



THE UNIVERSITY OF
CHICAGO



MPCS 51033 • AUTUMN 2017 • SESSION 5

BACKENDS FOR MOBILE APPLICATIONS

CLASS NEWS

CLASS NEWS

- Assignment 1&2 graded. Look for grade.txt file in repositories.
- Office hours Thursday 10AM in Young 308



#general

Andrew 10:37 AM added an integration to this channel: [twitter](#)

Andrew 9:09 PM Hi. I'm not going to have any formal office hours tomorrow but let me know if you have any questions. I plan on having regular office hours Thursday around 10 starting next week.

Thursday, September 28th

Andrew 10:58 AM <https://cloudplatform.googleblog.com/2016/10/introducing-Google-Cloud-Shells-new-code-editor.html?m=1>

Google Cloud Platform Blog [Introducing Google Cloud Shell's new code editor](#)
Posted by Sachin Kotwani, Product Manager We've heard from a lot of Google Cloud Platform (GCP) users that they like to edit code and c... (9kB)

Yi Wang 5 hours ago If it worked with your gmail, then we can go ahead and try it. I'll take a look in class before we actually start using it to make sure it looks ok. I will send an email to my contact at google just to double check.

Yi Wang 5 hours ago OK. Thank you!

Reply...

Here's an example of how you can use the Cloud Shell code editor to create a sample app, push your changes to Google Cloud Source Repository, deploy the app to Google App Engine Standard, and use Stackdriver Debugger.

Andrew 11:15 AM Message #general

CLASS NEWS

- Assignment 1 & 2 graded
- This assignment will be 2 week, but plan accordingly
- Halloween makeup class options?

The screenshot shows the Slack interface. On the left is the sidebar with channels like All Threads, Channels, # assignment1, # general (which is selected and highlighted in green), # random, Direct Messages, Apps, and a list of users including slackbot, Andrew (you), acham1, Xuefeng, and Yi Wang. A '+' icon is at the bottom of the sidebar.

#general

Andrew 10:37 AM added an integration to this channel: [twitter](#)

Andrew 9:09 PM Hi. I'm not going to have any formal office hours tomorrow but let me know if you have any questions. I plan on having regular office hours Thursday around 10 starting next week.

Thursday, September 28th

Andrew 10:58 AM <https://cloudplatform.googleblog.com/2016/10/introducing-Google-Cloud-Shells-new-code-editor.html?m=1>

Google Cloud Platform Blog [Introducing Google Cloud Shell's new code editor](#)
Posted by Sachin Kotwani, Product Manager We've heard from a lot of Google Cloud Platform (GCP) users that they like to edit code and c... (9kB)

Here's an example of how you can use the Cloud Shell code editor to create a sample app, push your changes to Google Cloud Source Repository, deploy the app to Google App Engine Standard, and use Stackdriver Debugger.

Andrew 11:15 AM

Message #general @ 😊

Thread

Yi Wang and you

Yi Wang Today at 10:56 AM Direct message
I received the coupon email in my uchicago email box, but when I clicked the url to redeem, it redirected to a page that is logged in with my gmail. I didn't notice that and finished the whole process.

2 replies

Andrew 6 hours ago If it worked with your gmail, then we can go ahead and try it. I'll take a look in class before we actually start using it to make sure it looks ok. I will send an email to my contact at google just to double check.

Yi Wang 5 hours ago OK. Thank you!

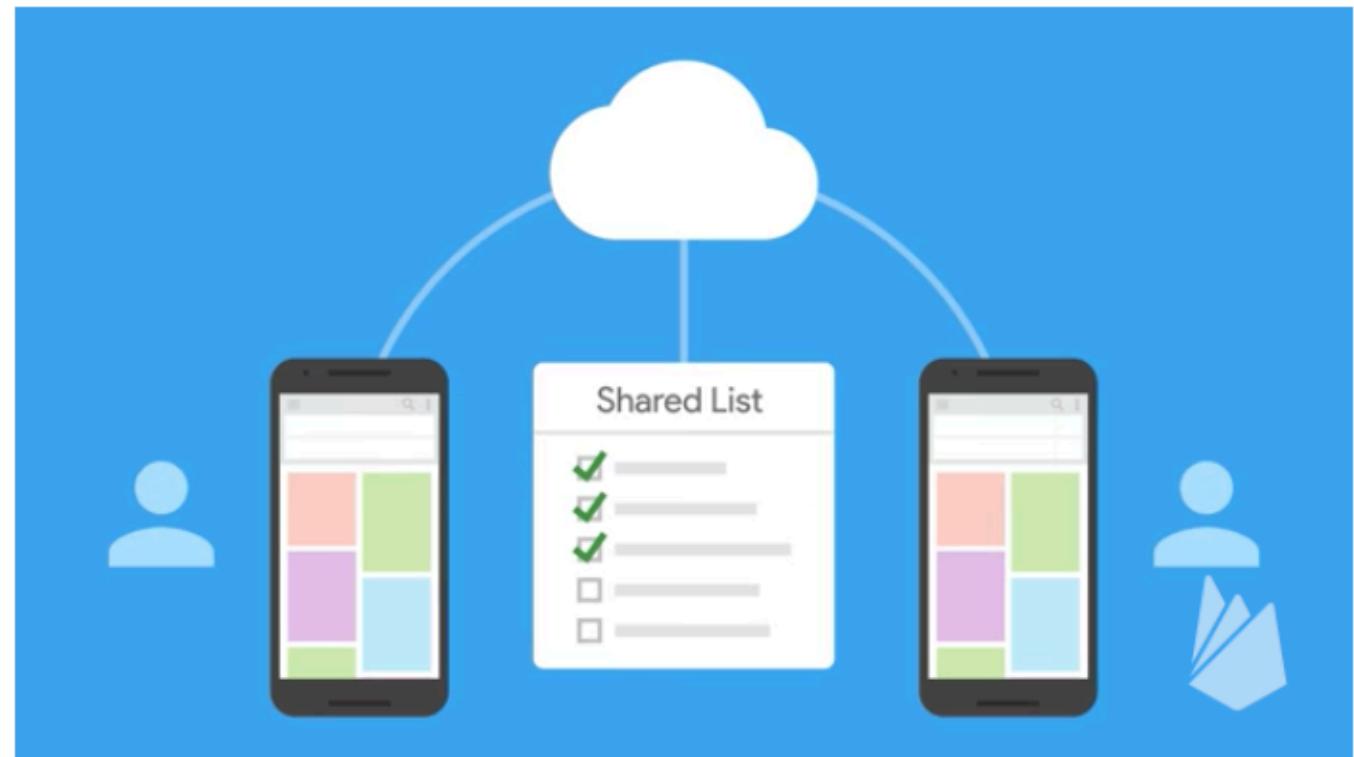
Reply... 😊

FIRESTORE



FIRESTORE

- https://youtu.be/QcsAb2RR52c?list=PLI-K7zZEesYLmOF_07layrTntevxtbUxDL



FIRESTORE

- Strong parity with realtime database
- Better
 - Querying
 - Scaling

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.
Expressive querying	In Cloud Firestore, you can use queries to retrieve individual, specific documents or to retrieve all the documents in a collection that match your query parameters. Your queries can include multiple, chained filters and combine filtering and sorting. They're also indexed by default, so query performance is proportional to the size of your result set, not your data set.
Realtime updates	Like Realtime Database, Cloud Firestore uses data synchronization to update data on any connected device. However, it's also designed to make simple, one-time fetch queries efficiently.
Offline support	Cloud Firestore caches data that your app is actively using, so the app can write, read, listen to, and query data even if the device is offline. When the device comes back online, Cloud Firestore synchronizes any local changes back to Cloud Firestore.
Designed to scale	Cloud Firestore brings you the best of Google Cloud Platform's powerful infrastructure: automatic multi-region data replication, strong consistency guarantees, atomic batch operations, and real transaction support. We've designed Cloud Firestore to handle the toughest database workloads from the world's biggest apps.

FIRESTORE

- Add to Podfile
- `pod install`

```
platform :ios, '9.0'

target 'FireChat' do
  # Comment the next line if you're not using Swift and don't want to use dynamic
  # frameworks
  use_frameworks!

  # Pods for FireChat
  pod 'Firebase/Core'
  pod 'Firebase/Database'

  pod 'Firebase/Auth'
  pod 'GoogleSignIn'

  pod 'Firebase/Crash'

  pod 'Firebase/Storage'
  pod 'Firebase/Invites'
  pod 'Firebase/Messaging'
  pod 'Firebase/DynamicLinks'

  pod 'Firebase/RemoteConfig'
  pod 'Firebase/Firestore'

  pod 'SDWebImage'

end
```

