



Android Development with Kotlin

Firebase

What is Firebase?

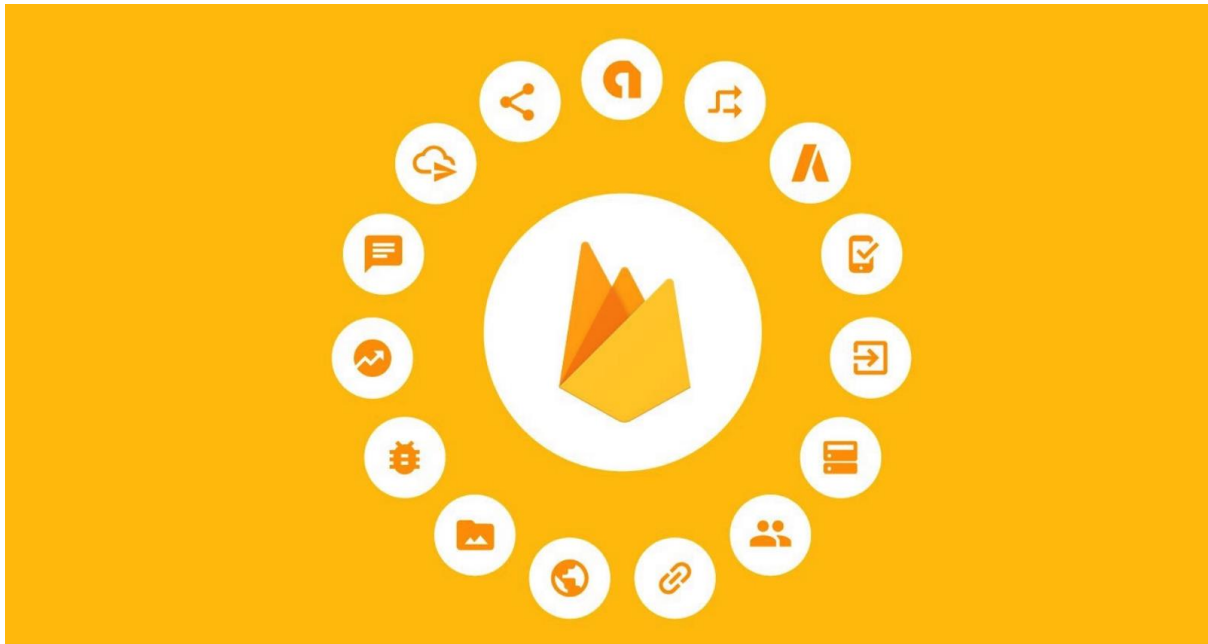


Image Src: <https://i.ytimg.com/vi/iosNuIdQoy8/maxresdefault.jpg>

Firebase is a mobile and web app development platform that provides developers with a plethora of tools and services to help them develop high-quality apps, grow their user base, and earn more profit.

Firestore Services:

Firebase Services can be divided into two groups:

