK and upgrade from version 8.

import { initializeApp } from "firebase/app";
import { getDatabase } from "firebase/database";

// TODO: Replace the following with your app's Firebase project configuration
// See: https://firebase.google.com/docs/web/learn-more#config-object
const firebaseConfig = {
  // ...
  // The value of `databaseURL` depends on the location of the database
  databaseURL: "https://DATABASE_NAME.firebaseio.com",
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);

// Initialize Realtime Database and get a reference to the service
const database = getDatabase(app);
```

Using the SDK

The screenshot shows a web browser displaying the Firebase documentation at <https://firebase.google.com/docs/database/web/read-and-write>. The page title is "Read and Write Data on the Web". The left sidebar navigation includes "Emulator Suite", "Authentication", "Realtime Database" (with "Introduction", "Choose a Database", "iOS+", "Android", "Web" sections), "Work with Lists of Data", "Enable Offline Capabilities", "Flutter", "Admin", "REST", "C++", "Unity", "Security & Rules", "Usage and Performance", "Automated Backups", "Extend with Cloud Functions", and "Video series: Firebase for SQL Developers". Below these are sections for "Cloud Firestore" and "Cloud Storage". The main content area is titled "Read data" and "Listen for value events". It explains how to use `onValue()` to observe events for static snapshots or changes. A note cautions against attaching listeners to the root of the database. An example code snippet for the Web version 9 (modular) shows how to import the database and listen for star count changes in a social blogging application. The code uses `getDatabase()`, `ref()`, and `onValue()` methods to update a post's star count.

```
import { getDatabase, ref, onValue} from "firebase/database";

const db = getDatabase();
const starCountRef = ref(db, 'posts/' + postId + '/starCount');
onValue(starCountRef, (snapshot) => {
  const data = snapshot.val();
  updateStarCount(postElement, data);
});
```

rtbd_social_listen_star_count.js

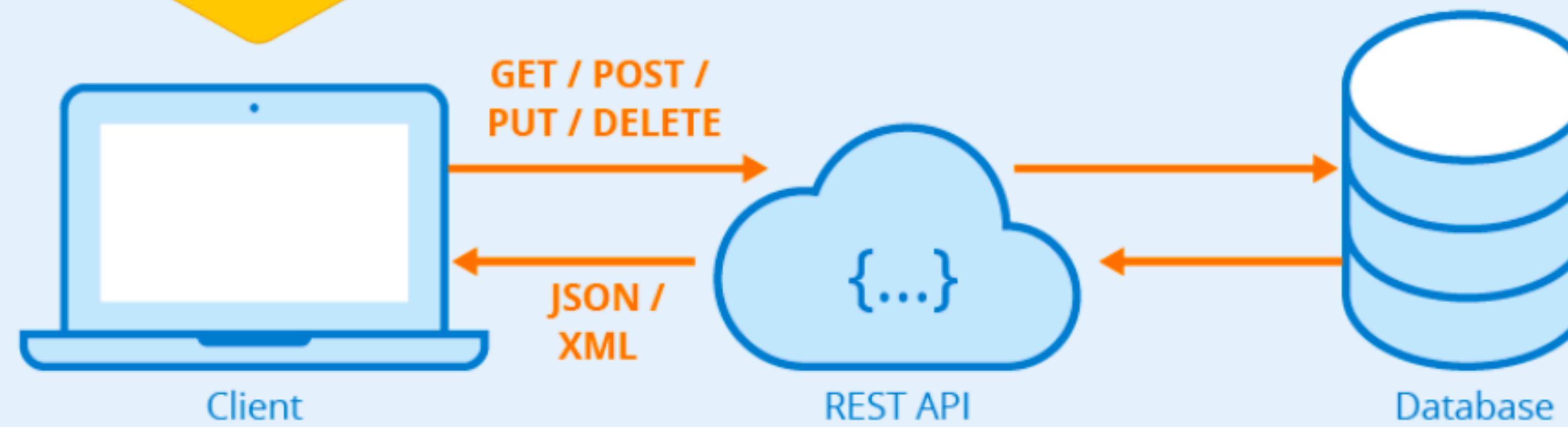
The listener receives a `snapshot` that contains the data at the specified location in the database at the time of the event. You can retrieve the data in the `snapshot` with the `val()` method.



Database



Firebase

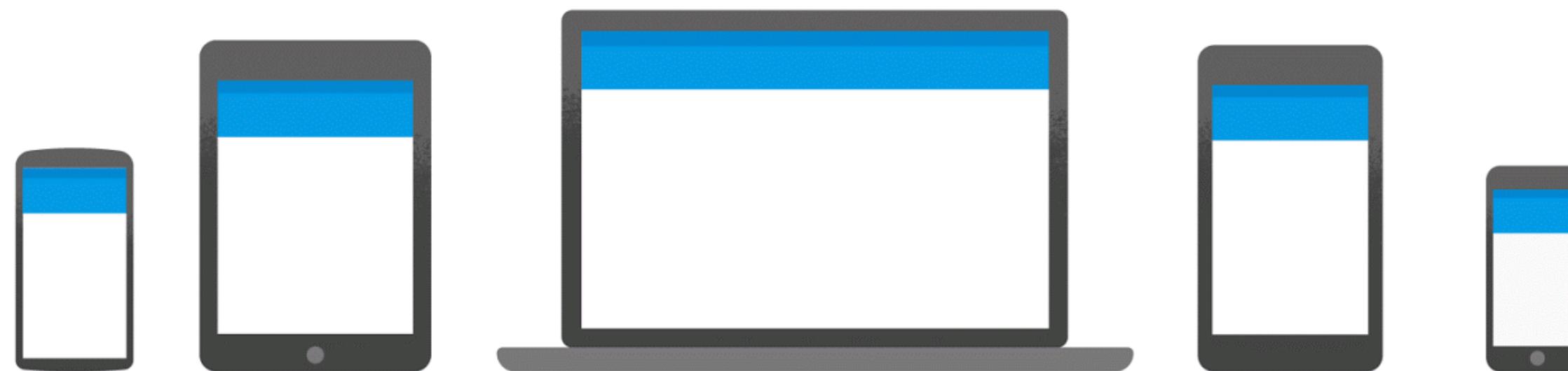


Ionic Post App w/ Realtime Database

Step Guide

Implement the Realtime Database SDK

REALTIME UPDATES





Cloud Firestore

<https://www.youtube.com/watch?v=QcsAb2RR52c>



CLOUD FIRESTORE

STORE AND SYNC DATA ACROSS PLATFORMS

<https://firebase.google.com/products/firestore/>



CLOUD FIRESTORE

CREATE, READ, UPDATE & DELETE DATA

<https://firebase.google.com/products/firestore/>



FIREBASE TEMPLATES

`firebase-*`

`spa-firebase-*`

`react-cdn-firebase-*`

`React-cdn-firebase-*`