FIRESTORE

- Add to Podfile
- `pod install`

```
platform :ios, '9.0'

target 'FireChat' do
  # Comment the next line if you're not using Swift and don't want to use dynamic
  # frameworks
  use_frameworks!

  # Pods for FireChat
  pod 'Firebase/Core'
  pod 'Firebase/Database'

  pod 'Firebase/Auth'
  pod 'GoogleSignIn'

  pod 'Firebase/Crash'

  pod 'Firebase/Storage'
  pod 'Firebase/Invites'
  pod 'Firebase/Messaging'
  pod 'Firebase/DynamicLinks'

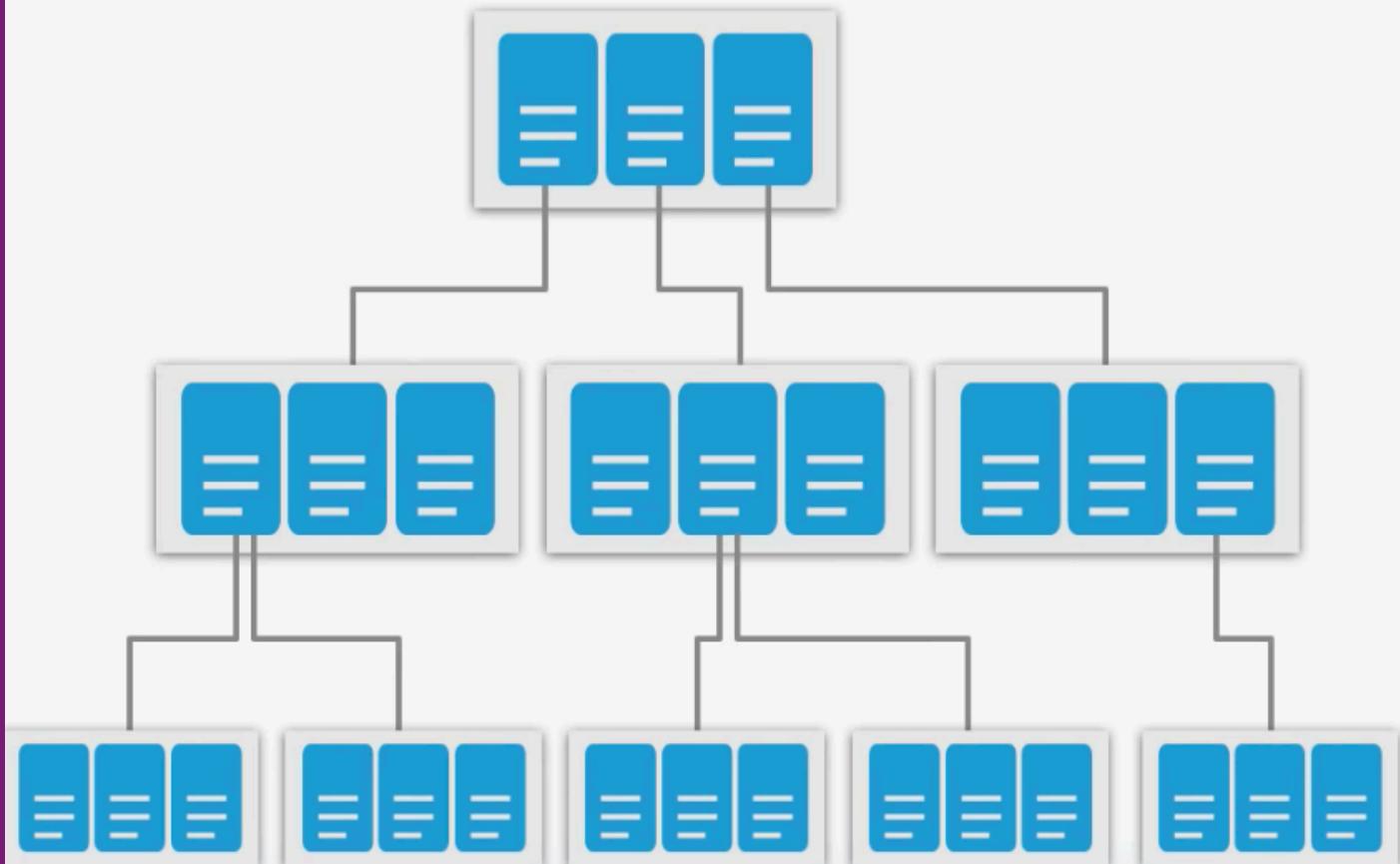
  pod 'Firebase/RemoteConfig'
  pod 'Firebase/Firestore'

  pod 'SDWebImage'

end
```

FIRESTORE

- Data is stored as documents and collections



FIRESTORE

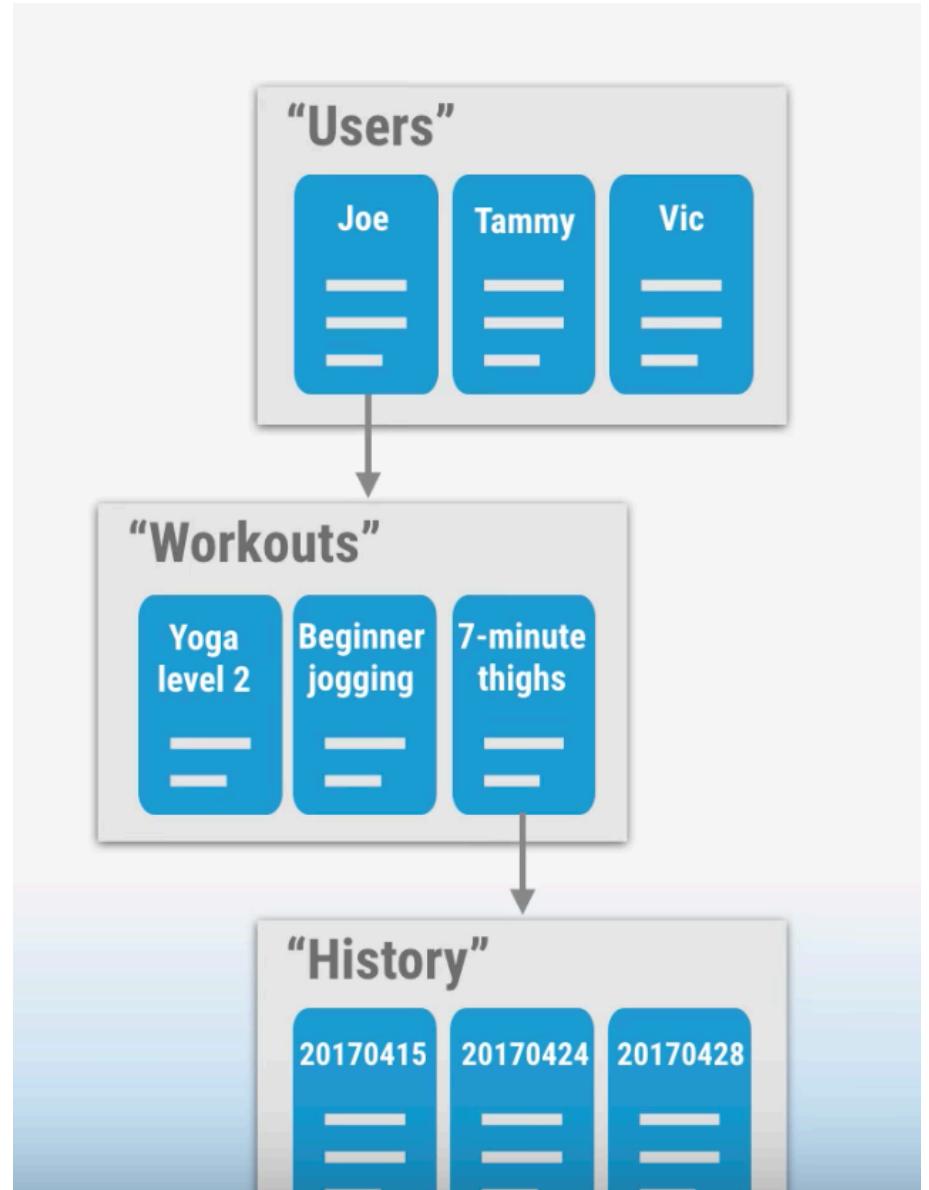
- Document is just key-value pairs

Document

```
bird_type: "swallow"
airspeed: 42.733
coconut_capacity: 0.62
isNative: false
icon: <binary data>
vector:
  {x: 36.4255,
   y: 25.1442,
   z: 18.8816}
distances_traveled:
  [42, 39, 12, 42]
```

FIRESTORE

- Collections are groups of documents
- Collections can only contain documents
- Documents can point to other collections
- The Root of your application has to be a collection



USING FIRESTORE

USING FIRESTORE

- Firestore can be used along side realtime database
- Security rules are private by default

The screenshot shows the Firebase console interface for Cloud Firestore. At the top, there's a navigation bar with the Firebase logo, a dropdown for 'FireChat', and links for 'Go to docs' and a user icon. Below the navigation is a blue header bar with tabs for 'Database', 'Cloud Firestore' (marked as 'BETA'), 'DATA', 'RULES', 'INDEXES', and 'USAGE'. A modal window titled 'Security rules for Cloud Firestore' is open. It contains instructions: 'Once you have defined your data structure you will have to write rules to secure your data.' with a 'Learn more' link. Two radio button options are shown: 'Start in locked mode' (selected) and 'Start in test mode'. The 'locked mode' option describes making the database private by denying all reads and writes. The 'test mode' option describes allowing all reads and writes. To the right of the modal, a code block shows the Firestore security rules for 'locked mode':

```
service cloud.firestore {  
  match /databases/{database}/documents {  
    match /{document=**} {  
      allow read, write: if false;  
    }  
  }  
}
```

A note below the code states: 'All third party reads and writes will be denied'. At the bottom of the modal, a note says: 'Enabling Cloud Firestore Beta will preclude you from using Cloud Datastore with this project, notably from the associated App Engine app.' There are 'CANCEL' and 'ENABLE' buttons at the bottom right.