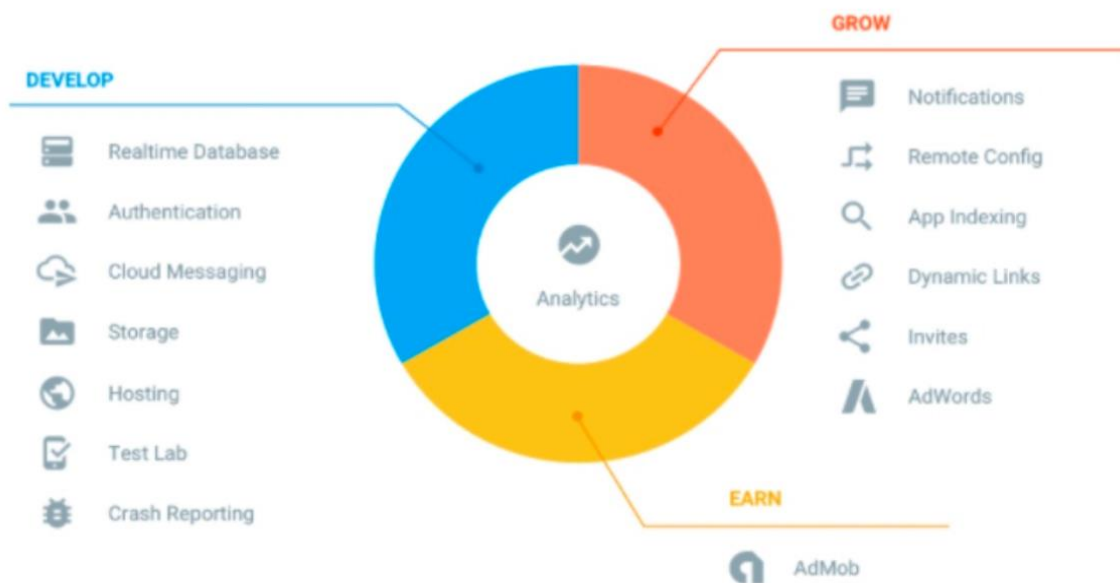


Image src: https://hackernoon.com/hn-images/1*rIDMNpRx8HnnY5mhhbXLUa.png

Develop & test your app :-

- Realtime Database
- Auth
- Test Lab
- Crashlytics
- Cloud Functions
- Firestore
- Cloud Storage
- Performance Monitoring
- Crash Reporting
- Hosting

Grow & Engage your audience:-

- Firebase Analytics
- Invites
- Cloud Messaging
- Predictions
- AdMob
- Dynamic Links
- Adwords
- Remote Config
- App Indexing

Realtime Database :

The Firebase Realtime Database is a cloud-hosted NoSQL database that lets you store and sync between your users in real time.

The Realtime Database is really just one big JSON object that the developers can manage in real time.



Realtime Database => A Tree of Values

With just a single API, the Firebase database provides your app with both the current value of the data and any updates to that data.



Realtime syncing makes it easy for your users to access their data from any device, be it web or mobile. Realtime Database also helps your users collaborate with one another.

Another amazing benefit of Realtime Database is that it ships with mobile and web SDKs, allowing you to build your apps without the need for servers.

When your users go offline, the Realtime Database SDKs use local cache on the device to serve and store changes. When the device comes online, the local data is automatically synchronized.

The Realtime Database can also integrate with Firebase Authentication to provide a simple and intuitive authentication process.

Firestore Database Query:

Firestore has simplified the process of retrieving specific data from the database through queries. Queries are created by chaining together one or more filter methods.

Firestore has 4 ordering functions:

- `orderByKey()`
- `orderByChild('child')`
- `orderByValue()`
- `orderByPriority()`

Note that you will only receive data from a query if you have used the `on()` or `once()` method. You can also use these advanced querying functions to further restrict data:

- `startAt('value')`
- `endAt('value')`
- `equalTo('child_key')`
- `limitToFirst(10)`
- `limitToLast(10)`

In SQL, the basics of querying involve two steps. First, you select the columns from your table. Here I am selecting the Users column. Next, you can apply a restriction to your query using the WHERE clause.

You can also use the LIMIT clause, which will restrict the number of results that you will get back from your query.

How to Store Data? => Firebase Storage:

Firebase Storage is a standalone solution for uploading user-generated content like images and videos from an iOS and Android device, as well as the Web.

Firebase Storage is designed specifically to **scale your apps, provide security, and ensure network resiliency.**

Firebase Storage uses a simple folder/file system to structure its data.

```
var storageRef = firebase.storage.ref("folderName/file.jpg");
var fileUpload = document.getElementById("fileUpload");
fileUpload.on('change',
  function(evt) {
    var firstFile = evt.target.files[0]; // get the first file uploaded
    var uploadTask = storageRef.put(firstFile);
  }
);
```

Firebase Test Labs

Firebase Test Labs provides a large number of mobile test devices to help you test your apps. Firebase Test Labs comes with 3 modes of testing:

Instrumentation Test

These are tests that you written specifically to test your app, using frameworks like Espresso and UI Automator 2.0

- **Robo Test**

This test is for people who just want to relax and let Firebase worry about tests. Firebase Test Labs can simulate user touch and see how each component of the app functions.

- **Game Loop Test**

Test Labs support game app testing. It comes with a beta support for using a “demo mode” where the game app runs while simulating the actions of the player.

Authentication:



Firebase Authentication provides backend services, easy-to-use SDKs, and ready-made UI libraries to authenticate users to your app.

Normally, it would take you months to set up your own authentication system. And even after that, you would need to keep a dedicated team to maintain that system. But if you use Firebase, you can set up the entire system in under 10 lines of code that will handle everything for you, including complex operations like account merging.