FIREBASE DOCS

Getting started video: https://youtu.be/iosNuldQoy8?list=PLI-K7zZEsYLmOF_07IayrTntevxtbUxDL

Cloud Firestore Introduction: <https://firebase.google.com/docs/firestore>

Get started with Cloud Firestore: <https://firebase.google.com/docs/firestore/quickstart>

Add and manage data: <https://firebase.google.com/docs/firestore/manage-data/add-data>

Delete data: <https://firebase.google.com/docs/firestore/manage-data/delete-data>

Read data: <https://firebase.google.com/docs/firestore/query-data/get-data>

Read data - realtime updates: <https://firebase.google.com/docs/firestore/query-data/listen>

Order and limit data: <https://firebase.google.com/docs/firestore/query-data/order-limit-data>



Documentation

<https://firebase.google.com/docs/firestore/>[Overview](#) [Guides](#) [Reference](#) [Samples](#) [Libraries](#)[Send feedback](#)

Guides

[Get started with Firebase](#)[Manage your Firebase projects](#)[Prototype and test with Emulator Suite](#)

Analytics

[DEVELOP](#)

Authentication

[Realtime Database](#)

Cloud Firestore

Introduction

- [Get started](#)
- [Understand Cloud Firestore](#)
- [Add and manage data](#)
- [Read data](#)
- [Secure and validate data](#)
- [Solutions](#)
- [Usage, limits, and pricing](#)
- [Cloud Firestore integrations](#)
- [API reference](#)
- [Samples](#)

Storage

Firebase > Docs > Guides



Cloud Firestore

 [iOS](#) [Android](#) [Node.js](#) [Java](#) [Python](#) [Go](#)

Use our flexible, scalable NoSQL cloud database to store and sync data for client- and server-side development.

Cloud Firestore is a flexible, scalable database for mobile, web, and server development from Firebase and Google Cloud Platform. Like Firebase Realtime Database, it keeps your data in sync across client apps through realtime listeners and offers offline support for mobile and web so you can build responsive apps that work regardless of network latency or Internet connectivity. Cloud Firestore also offers seamless integration with other Firebase and Google Cloud Platform products, including Cloud Functions.

[Get started](#)

Key capabilities

Flexibility

The Cloud Firestore data model supports flexible, hierarchical data structures. Store your data in documents, organized into collections. Documents can contain complex nested objects in addition to subcollections.

Contents

[Key capabilities](#)[How does it work?](#)[Implementation path](#)[Next steps](#)

CREATE

```
addDoc(_usersRef, newUser);
```

READ

```
onSnapshot(_usersRef, snapshot => { ...  
});
```

UPDATE

```
const userRef = doc(_usersRef, _selectedUserId);  
updateDoc(userRef, userToUpdate);
```

DELETE

```
const userRef = doc(_usersRef, id);  
deleteDoc(userRef);
```



Getting Started

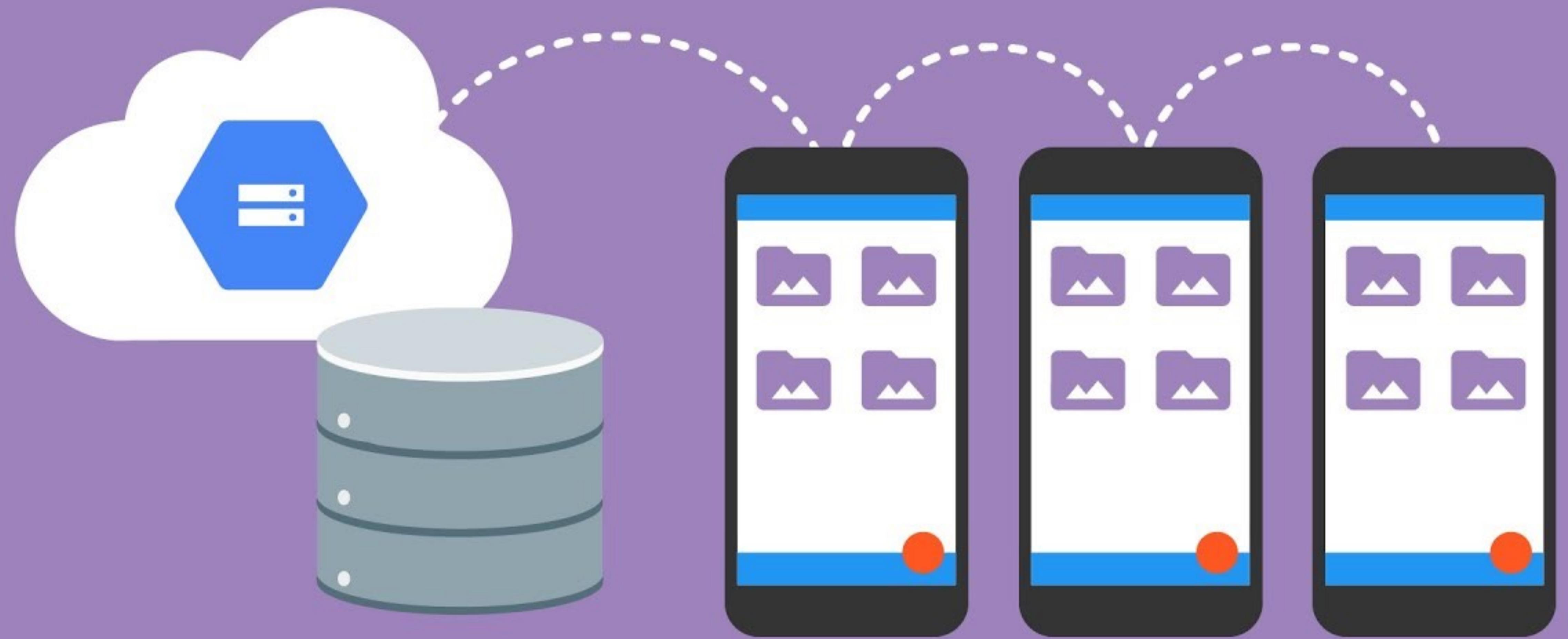
Getting Started with Firebase Cloud Firestore & React



Cloud Storage

Store and serve content, images, files, etc.