USING FIRESTORE

```
// Firestore Ref
docRef = Firestore.firestore().document("trending/hot")

// Load some data up in Firestore
let trendingData: [String: Any] = ["group" : "art",
                                    "timestamp" : "today",
                                    "votes" : 200]

docRef.setData(trendingData) { (error) in
    if error != nil {
        print("Data!!")
    }
}
```

USING FIRESTORE

The screenshot shows the Google Cloud Platform Firestore interface. At the top, there's a navigation bar with a home icon, followed by 'trending' and 'hot'. Below this is a table structure representing a document.

firechat-66f87	trending	hot
+ ADD COLLECTION	+ ADD DOCUMENT	+ ADD COLLECTION
trending >	hot >	+ ADD FIELD group: "art" timestamp: "today" votes: 200

USING FIRESTORE

```
/// Get a one time snapshot
func oneTimeSnapshot() {
    docRef.getDocument { (docSnapshot, error) in
        guard let docSnapshot = docSnapshot, docSnapshot.exists else {return}
        let data = docSnapshot.data()
        self.trendingLabel.text = data["group"] as? String ?? ""
    }
}
```

USING FIRESTORE

```
/// Set up an observer
func observeDocument() {
    self.listenerRef = docRef.addSnapshotListener { (docSnapshot, error) in
        guard let docSnapshot = docSnapshot, docSnapshot.exists else {return}
        let data = docSnapshot.data()
        self.trendingLabel.text = data["group"] as? String ?? ""
    }
}

override func viewWillDisappear(_ animated: Bool) {
    super.viewWillDisappear(animated)
    self.listenerRef.remove()
}
```

USING FIRESTORE

App Engine		Quotas	VIEW USAGE HISTORY																																																																																																																																																								
 Dashboard	<p>The quota details for this application are grouped by API and are listed below. If your application exceeds 50% of any particular quota halfway through the day, it may exceed the quota before the day is over. To learn more about how quotas work, read Understanding Quotas ↗ and Why is My App Over Quota? ↗</p> <h4>Storage</h4> <table><thead><tr><th>Resource</th><th>Usage today</th><th>Daily quota</th><th>Per-minute quota</th><th>Rate limit status</th></tr></thead><tbody><tr><td>Cloud Storage Class B Operations</td><td>0 of 0.05</td><td><div style="width: 0%;"><div style="width: 100%;">0%</div></div></td><td>--</td><td>Standard rate</td></tr><tr><td>Cloud Storage Class A Operations</td><td>0 of 0.02</td><td><div style="width: 0%;"><div style="width: 100%;">0%</div></div></td><td>--</td><td>Standard rate</td></tr><tr><td>Cloud Storage Standard Storage</td><td>0.00098 of 5 GB</td><td><div style="width: 0%;"><div style="width: 100%;">0%</div></div></td><td>--</td><td>Standard rate</td></tr><tr><td>Cloud Storage Network (Egress) - Americas and EMEA</td><td>0.000001 of 1 GB</td><td><div style="width: 0%;"><div style="width: 100%;">0%</div></div></td><td>--</td><td>Standard rate</td></tr><tr><td>Firebase Read Operations</td><td>0.000023 of 0.05 Million Ops</td><td><div style="width: 0%;"><div style="width: 100%;">0%</div></div></td><td>--</td><td>Standard rate</td></tr><tr><td>Firebase Small Operations</td><td>0.000003 of 0.05 Million Ops</td><td><div style="width: 0%;"><div style="width: 100%;">0%</div></div></td><td>--</td><td>Standard rate</td></tr><tr><td>Firebase API Calls</td><td>81</td><td>--</td><td>--</td><td>Standard rate</td></tr><tr><td>Data Sent to Firebase API</td><td>0.000001 GB</td><td>--</td><td>--</td><td>Standard rate</td></tr><tr><td>Firebase Entity Fetch Ops</td><td>12</td><td>--</td><td>--</td><td>Standard rate</td></tr><tr><td>Blobstore</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>Memcache</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>Search</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>Settings</td><td></td><td></td><td></td><td></td><td></td></tr></tbody></table>	Resource					Usage today	Daily quota	Per-minute quota	Rate limit status	Cloud Storage Class B Operations	0 of 0.05	<div style="width: 0%;"><div style="width: 100%;">0%</div></div>	--	Standard rate	Cloud Storage Class A Operations	0 of 0.02	<div style="width: 0%;"><div style="width: 100%;">0%</div></div>	--	Standard rate	Cloud Storage Standard Storage	0.00098 of 5 GB	<div style="width: 0%;"><div style="width: 100%;">0%</div></div>	--	Standard rate	Cloud Storage Network (Egress) - Americas and EMEA	0.000001 of 1 GB	<div style="width: 0%;"><div style="width: 100%;">0%</div></div>	--	Standard rate	Firebase Read Operations	0.000023 of 0.05 Million Ops	<div style="width: 0%;"><div style="width: 100%;">0%</div></div>	--	Standard rate	Firebase Small Operations	0.000003 of 0.05 Million Ops	<div style="width: 0%;"><div style="width: 100%;">0%</div></div>	--	Standard rate	Firebase API Calls	81	--	--	Standard rate	Data Sent to Firebase API	0.000001 GB	--	--	Standard rate	Firebase Entity Fetch Ops	12	--	--	Standard rate	Blobstore						Memcache						Search						Settings						<p>The quota details for this application are grouped by API and are listed below. If your application exceeds 50% of any particular quota halfway through the day, it may exceed the quota before the day is over. To learn more about how quotas work, read Understanding Quotas ↗ and Why is My App Over Quota? ↗</p> <h4>Storage</h4> <table><thead><tr><th>Resource</th><th>Usage today</th><th>Daily quota</th><th>Per-minute quota</th><th>Rate limit status</th></tr></thead><tbody><tr><td>Cloud Storage Class B Operations</td><td>0 of 0.05</td><td><div style="width: 0%;"><div style="width: 100%;">0%</div></div></td><td>--</td><td>Standard rate</td></tr><tr><td>Cloud Storage Class A Operations</td><td>0 of 0.02</td><td><div style="width: 0%;"><div style="width: 100%;">0%</div></div></td><td>--</td><td>Standard rate</td></tr><tr><td>Cloud Storage Standard Storage</td><td>0.00098 of 5 GB</td><td><div style="width: 0%;"><div style="width: 100%;">0%</div></div></td><td>--</td><td>Standard rate</td></tr><tr><td>Cloud Storage Network (Egress) - Americas and EMEA</td><td>0.000001 of 1 GB</td><td><div style="width: 0%;"><div style="width: 100%;">0%</div></div></td><td>--</td><td>Standard rate</td></tr><tr><td>Firebase Read Operations</td><td>0.000023 of 0.05 Million Ops</td><td><div style="width: 0%;"><div style="width: 100%;">0%</div></div></td><td>--</td><td>Standard rate</td></tr><tr><td>Firebase Small Operations</td><td>0.000003 of 0.05 Million Ops</td><td><div style="width: 0%;"><div style="width: 100%;">0%</div></div></td><td>--</td><td>Standard rate</td></tr><tr><td>Firebase API Calls</td><td>81</td><td>--</td><td>--</td><td>Standard rate</td></tr><tr><td>Data Sent to Firebase API</td><td>0.000001 GB</td><td>--</td><td>--</td><td>Standard rate</td></tr><tr><td>Firebase Entity Fetch Ops</td><td>12</td><td>--</td><td>--</td><td>Standard rate</td></tr><tr><td>Blobstore</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>Memcache</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>Search</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>Settings</td><td></td><td></td><td></td><td></td><td></td></tr></tbody></table>						Resource	Usage today	Daily quota	Per-minute quota	Rate limit status	Cloud Storage Class B Operations	0 of 0.05	<div style="width: 0%;"><div style="width: 100%;">0%</div></div>	--	Standard rate	Cloud Storage Class A Operations	0 of 0.02	<div style="width: 0%;"><div style="width: 100%;">0%</div></div>	--	Standard rate	Cloud Storage Standard Storage	0.00098 of 5 GB	<div style="width: 0%;"><div style="width: 100%;">0%</div></div>	--	Standard rate	Cloud Storage Network (Egress) - Americas and EMEA	0.000001 of 1 GB	<div style="width: 0%;"><div style="width: 100%;">0%</div></div>	--	Standard rate	Firebase Read Operations	0.000023 of 0.05 Million Ops	<div style="width: 0%;"><div style="width: 100%;">0%</div></div>	--	Standard rate	Firebase Small Operations	0.000003 of 0.05 Million Ops	<div style="width: 0%;"><div style="width: 100%;">0%</div></div>	--	Standard rate	Firebase API Calls	81	--	--	Standard rate	Data Sent to Firebase API	0.000001 GB	--	--	Standard rate	Firebase Entity Fetch Ops	12	--	--	Standard rate	Blobstore						Memcache						Search						Settings	
Resource	Usage today	Daily quota	Per-minute quota	Rate limit status																																																																																																																																																							
Cloud Storage Class B Operations	0 of 0.05	<div style="width: 0%;"><div style="width: 100%;">0%</div></div>	--	Standard rate																																																																																																																																																							
Cloud Storage Class A Operations	0 of 0.02	<div style="width: 0%;"><div style="width: 100%;">0%</div></div>	--	Standard rate																																																																																																																																																							
Cloud Storage Standard Storage	0.00098 of 5 GB	<div style="width: 0%;"><div style="width: 100%;">0%</div></div>	--	Standard rate																																																																																																																																																							
Cloud Storage Network (Egress) - Americas and EMEA	0.000001 of 1 GB	<div style="width: 0%;"><div style="width: 100%;">0%</div></div>	--	Standard rate																																																																																																																																																							
Firebase Read Operations	0.000023 of 0.05 Million Ops	<div style="width: 0%;"><div style="width: 100%;">0%</div></div>	--	Standard rate																																																																																																																																																							
Firebase Small Operations	0.000003 of 0.05 Million Ops	<div style="width: 0%;"><div style="width: 100%;">0%</div></div>	--	Standard rate																																																																																																																																																							
Firebase API Calls	81	--	--	Standard rate																																																																																																																																																							
Data Sent to Firebase API	0.000001 GB	--	--	Standard rate																																																																																																																																																							
Firebase Entity Fetch Ops	12	--	--	Standard rate																																																																																																																																																							
Blobstore																																																																																																																																																											
Memcache																																																																																																																																																											
Search																																																																																																																																																											
Settings																																																																																																																																																											
Resource	Usage today	Daily quota	Per-minute quota	Rate limit status																																																																																																																																																							
Cloud Storage Class B Operations	0 of 0.05	<div style="width: 0%;"><div style="width: 100%;">0%</div></div>	--	Standard rate																																																																																																																																																							
Cloud Storage Class A Operations	0 of 0.02	<div style="width: 0%;"><div style="width: 100%;">0%</div></div>	--	Standard rate																																																																																																																																																							
Cloud Storage Standard Storage	0.00098 of 5 GB	<div style="width: 0%;"><div style="width: 100%;">0%</div></div>	--	Standard rate																																																																																																																																																							
Cloud Storage Network (Egress) - Americas and EMEA	0.000001 of 1 GB	<div style="width: 0%;"><div style="width: 100%;">0%</div></div>	--	Standard rate																																																																																																																																																							
Firebase Read Operations	0.000023 of 0.05 Million Ops	<div style="width: 0%;"><div style="width: 100%;">0%</div></div>	--	Standard rate																																																																																																																																																							
Firebase Small Operations	0.000003 of 0.05 Million Ops	<div style="width: 0%;"><div style="width: 100%;">0%</div></div>	--	Standard rate																																																																																																																																																							
Firebase API Calls	81	--	--	Standard rate																																																																																																																																																							
Data Sent to Firebase API	0.000001 GB	--	--	Standard rate																																																																																																																																																							
Firebase Entity Fetch Ops	12	--	--	Standard rate																																																																																																																																																							
Blobstore																																																																																																																																																											
Memcache																																																																																																																																																											
Search																																																																																																																																																											
Settings																																																																																																																																																											