You can authenticate your app's users through the following methods:

- Email & Password
- Phone numbers
- Google
- Facebook
- Twitter
- & more!

Using Firebase Authentication makes building secure authentication systems easier, while also improving the sign-in and onboarding experience for end users.

Firebase Authentication is built by the same people who created Google Sign-in, Smart Lock, and Chrome Password Manager.

Firestore Cloud Messaging (FCM):

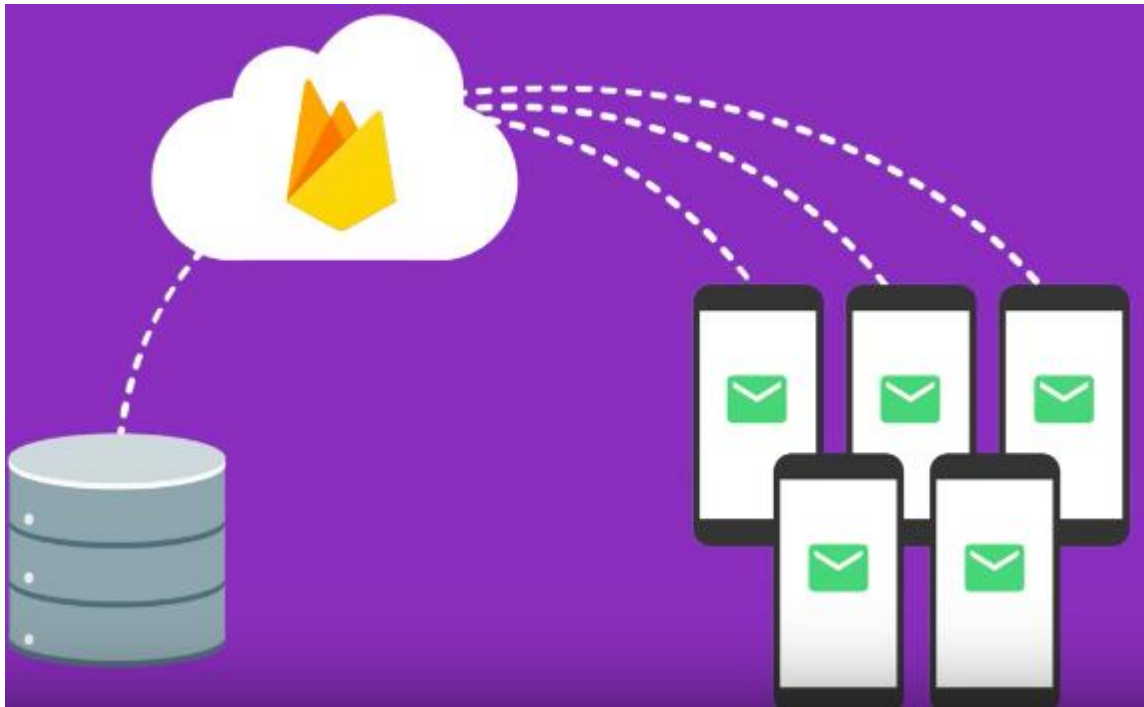


Image Src: https://hackernoon.com/hn-images/1*HtZwrGjaV2kJ4tu9D_phuw.png

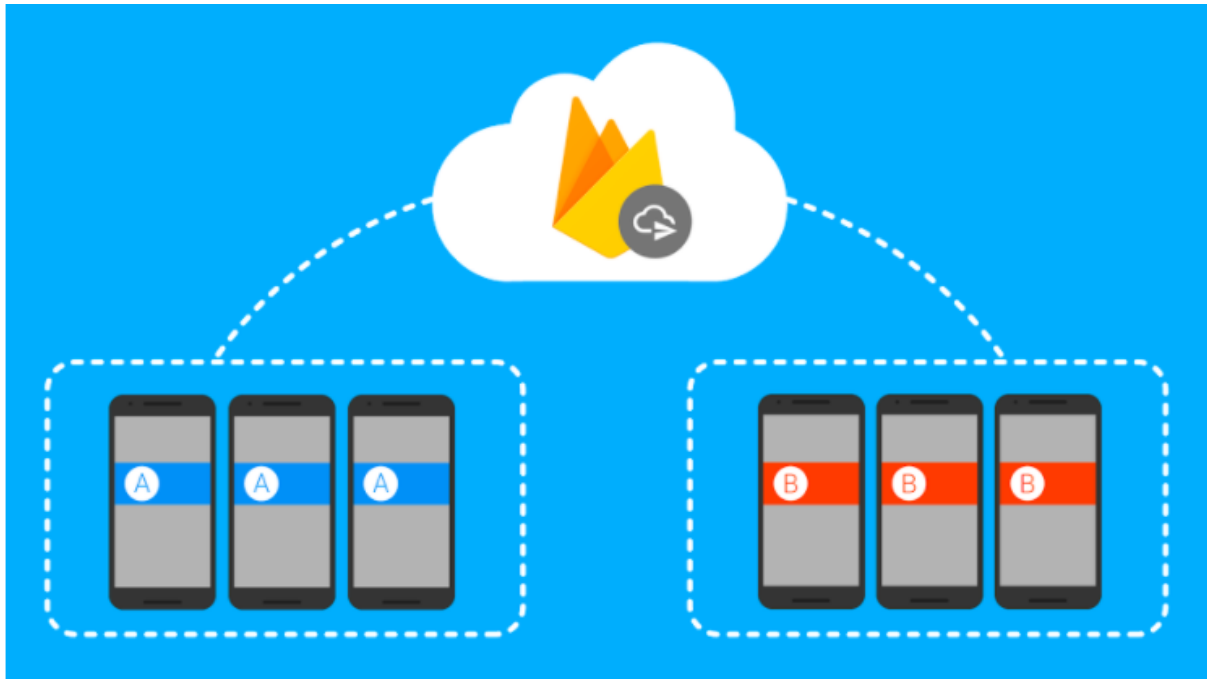
Firestore Cloud Messaging (FCM) provides a reliable and battery-efficient connection between your server and devices that allows you to deliver and receive messages and notifications on iOS, Android, and the web at no cost.

You can send notification messages (2KB limit) and data messages (4KB limit).

Using FCM, you can easily target messages using predefined segments or create your own, using demographics and behavior. You can send messages to a group of devices that are subscribed to specific topics, or you can get as granular as a single device.

FCM can deliver messages instantly, or at a future time in the user's local time zone. You can send custom app data like setting priorities, sounds, and expiration dates, and also track custom conversion events.

The best thing about FCM is that there is hardly any coding involved! FCM is completely integrated with Firestore Analytics, giving you detailed engagement and conversion tracking.



You can also use A/B testing to try out different versions of your notification messages, and then select the one which performs best against your goals.

Remote Config:

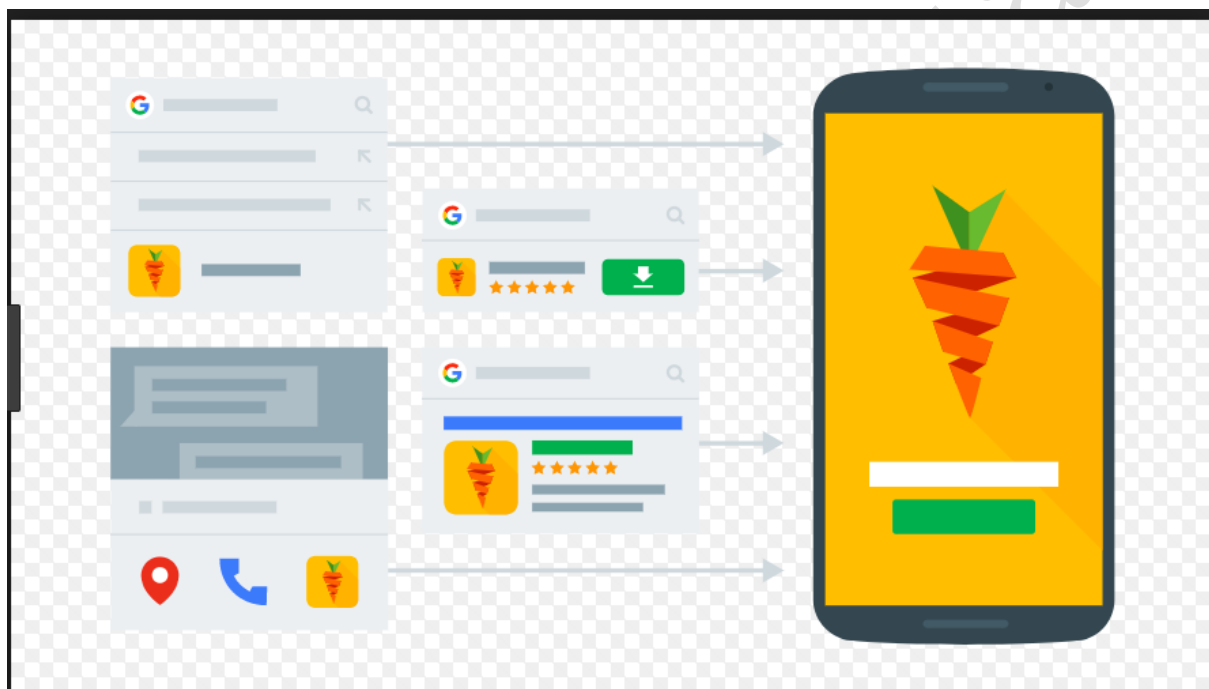


Remote config essentially allows us to publish updates to our users immediately. Whether we wish to change the color scheme for a screen, the layout for a particular section in our app or show promotional/seasonal options—this is completely doable using the server side parameters without the need to publish a new version.

Remote Config gives us the power to:

- Quickly and easily update our applications without the need to publish a new build to the app/play store.
- Effortlessly set how a segment behaves or looks in our application based on the user/device that is using it.

Firestore App Indexing :



To get your app's content indexed by Google, use the same URLs in your app that you use on your website and verify that you own both your app and your website. Google Search crawls the links on your website and serves them in Search results. Then, users who've installed your app on their devices go directly to the content in your app when they click on a link.

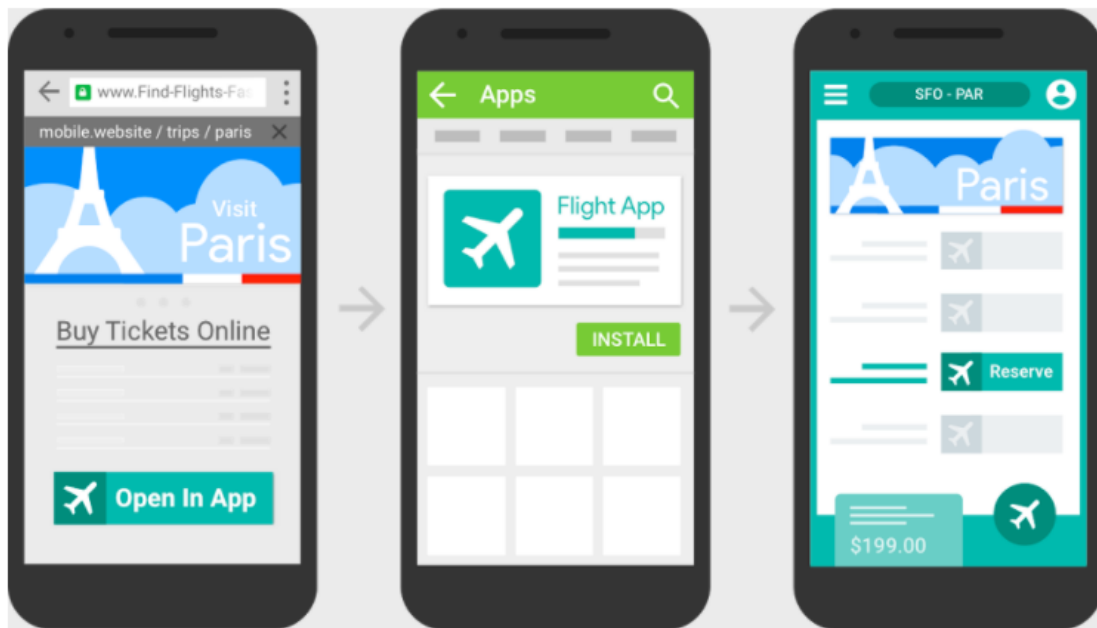
Firestore Dynamic Links:

Deep links are URLs that take you to a content. Most web links are deep links.