<https://firebase.google.com/products/storage>



Storage

Ionic Post App w/ Cloud Storage

Step Guide

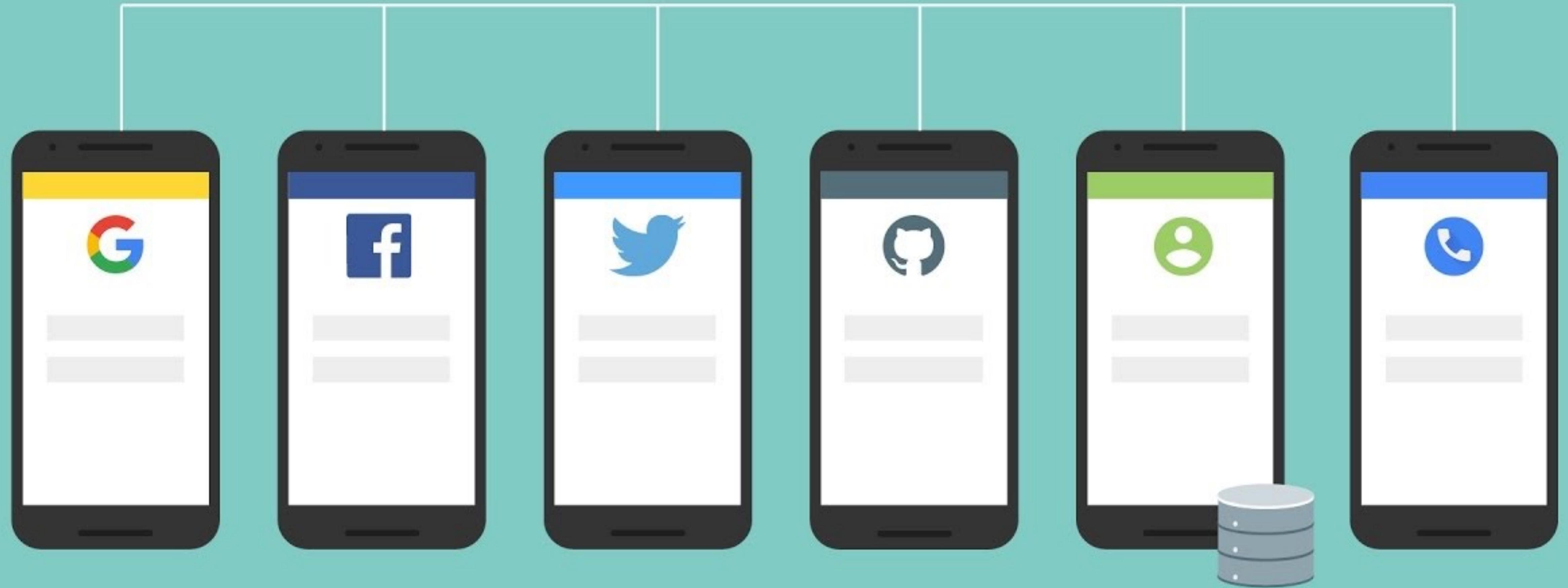
Implement the Cloud Storage SDK



Firebase Authentication

Sign Up, Sign In and authenticate users

<https://firebase.google.com/products/auth>



Authentication



Project Overview



Develop

Authentication

Database

Storage

Hosting

Functions

ML Kit

Quality

Crashlytics, Performance, Test Lab

Analytics

Dashboard, Events, Conversions, Au...

Grow

Predictions, A/B Testing, Cloud Mes...

Spark
Free \$0/month

Upgrade

Authentication

Users Sign-in method Templates Usage

Sign-in providers

Provider	Status
Email/Password	<input checked="" type="checkbox"/> Enable
Email link (passwordless sign-in)	<input type="checkbox"/> Enable

Allow users to sign up using their email address and password. Our SDKs also provide email address verification, password recovery and email address change primitives. [Learn more](#)

[Cancel](#) [Save](#)

Phone	Disabled
Google	Disabled
Play Games	Disabled
Game Center <small>Beta</small>	Disabled
Facebook	Disabled
Twitter	Disabled
Github	Disabled



Project Overview



Develop

Authentication

Database

Storage

Hosting

Functions

ML Kit

Quality

Crashlytics, Performance, Test Lab

Analytics

Dashboard, Events, Conversions, Au...

Grow

Predictions, A/B Testing, Cloud Mes...