REALTIME DATABASE VS. FIRESTORE

REALTIME DATABASE VS. FIRESTORE

Realtime Database is Firebase's original database. It's an efficient, low-latency solution for mobile apps that require synced states across clients in realtime.

Cloud Firestore is Firebase's new flagship database for mobile app development. It improves on the successes of the Realtime Database with a new, more intuitive data model. Cloud Firestore also features richer, faster queries and scales better than the Realtime Database.

- Flagship...gives it all away

REALTIME DATABASE VS. FIRESTORE

Realtime Database

Stores data as one large JSON tree.

- Simple data is very easy to store.
- Complex, hierarchical data is harder to organize at scale.

Learn more about the [Realtime Database data model](#).

Cloud Firestore

Stores data in documents organized in collections.

- Simple data is easy to store in documents, which are very similar to JSON.
- Complex, hierarchical data is easier to organize at scale, using subcollections within documents.
- Requires less denormalization and data flattening.

Learn more about the [Cloud Firestore data model](#).

REALTIME DATABASE VS. FIRESTORE

Realtime Database	Cloud Firestore
<p>Deep queries with limited sorting and filtering functionality.</p> <ul style="list-style-type: none">• You can only sort or filter on a property, not sort <i>and</i> filter on a property, in a single query.• Queries are deep by default: They always return the entire subtree.	<p>Indexed queries with compound sorting and filtering.</p> <ul style="list-style-type: none">• You can chain filters and combine filtering and sorting on a property in a single query.• Write shallow queries for subcollections: You can query subcollections within a document instead of an entire collection, or even an entire document.• Queries are indexed by default: Query performance is proportional to the size of your result set, not your data set.

- Major limitation of Realtime DB

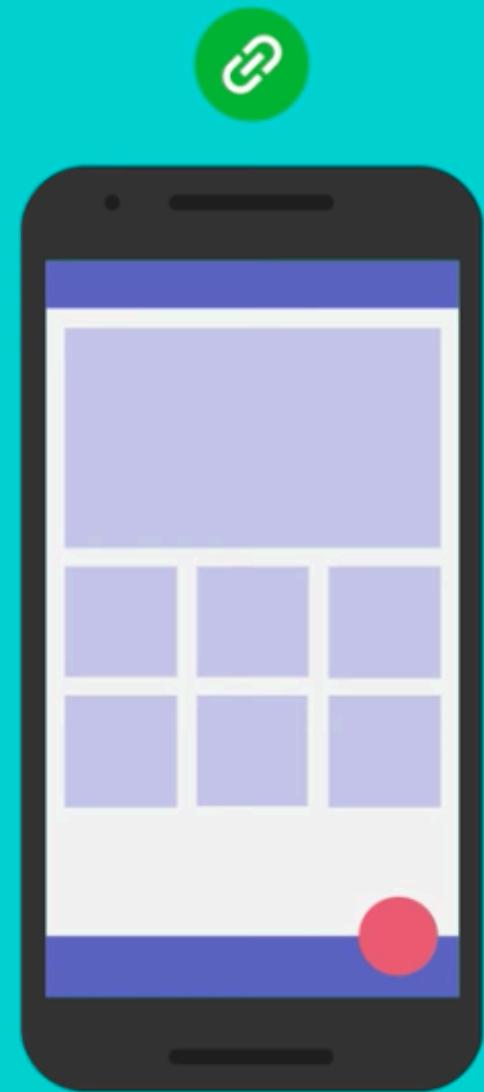
REALTIME DATABASE VS. FIRESTORE

- Firestore
 - Still in beta
 - Better choice for complex data modeling
 - Could be more expensive
- Realtime
 - Limited scalability
 - Tree can be difficult to manage
 - Data structure is simple

FIREBASE DYNAMIC LINKS

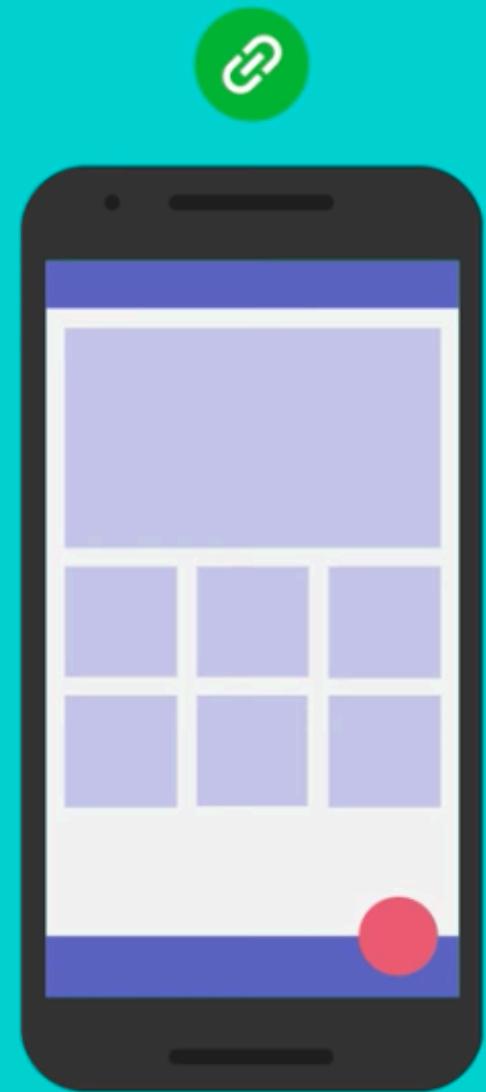
DYNAMIC LINKS

- Firebase dynamic links
 - https://youtu.be/LvY1JMcrPF8?list=PLIK7zZEsvYLmOF_07IayrTntevxtbUxDL



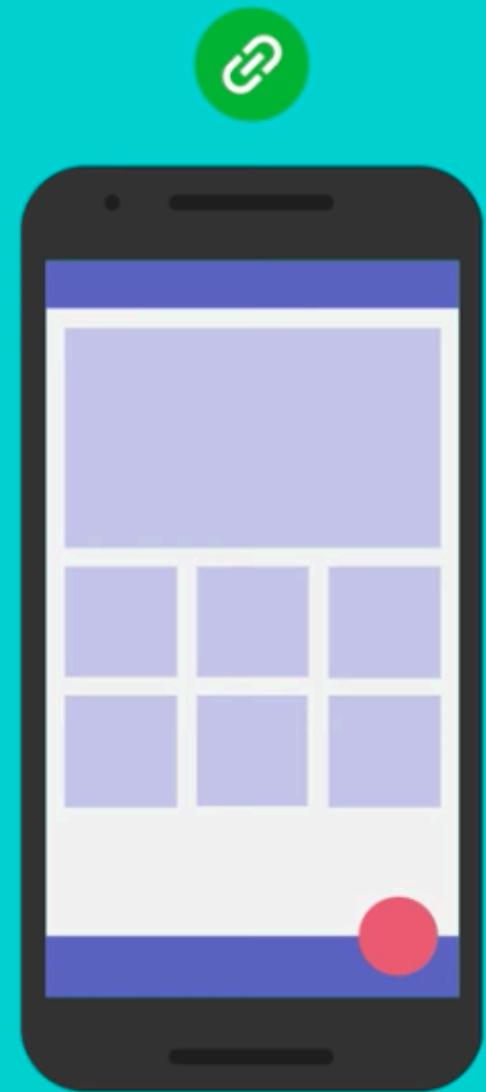
DYNAMIC LINKS

- Dynamic links can be taken directly to the linked content in your native app
- If a user opens the same Dynamic Link in a desktop browser, they can be taken to the equivalent content on your website