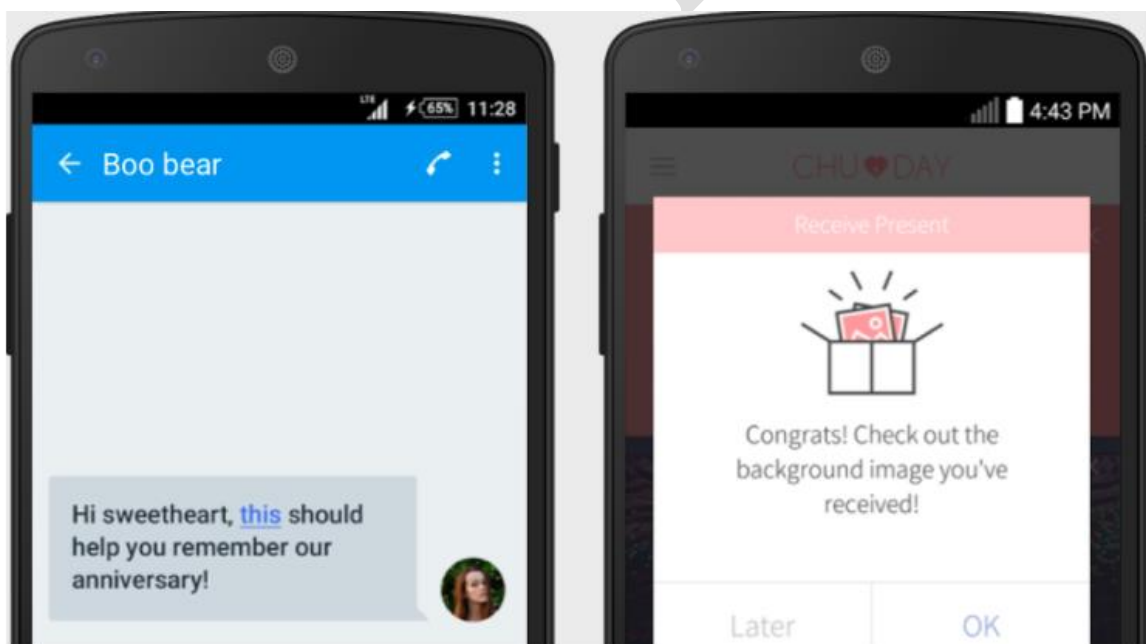
Firestore can now modify deep links into Dynamic Links! Dynamic Links allow the user to directly come to a particular location in your app.

There are 3 fundamental uses for Dynamic Links

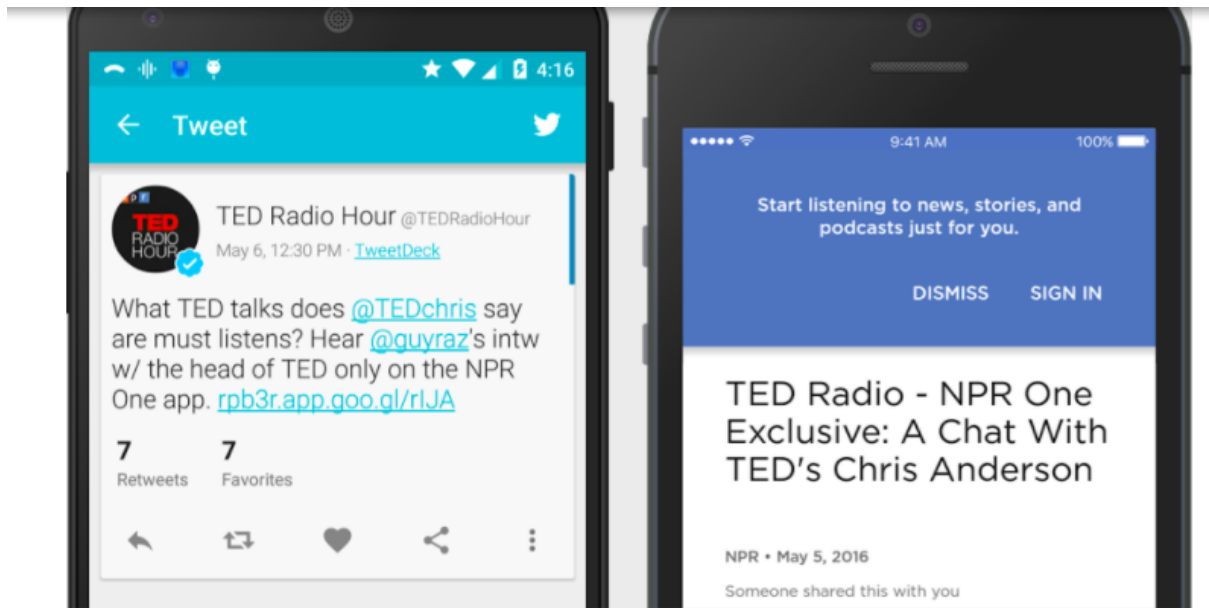
- Convert Mobile Web Users to Native App Users.