Spark
Free \$0/month

Upgrade

Authentication

Users Sign-in method Templates Usage



Search by email address, phone number or user UID

Identifier	Providers	Created	Signed In	User UID ↑
rasmus@cederdorff.com		9 Sep 2019	9 Sep 2019	mdAhTk3kLwhl3enYjMspg0e6g492

Rows per page: 50 ▾ 1-1 of 1 < >

when you ask Rasmus
for help and he says
"Read documentation"



```
function App() {
  const auth = getAuth();
  const [isAuth, setIsAuth] = React.useState(localStorage.getItem("isAuth"));

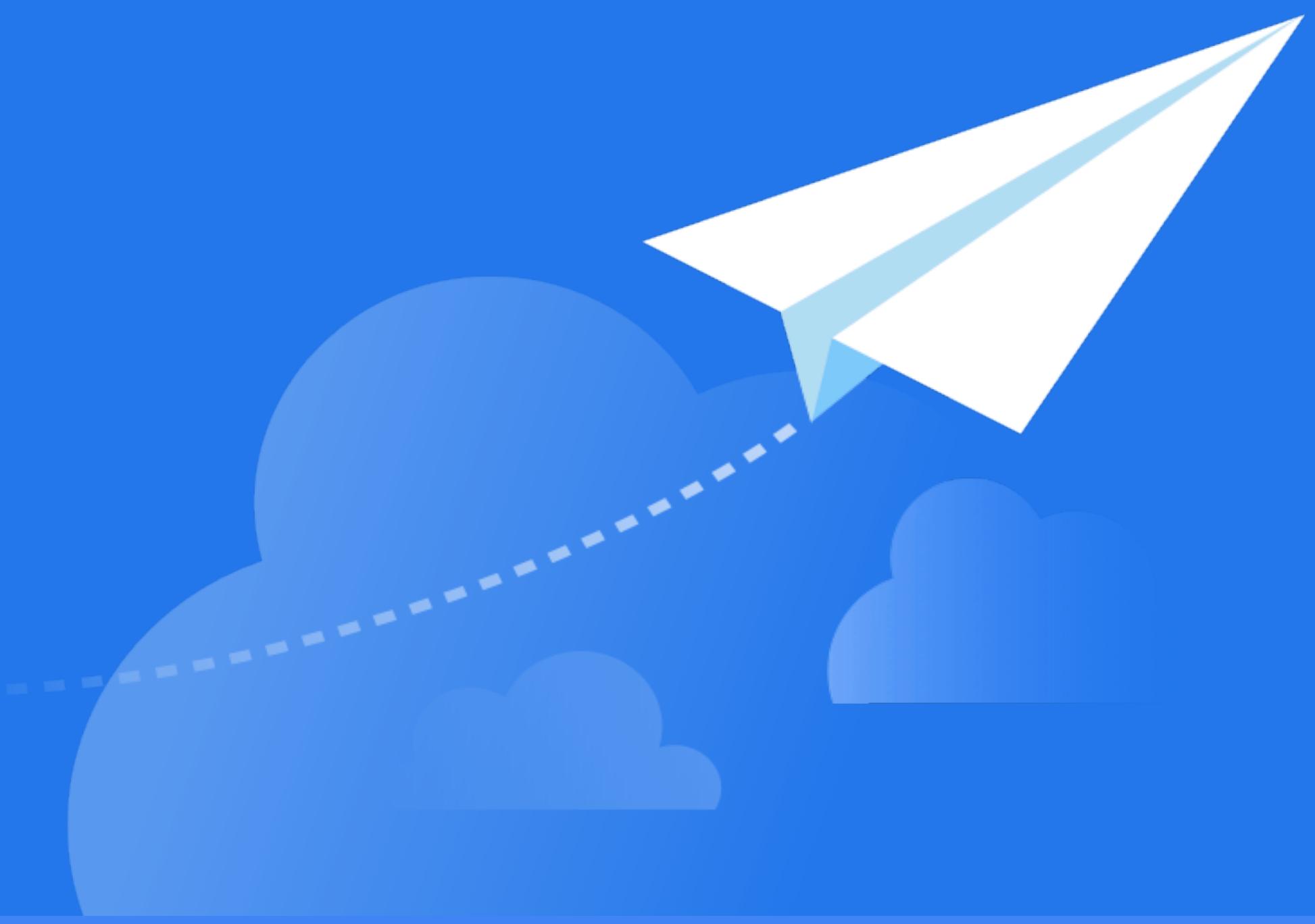
  onAuthStateChanged(auth, user => {
    if (user) {
      // User is signed in
      setIsAuth(true);
      localStorage.setItem("isAuth", true);
    } else {
      // User is signed out
      setIsAuth(false);
      localStorage.removeItem("isAuth");
    }
  });
}
```

```
return (
  <main>
    {isAuth ? (
      <>
        <Nav />
        <Routes>
          <Route path="/" element={<PostsPage />} />
          <Route path="/create" element={<CreatePage />} />
          <Route path="/profile" element={<ProfilePage />} />
          <Route path="/posts/:postId" element={<UpdatePage />} />
          <Route path="*" element={<Navigate to="/" />} />
        </Routes>
      </>
    ) : (
      <Routes>
        <Route path="/signin" element={<SignInPage />} />
        <Route path="/signup" element={<SignUpPage />} />
        <Route path="*" element={<Navigate replace to="/signin" />} />
      </Routes>
    )}
  </main>
);
}
```

Ionic Post App w/ Authentication

Step Guide

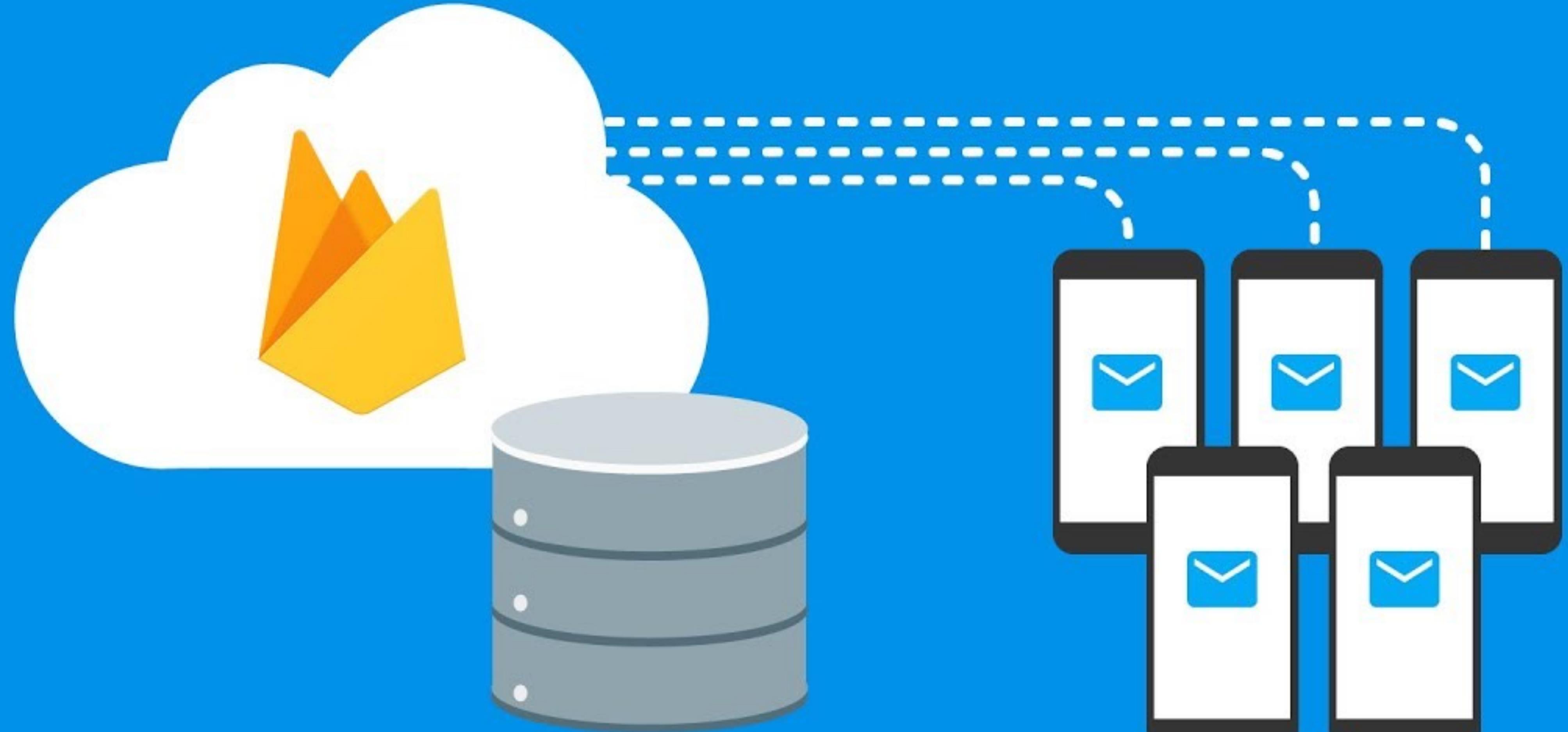
ionic-post-app-realtime-db-storage-auth



Cloud Messaging

Send notifications across platforms

<https://firebase.google.com/products/cloud-messaging>



Cloud Messaging

A screenshot of a web browser window displaying the Capacitor Push Notifications API documentation. The URL in the address bar is <https://capacitorjs.com/docs/apis/push-notifications>. The page is part of the Ionic Capacitor Docs, version 3.

The left sidebar shows a navigation tree under the 'APIs' category:

- > Overview
- ▽ APIs
 - Action Sheet
 - App
 - App Launcher
 - Browser
 - Camera
 - Clipboard
 - Device
 - Dialog
 - Filesystem
 - Geolocation
 - Haptics
 - Keyboard
 - Local Notifications

The main content area features a large heading **@capacitor/push-notifications**. Below it, a sub-section titled **Install** contains the following command-line instructions:

```
npm install @capacitor/push-notifications
npx cap sync
```

To the right of the main content, there is a sidebar titled **CONTENTS** which lists various topics related to push notifications:

- Install
- iOS
- Android
- Variables
- Push Notifications icon
- Push notifications appearance in foreground
- Examples
- Silent Push Notifications / Data-only Notifications
 - iOS
 - Android
- Common Issues
- Example
- API
 - register()
 - getDeliveredNotifications(..)

<https://capacitorjs.com/docs/apis/push-notifications>