DYNAMIC LINKS

- Options for creating links
 - Firebase console
 - REST API
 - iOS Builder API
 - Programmatically



DYNAMIC LINKS

USE CASES

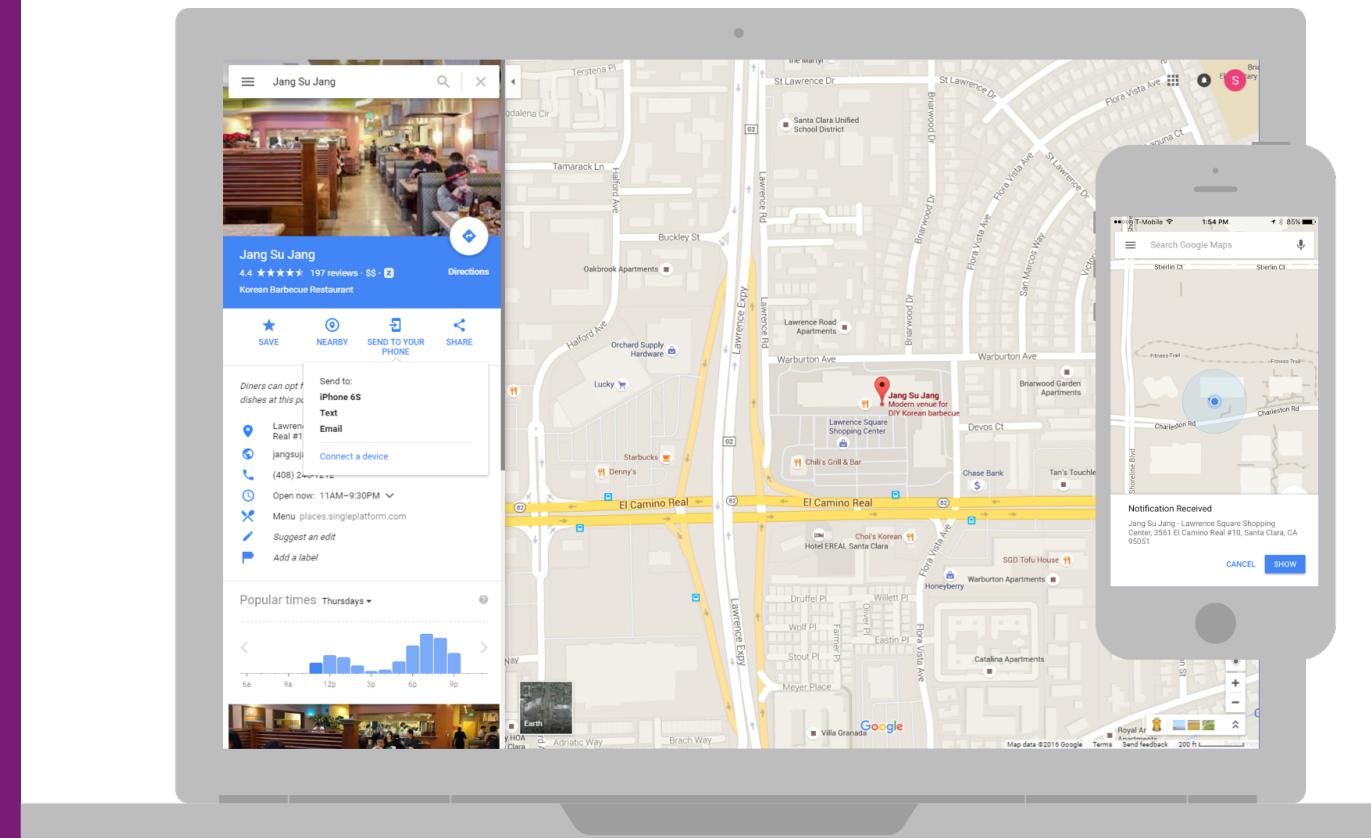
- Converting web (or desktop) users to app users