- Increase conversion for user-to-user sharing. By converting your app's users, when the app is shared with other users you can skip the generic message which is shown when a user downloads it from the store. Instead, you can show them personalised greeting message.



- Drive installs from the third party. You can use social media networks, email, and SMS can be used to increase your target audience. When users install the app, they can see the exact content of your campaigns.



Firestore :

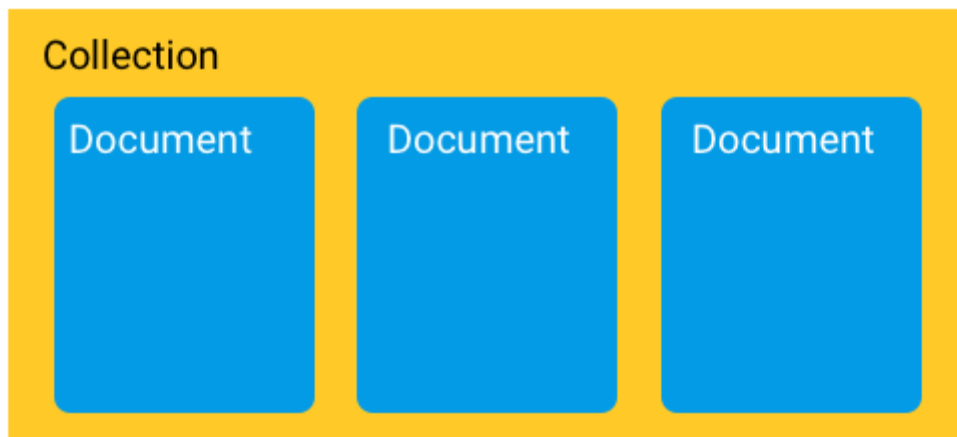


Cloud Firestore is a NoSQL document database that lets you easily store, sync, and query data for your mobile and web apps—at a global scale.

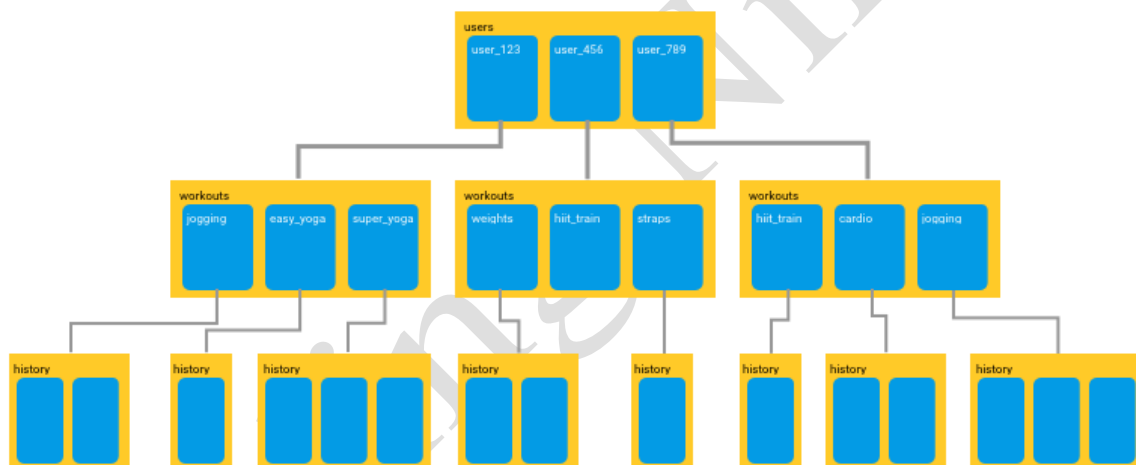
Though this may sound like something similar to the Realtime Database, Firestore brings many new things to the platform that makes it into something completely different from Realtime Database.

Improved Querying and Data Structure:

Where Realtime Database stores data in the form of a giant JSON tree, Cloud Firestore takes a much more structured approach. Firestore keeps its data inside objects called documents. These documents consist of key-value pairs and can contain any kind of data, from strings to binary data to even objects that resemble JSON trees (Firestore calls it as maps). The documents, in turn, are grouped into collections.