The screenshot shows a web browser window with the title "Push Notifications with Firebase" at the top. The URL in the address bar is <https://capacitorjs.com/docs/guides/push-notifications-firebase>. The page is part of the Capacitor.js documentation, specifically the "Push Notifications with Firebase" guide.

The left sidebar contains a navigation menu:

- > Getting Started
- > Basics
- > Upgrade Guides
- > Cordova/PhoneGap
- ▼ Concepts
 - Ads
 - Angular
 - Automated Configuration
 - CI/CD
 - Deep Links
 - Deploying and Updating
 - Environment Specific Configurations
 - In App Purchases
 - Live Reload
 - Mocking Plugins
 - Push Notifications - Firebase**
 - React Hooks
 - Screen Orientation
 - Security

The main content area features a large heading: **Using Push Notifications with Firebase in an Ionic + Angular App**.

Below the heading, there's a section titled "Web Framework: Angular Platforms: iOS, Android".

The text explains: "One of the most common features provided by application developers to their users is push notifications. In this tutorial, we'll walk through all the steps needed to get [Firebase Cloud Messaging](#) working on iOS and Android."

Further down, it says: "For the purposes of registering and monitoring for push notifications from Firebase, we'll make use of the [Push Notification API for Capacitor](#) in an Ionic + Angular application."

A section titled "Required Dependencies" is visible at the bottom of the main content area.

The right sidebar is titled "CONTENTS" and lists several sections:

- Required Dependencies
- Prepare an Ionic Capacitor App
- Building the App & Adding Platforms
- Using the Capacitor Push Notification API
- Creating a Project for your App on Firebase
- Android
 - Integrating Firebase with the Android app
 - Download and Use the google-services.json file
- iOS
 - Prerequisites
 - Integrating Firebase with our native iOS app
 - Add the GoogleService-Info.plist file to your iOS app
 - Add the Firebase SDK via CocoaPods
 - Update the Project
 - Add Initialization Code

<https://capacitorjs.com/docs/guides/push-notifications-firebase>

ionic-push-notifications

Course Overview - Part 1

Date	Name	Notes
23/08/2022	1. Introduction to Mobile App Dev	Course Intro
30/08/2022	2. Web Native, Tools & UI Components	
06/09/2022	3. Native Device Features	Ionic Project Kick Off
13/09/2022	4. Ionic Project 1	Group Work
20/09/2022	5. Storage & Native APIs	Internship info at 14.00
27/09/2022	6. Firebase & App Dev	Final Ionic React lecture
04/10/2022	7. Ionic Project 2	Group Work
11/10/2022	8. Ionic Project 3	Group Work
18/08/2022	9. Ionic Project 4	Group Work

Exam

The exam is an individual written exam in the form of a report with a maximum of 15 pages including images, which is prepared on the basis of a group project.

Two Mobile App Projects

During the Mobile App Dev Course, you will be introduced to two approaches of Mobile App Development (Ionic React and Unity), finalised by a project.