DYNAMIC LINKS

USE CASES

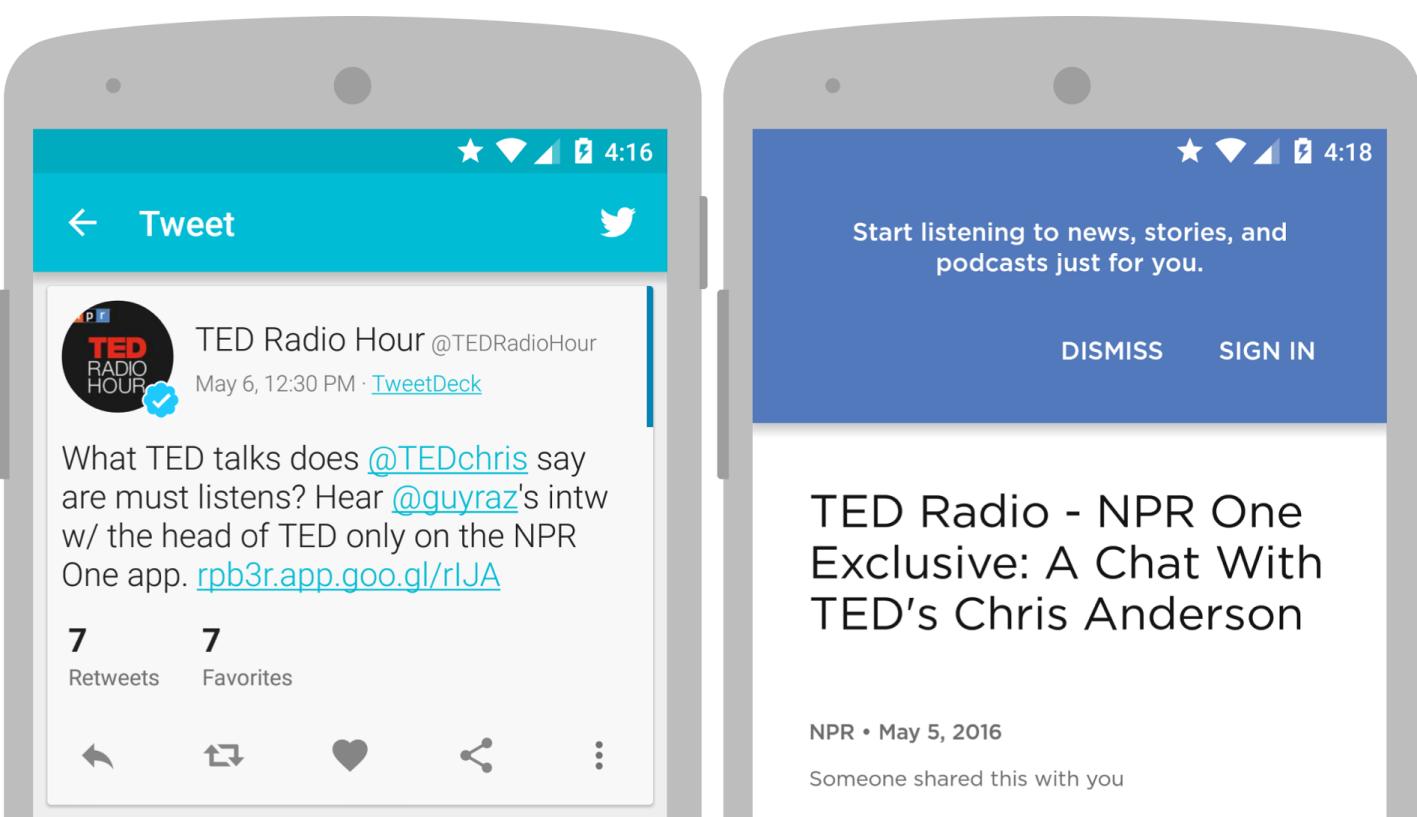
- Don't loose place



DYNAMIC LINKS

USE CASES

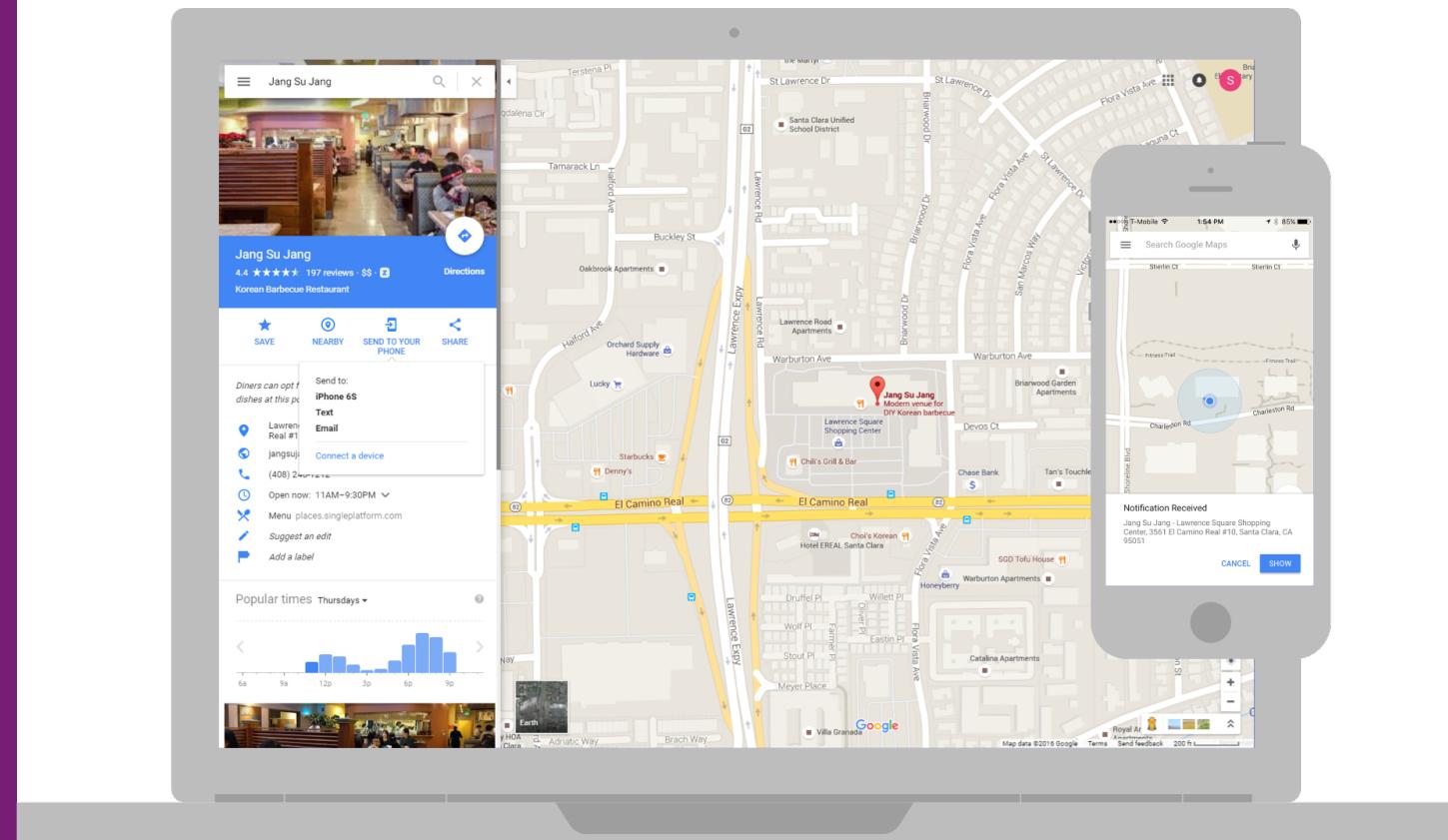
- Send links through social media
 - Direct content links
 - Track engagement



DYNAMIC LINKS

USE CASES

- Send links through social media
 - Direct content links
 - Track engagement



DYNAMIC LINKS

FIREBASE

WHERE WE WERE

- Open this page in our app!

APP

DYNAMIC LINKS

- <https://abc123.app.goo.gl/?link=https://example.com/invitation?gameid%3D1234%26referrer%3D555&apn=com.example.android&ibi=com.example.ios&isi=12345>

FIREBASE

ADDITIONAL DATA

APP

DYNAMIC LINKS

URL SHORTENER

- <https://abc123.app.goo.gl/WXYZ>

DYNAMIC LINKS

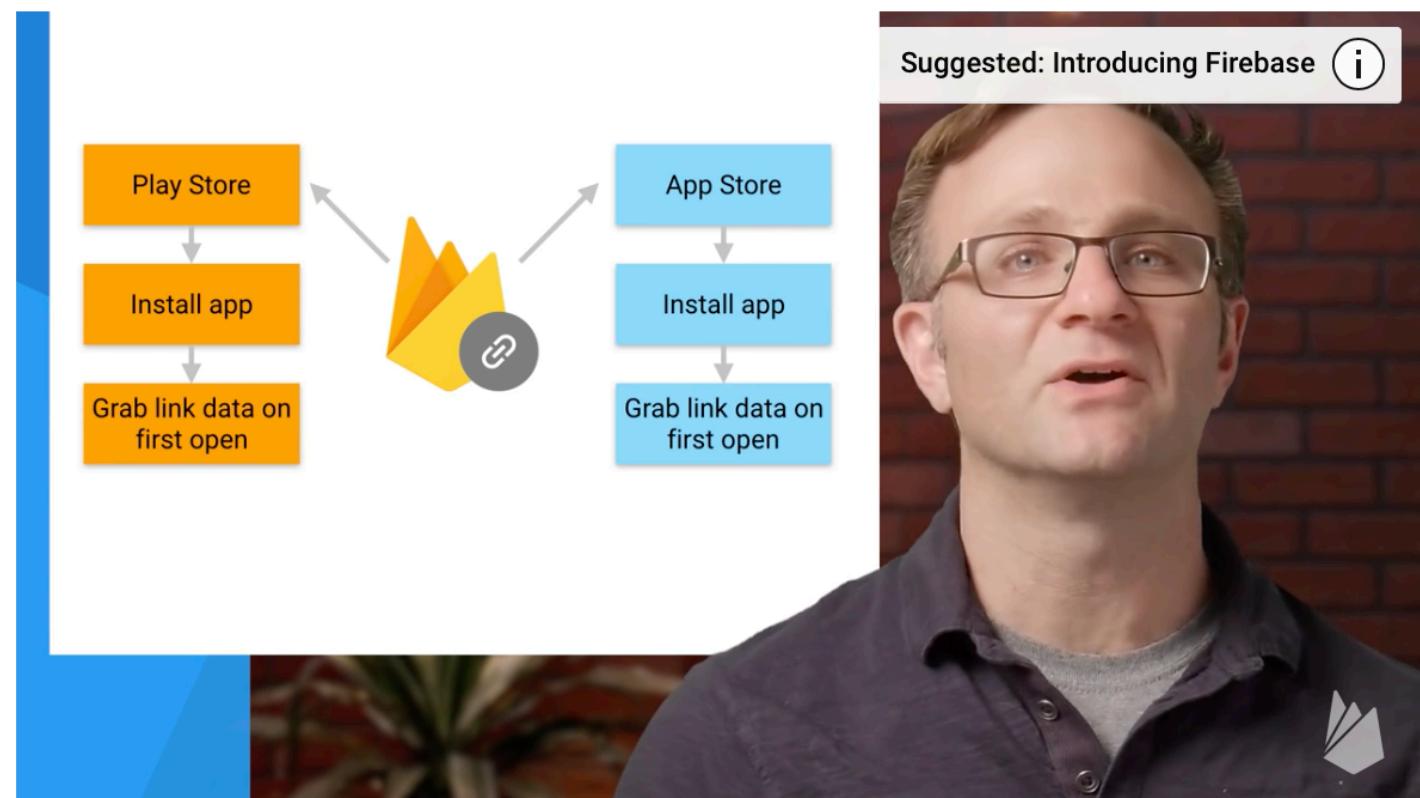
- Implementing deep links

- 1 Set up Firebase and the Dynamic Links SDK
Enable Firebase Dynamic Links for your Firebase project in the Firebase console. Then, include the Dynamic Links SDK in your app.
- 2 Create Dynamic Links
You can create Dynamic Links programmatically or by using the Firebase console.
- 3 Handle Dynamic Links in your app
When your app opens, use the Dynamic Links SDK to check if a Dynamic Link was passed to it. If so, get the link from the Dynamic Link data and handle the link as necessary.
- 4 View analytics data
Track the performance of your Dynamic Links in the Firebase console.

SETTING UP DYNAMICS LINKS

SETTING UP DYNAMICS LINKS

- <https://youtu.be/sFPo296OQqk>
- 15 minutes
Firecast



SETTING UP DYNAMICS LINKS

- Custom URL for your application

`https://www.google.com`

Safari

`mailto:joe@gmail.com`

Mail

`tel:415-555-1212`

Phone

`com.example.myapp://stuffhere`

Your app!

SETTING UP DYNAMICS LINKS

- Set up in the console

Firebase FireChat Go to docs : ?

Dynamic Links

<https://a795j.app.goo.gl/> ⓘ



Dynamic Links are URLs that reduce friction and get users to relevant screens of your app whether or not it is installed

ⓘ [Learn more](#)

[NEW DYNAMIC LINK](#)

SETTING UP DYNAMICS LINKS

- Project settings
- Uniquely identifies the developer

The screenshot shows the Firebase Settings page for an app named "mobi.uchicago.firechat". The left sidebar lists "iOS apps" and the right panel displays various configuration settings:

- Download the latest config file**: A button to download "GoogleService-Info.plist".
- App ID**: 1:487781121319:ios:78425b1380795a79
- App nickname**: An input field with placeholder "Add a nickname" and a pencil icon.
- Bundle ID**: mobi.uchicago.firechat
- App Store ID**: 123456789 (with a pencil icon)
- App ID Prefix**: 723KYE3F9E (highlighted with a yellow background)
- Reporting currency**: An input field with placeholder "USD" and a pencil icon.

At the bottom right are "CANCEL" and "SAVE" buttons.

SETTING UP DYNAMICS LINKS

- Apple developer > Membership
- Team ID

Andrew Binkowski
University of Chicago (Department of Computer Science) ▾



Membership Details

Your team's membership information and legal agreements.

Membership Information

Program Type	iOS Developer University Program
Team Name	University of Chicago (Department of Computer Science)
Team ID	723KYE3F9E
Entity Type	Education / University
Phone	1-773-7026614
Address	1100 East 58th Street Chicago, Illinois 60637 United States

Device Reset Date

Team Agent	Andrew Binkowski
Your Role	Agent

[Need to edit this information?](#)

SETTING UP DYNAMICS LINKS

- Only published apps will have a App Store ID
- Substitute your favorite app
- during development

The screenshot shows the Firebase console interface. On the left, a sidebar lists various services: Overview, Analytics, Authentication, Database, Storage, Hosting, Functions, Test Lab, Crash Reporting, Performance, Notifications, Remote Config, Dynamic Links, and AdMob. The 'Dynamic Links' option is highlighted. At the bottom of the sidebar, there's a 'UPGRADE' button. The main content area is titled 'FireChat – Settings – Firebase' and shows the 'iOS apps' section for the app 'mobi.uchicago.firechat'. A blue callout box highlights the 'App Store ID' field, which contains '284910350'. A tooltip over this field states: 'You can find your App Store ID in your app's URL. In the example below, 123456789 is the App Store ID. https://itunes.apple.com/us/app/yourapp/id123456789'. Other fields visible include 'App ID' (1:487781121319:ios:78425b1380795a79), 'App nickname' (Add a nickname), 'Bundle ID' (mobi.uchicago.firechat), 'App ID Prefix' (723KYE3F9E), 'Reporting currency' ((USD \$) US Dollar), and 'Time zone' ((GMT-05:00) GMT-05:00). A 'DELETE THIS APP' button is located at the bottom right.