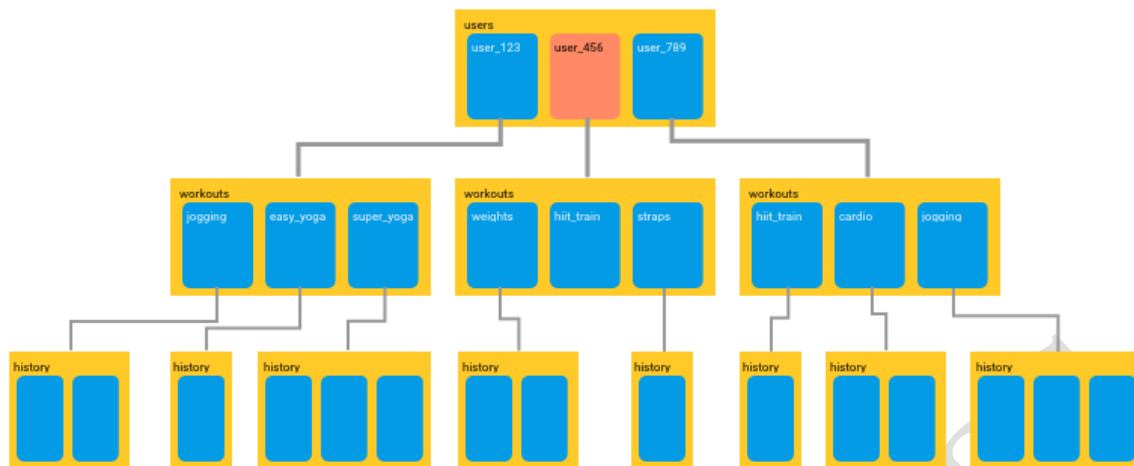


Firestore database can consist of multiple collections that can contain documents pointing towards sub-collections. These sub-collections can again contain documents that point to other sub-collections, and so on.



You can build hierarchies to store related data and easily retrieve any data that you need using queries. All queries can scale with the size of your result set, so your app is ready to scale from its first day itself.

Firestore's queries are *shallow*. By this, I mean to say that in Firestore, you can simply fetch any document that you want without having to fetch all of the data that is contained in any of its linked sub-collections.



You can fetch a single document without having to grab any of its sub-collections

Query with Firestore:

Imagine that you have created a collection in Firestore that contains a list of Cities. So, before you can send out a query, you will have to store the database inside a variable.

```
var citiesRef = db.collection("cities");
```

Here, citiesRef is that variable that contains your collection of cities. Now, if you want to find a list of capital cities, you would write a query like this:

```
var query = citiesRef.where("capital", "==", true)
```

Here's another example of queries in Firestore. Say you want to see only 2 of cities from your database whose population is more than 100,000.

```
citiesRef.where("population", ">", 100000).orderBy("population").limit(2)
```

But Cloud Firestore can make querying even easier! In some cases, Cloud Firestore can automatically search your database across multiple fields. Firestore will guide you towards automatically building an index that will help Firestore to make querying extremely simple.

Index	Collection(s)	Fields
	cities	↑ state ↑ population

CANCEL **CREATE INDEX**

Creation time depends on the size of your data

Better Scalability

Though Firebase's Realtime Database is capable of scaling, things will start to get crazy when your app becomes really popular or if your database becomes really massive. Cloud Firestore is based on Google's Cloud infrastructure. This allows it to scale much more easily and to a greater capacity than the Realtime Database.

Multi-Region Database

In Firestore, your data is automatically copied to various regions. So if one data center goes offline due to some unforeseen reason, you can be sure that your app's data is still safe somewhere else.

Firestore's multi-region database also provides strong consistency. Any changes to your data will be mirrored across every copy of your database.

Different Pricing Model

The Realtime Database charges its users based on the amount of data that you have stored in the database.

Cloud Firestore also charges you for the same, but the cost is significantly lower than that of Realtime Database and instead of basing the cost on the amount of data stored, Firestore's pricing is driven by the number of reads/writes that you perform.

We have covered all the topics in the firebase lecture and notes. Feel free to explore more using the below links:

Firestore: <https://firebase.google.com/>

Firestore Article: <https://medium.com/firebase-developers/what-is-firebase-the-complete-story-abridged-bcc730c5f2c0>

Coding Ninjas