The projects forms the basis of your exam in Mobile App Dev and must be handed in, in order to pass the course and participate in the exam at the end of the semester.

Comparing the different approaches, will be a part of the final individual written exam. Read more about the exam in the curriculum.

Two Mobile App Projects

Scope

- The exam is an individual written exam in the form of a report with a maximum of 15 pages including images, which is prepared based on either your Ionic App or Unity App group project.
- During the exam, you must review your previous project, and improve and extend your mobile app (Ionic or Unity).
- The technical requirements are equal to the listed in the project descriptions (Ionic App Project or Unity App Project)
- In the exam project, you must compare the two different approaches to mobile app development (Ionic VS Unity) and reflect on the pros and cons.

GitHub

The screenshot shows a GitHub repository page for the user 'cederdorff' with the repository name 'web-mobile-app-dev'. The page is in dark mode. The 'Code' tab is selected, displaying a list of 85 commits. The commits are organized into several branches, with the main branch having 1 commit. The commits are dated 6 months ago and involve changes to files like '_slides', 'data', 'ionic-camera-app', 'ionic-geolocation', 'ionic-post-app-firebase-rest-v0', 'ionic-post-app-firebase-rest-v1', 'ionic-post-app-firebase-rest-v2', 'ionic-post-app-firebase-rest-v3', 'ionic-post-app-firebase-rest-v4', 