SETTING UP DYNAMICS LINKS

- Only published apps will have a App Store ID
- Substitute your favorite app during development

Secure | https://console.firebaseio.google.com/u/0/project/firechat-66f87/settings/general/ios:mobi.uchicago.firechat

Firebase FireChat Settings Overview ADD APP

Analytics

ELOP

Authentication

Database

Storage

Hosting

Functions

Test Lab

Crash Reporting

Performance Monitoring

W

Notifications

Remote Config

Dynamic Links

AdMob

Workshop \$0/month UPGRADE

iOS apps

iOS mobi.uchicago.firechat

Download the latest config file

GoogleService-Info.plist

App ID ② 1:487781121319:ios:78425b1380795a79

App nickname Add a nickname

Bundle ID mobi.uchicago.firechat

App Store ID ② You can find your App Store ID in your app's URL. In the example below, 123456789 is the App Store ID.
284910350 https://itunes.apple.com/us/app/yourapp/id123456789

App ID Prefix ② 723KYE3F9E

Reporting currency ② (USD \$) US Dollar

Time zone ② (GMT-05:00) GMT-05:00

DELETE THIS APP

IF THE USER DOESN'T HAVE THE APP INSTALLED, ONE REDIRECT OPTION IS TO GO TO APP STORE

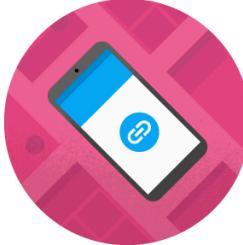
SETTING UP DYNAMICS LINKS

- Dynamic links domain
- Hosts redirects based on client

Firebase FireChat Go to docs : ?

Dynamic Links

<https://a795j.app.goo.gl/> ?

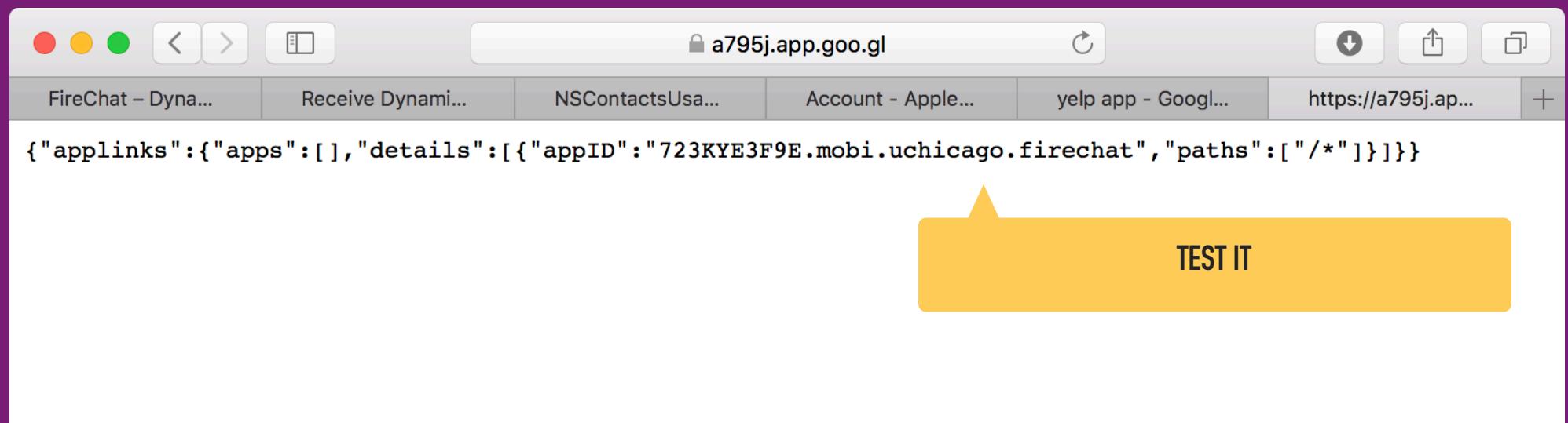


Dynamic Links are URLs that reduce friction and get users to relevant screens of your app whether or not it is installed

Learn more

NEW DYNAMIC LINK

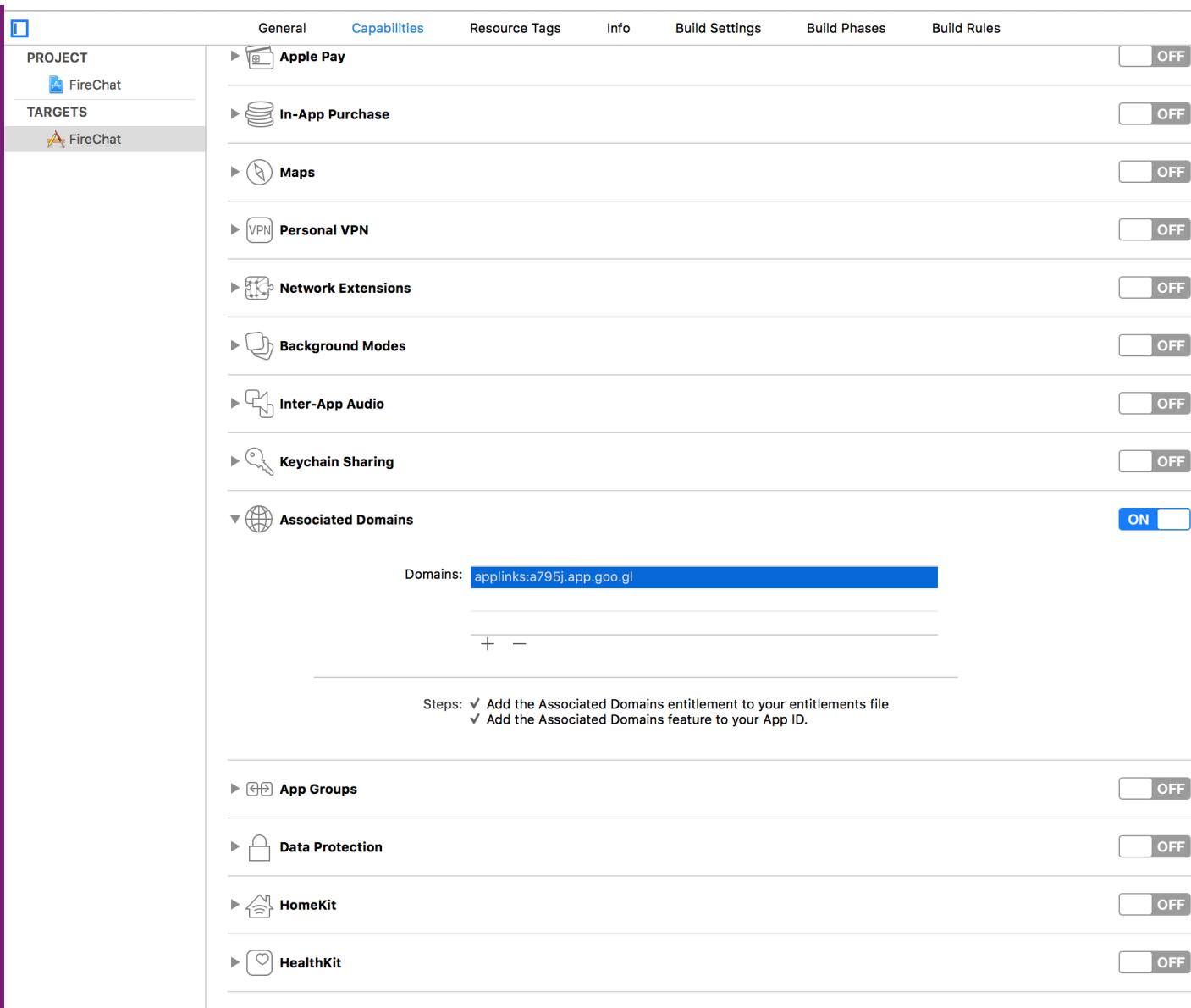
SETTING UP DYNAMICS LINKS



- <https://a795j.app.goo.gl/apple-app-site-association>
- This URL will redirect to my application