'ionic-post-app-firebase-rest-v5', 'ionic-post-app-firebase-rest-v6', 'ionic-post-app-realtime-db-storage', 'ionic-post-app-realtime-db-storage', 'ionic-post-app-realtime-db', 'ionic-post-app-v2', and 'ionic-post-app-v3'. The repository has 1 star, 1 watching, and 4 forks. It also has no releases or packages published. The Languages section shows TypeScript (40.2%), JavaScript (32.6%), CSS (24.1%), and HTML (3.1%).

cederdorff / web-mobile-app-dev Public

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags Go to file Add file Code

cederdorff Button disabled when new post not valid 33a1bdc on 4 Mar 85 commits

_slides Create 05-Firebase&AppDev.pdf 6 months ago

data new project templated added 6 months ago

ionic-camera-app camera app added 6 months ago

ionic-geolocation Update Home.jsx 6 months ago

ionic-post-app-firebase-rest-v0 New post app templates added 6 months ago

ionic-post-app-firebase-rest-v1 uid from number to string 6 months ago

ionic-post-app-firebase-rest-v2 uid from number to string 6 months ago

ionic-post-app-firebase-rest-v3 uid from number to string 6 months ago

ionic-post-app-firebase-rest-v4 uid from number to string 6 months ago

ionic-post-app-firebase-rest-v5 uid from number to string 6 months ago

ionic-post-app-firebase-rest-v6 Clean up and validation. 6 months ago

ionic-post-app-realtime-db-storage Button disabled when new post not valid 6 months ago

ionic-post-app-realtime-db-storage Clean up 6 months ago

ionic-post-app-realtime-db Update PostsPage.jsx 6 months ago

ionic-post-app-v2 v2 added 6 months ago

ionic-post-app-v3 Update UserPage.jsx 6 months ago

About No description, website, or topics provided.

Readme

1 star

1 watching

4 forks

Releases No releases published Create a new release

Packages No packages published Publish your first package

Languages

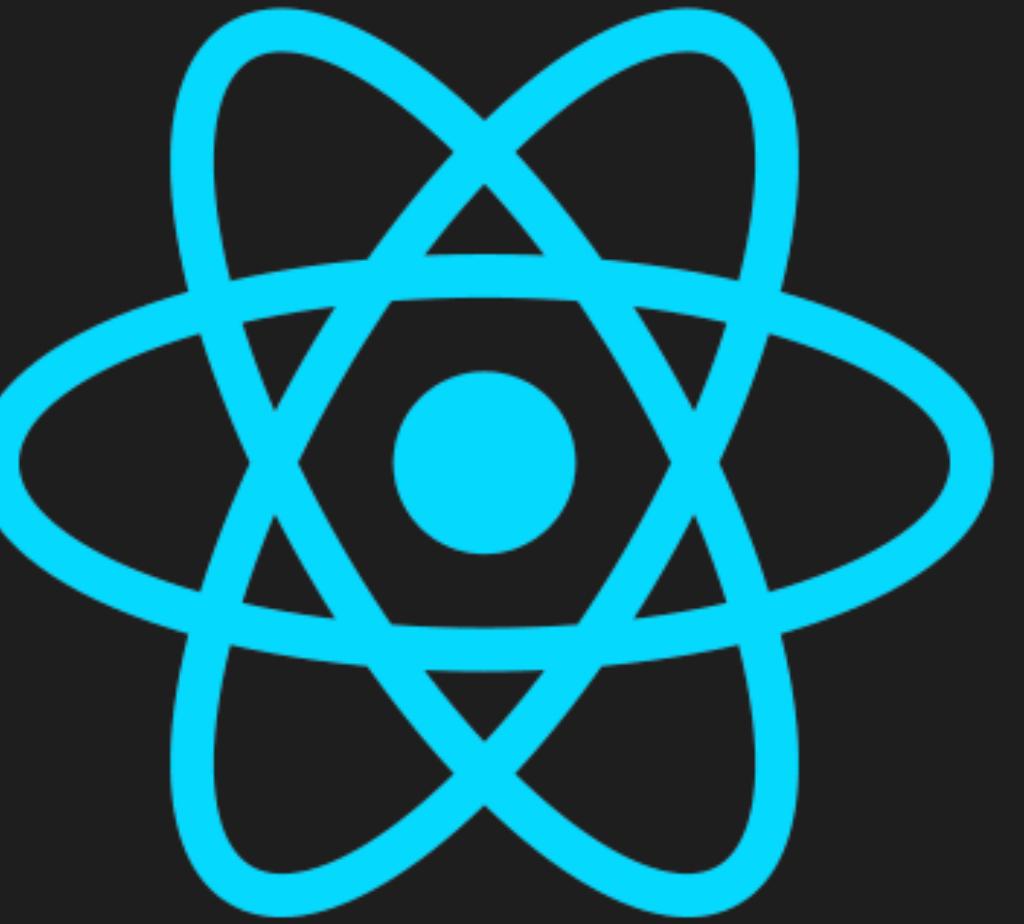
TypeScript 40.2% JavaScript 32.6%

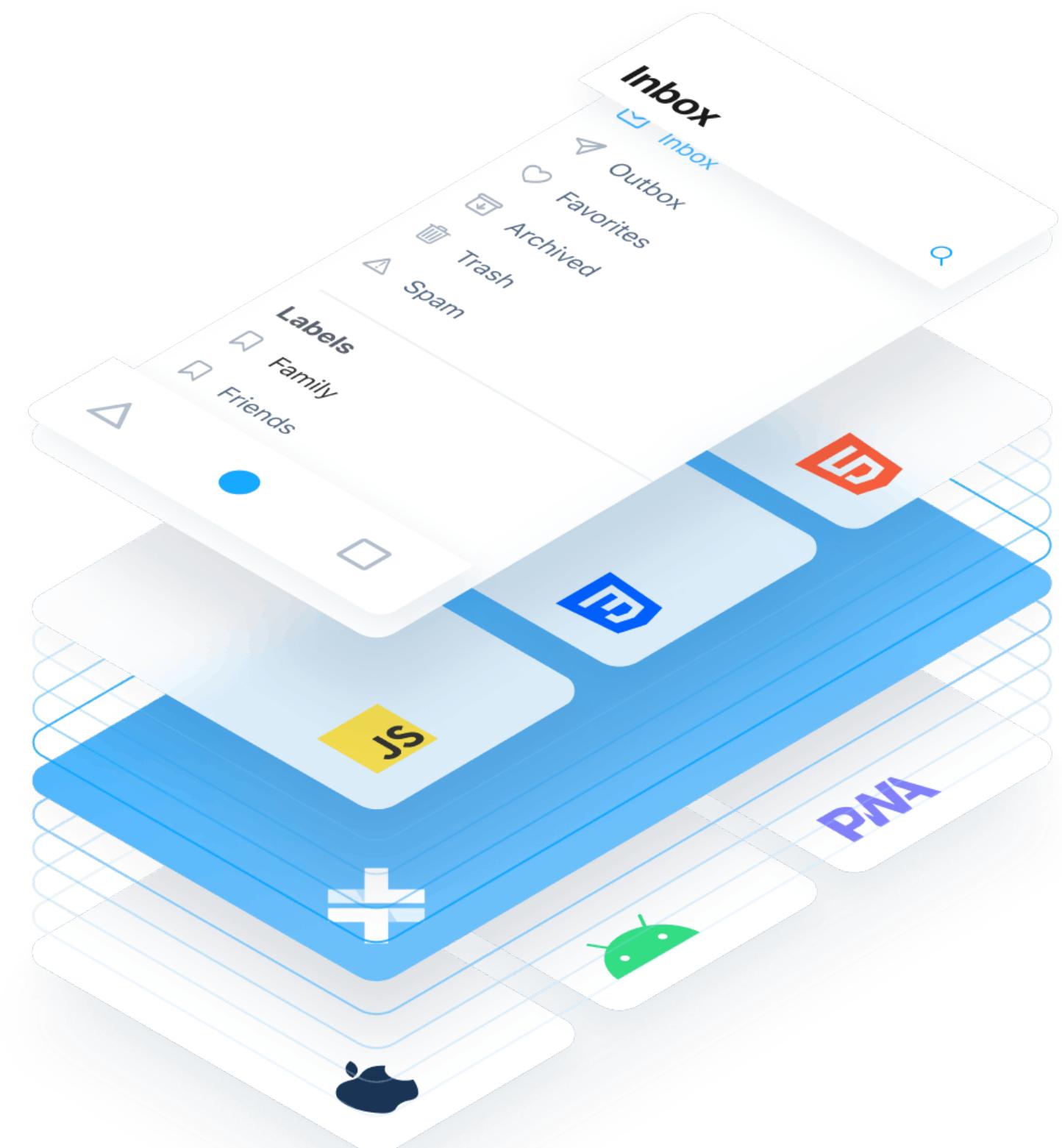
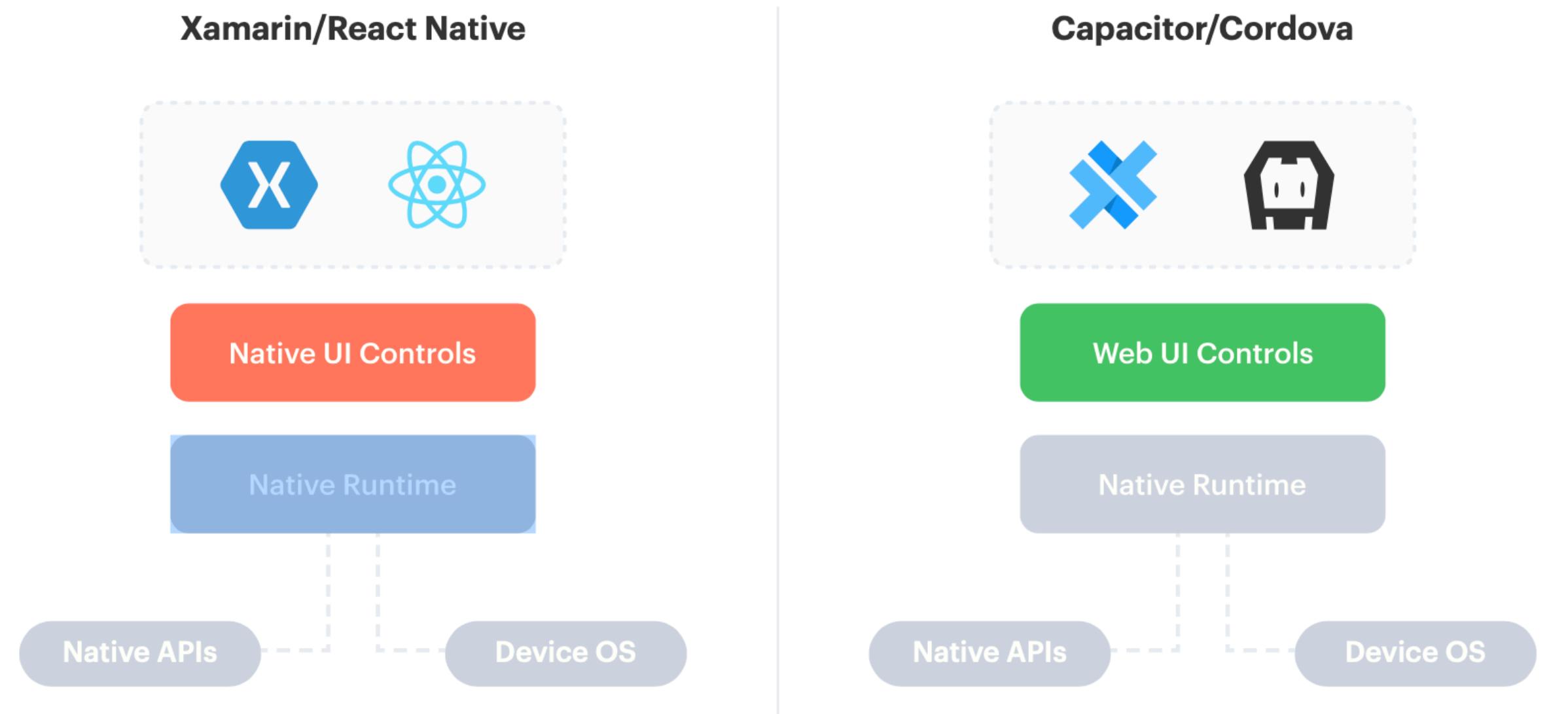
CSS 24.1% HTML 3.1%

<https://github.com/cederdorff/web-mobile-app-dev>

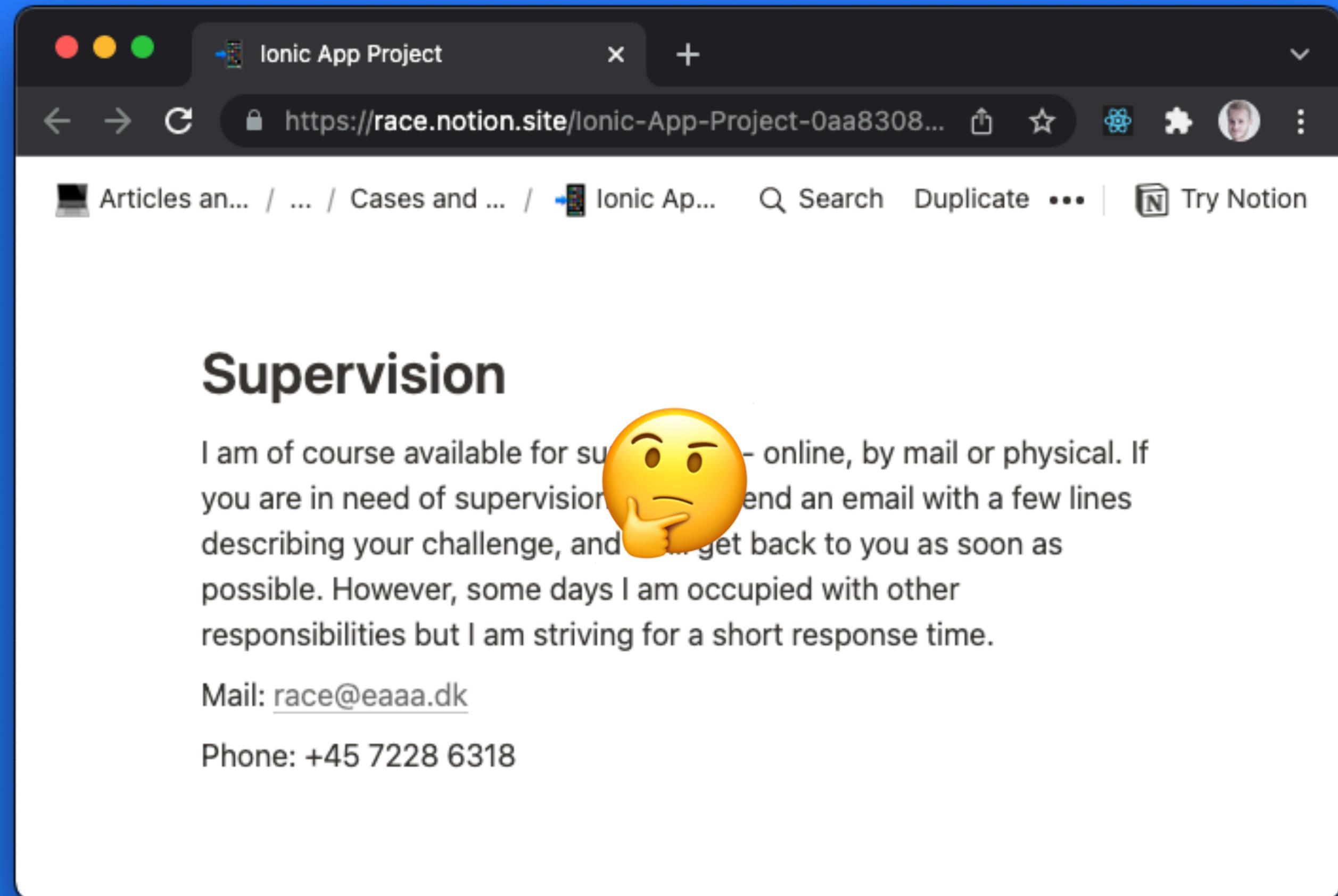
Next Steps

- Practice your skills and competencies with Ionic App Project
- Deploying a PWA
- Deploying a Desktop App
- Publish an App through Google Play and/or App Store
- AppFlow
- Games with Ionic and Phaser: @ionic/phaser/react - npm
- React Native
- ... Unity





Supervision 🤝



A screenshot of a web browser window titled "Ionic App Project". The address bar shows the URL <https://race.notion.site/Ionic-App-Project-0aa8308...>. The page content is a Notion page titled "Supervision". The text on the page reads:

I am of course available for supervision – online, by mail or physical. If you are in need of supervision, please send an email with a few lines describing your challenge, and I will get back to you as soon as possible. However, some days I am occupied with other responsibilities but I am striving for a short response time.

Mail: race@eaaa.dk
Phone: +45 7228 6318