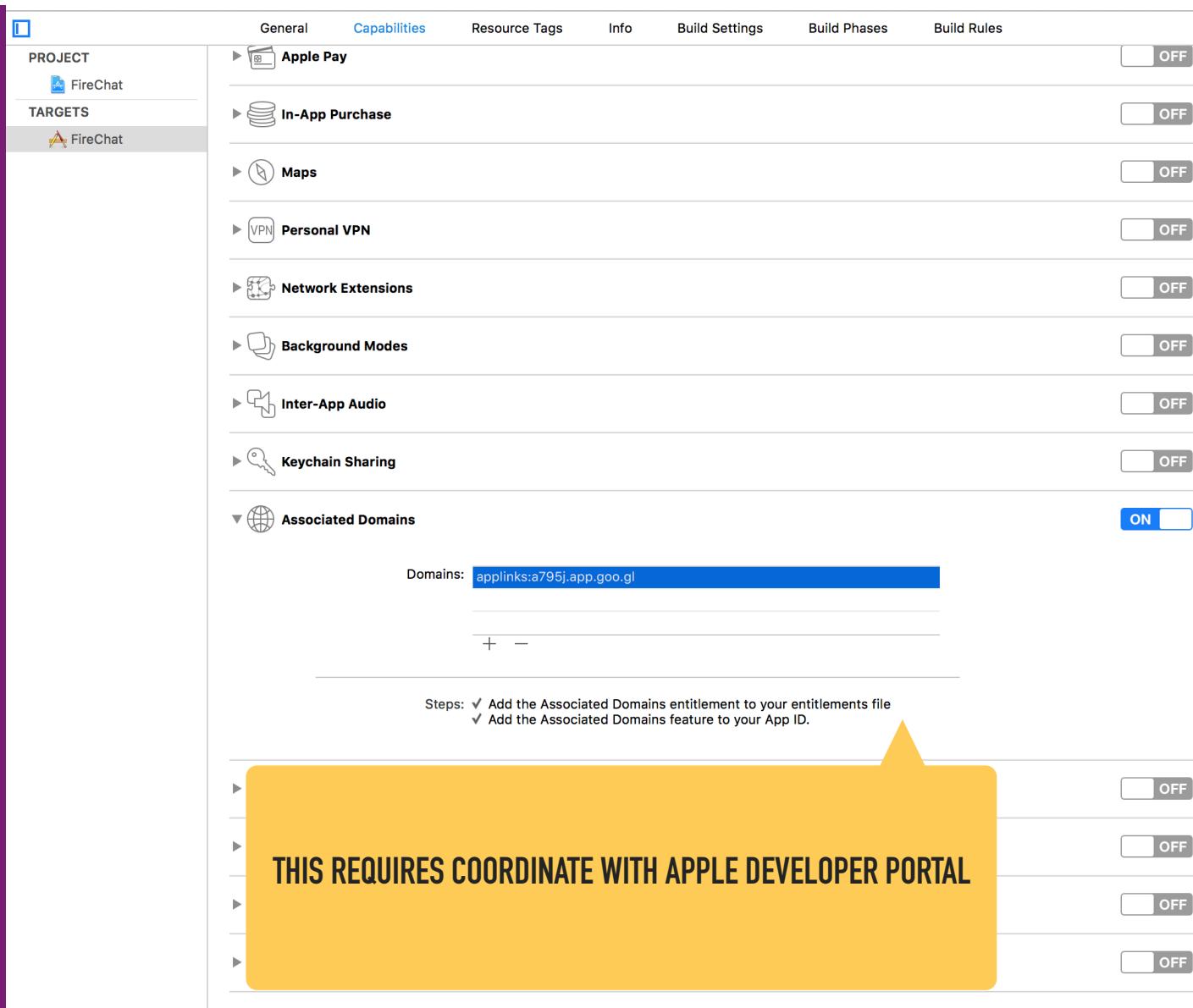
SETTING UP DYNAMICS LINKS

- Set up iOS app to handle Universal Links
- Firebase is hosting our associated domain



SETTING UP DYNAMICS LINKS

- Set up iOS app to handle Universal Links
- Firebase is hosting our associated domain



SETTING UP DYNAMICS LINKS

- Enable Associated Domains for App
- Xcode will (should) do this automatically for you

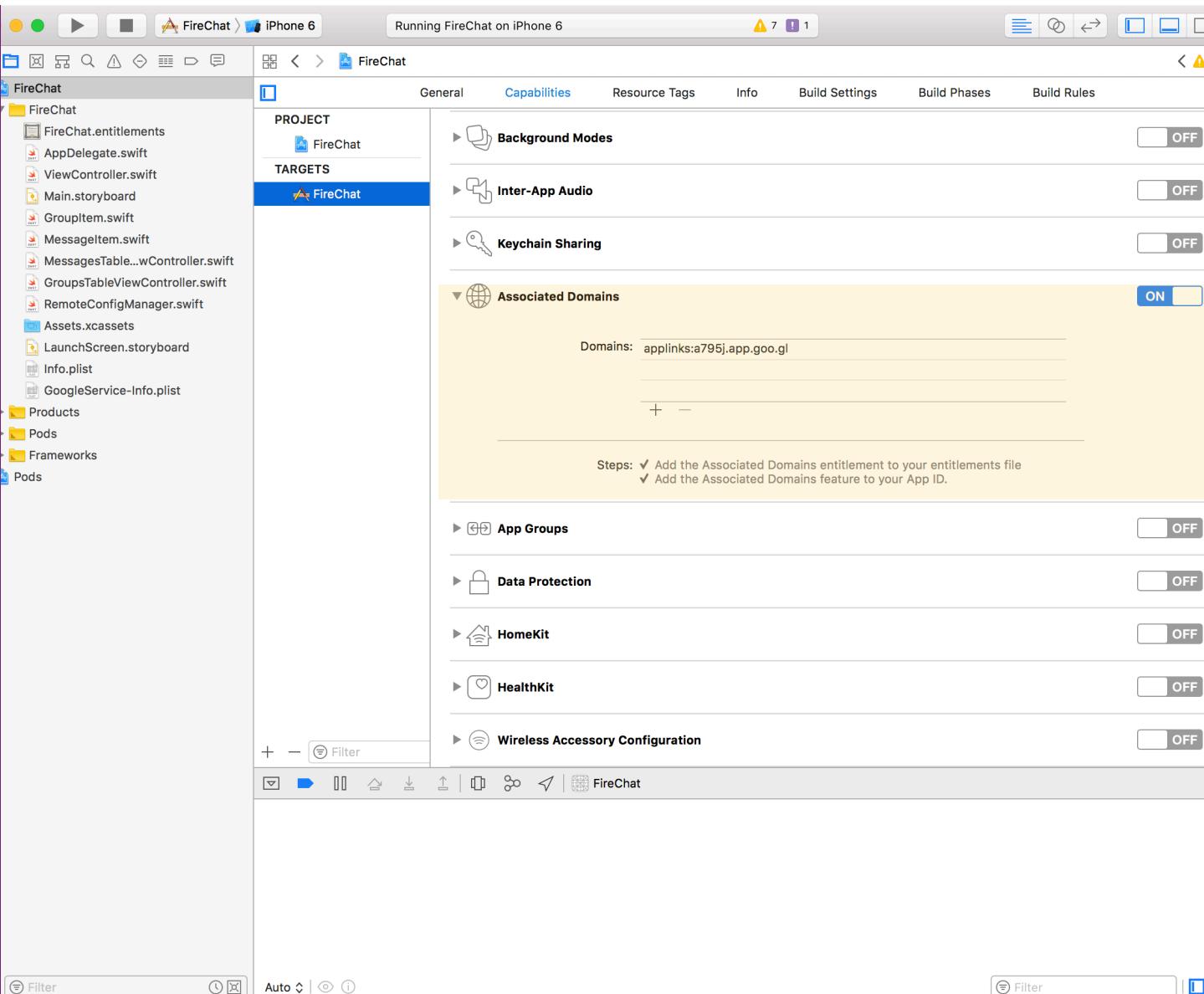
The screenshot shows the Xcode Identity & Capabilities editor. On the left, there's a sidebar with categories: Keys, Identifiers, Devices, and Provisioning Profiles. Under Identifiers, 'App IDs' is selected, showing two entries: 'XC mobi uchicago -016-ShareExtension Fa...' and 'XC mobi uchicago -016-ShareExtension To...'. Under Devices, 'All' is selected, listing Apple TV, Apple Watch, iPad, iPhone, and iPod Touch. Under Provisioning Profiles, 'All' is selected, showing 'Development'.

The main pane displays the 'XC mobi uchicago FireChat' configuration. It includes fields for Name (XC mobi uchicago FireChat), Prefix (723KYE3F9E), and ID (mobi.uchicago.FireChat). Below these are sections for Application Services, showing various options like App Groups, Associated Domains, Data Protection, Game Center, HealthKit, HomeKit, iCloud, In-App Purchase, Inter-App Audio, Personal VPN, Push Notifications, Wallet, and Wireless Accessory Configuration. The 'Associated Domains' row is highlighted with a yellow background, indicating it's the current selection. At the bottom, there's an 'Edit' button and another entry for 'XC mobi uchicago instaWatchItOnTv'.

Service	Development	Distribution
App Groups	<input type="radio"/> Disabled	<input type="radio"/> Disabled
Associated Domains	<input checked="" type="radio"/> Enabled	<input checked="" type="radio"/> Enabled
Data Protection	<input type="radio"/> Disabled	<input type="radio"/> Disabled
Game Center	<input checked="" type="radio"/> Enabled	<input checked="" type="radio"/> Enabled
HealthKit	<input type="radio"/> Disabled	<input type="radio"/> Disabled
HomeKit	<input type="radio"/> Disabled	<input type="radio"/> Disabled
iCloud	<input type="radio"/> Disabled	<input type="radio"/> Disabled
In-App Purchase	<input checked="" type="radio"/> Enabled	<input checked="" type="radio"/> Enabled
Inter-App Audio	<input type="radio"/> Disabled	<input type="radio"/> Disabled
Personal VPN	<input type="radio"/> Disabled	<input type="radio"/> Disabled
Push Notifications	<input type="radio"/> Disabled	<input type="radio"/> Disabled
Wallet	<input type="radio"/> Disabled	<input type="radio"/> Disabled
Wireless Accessory Configuration	<input type="radio"/> Disabled	<input type="radio"/> Disabled

SETTING UP DYNAMICS LINKS

- Enable Associated Domains for App
- Xcode will (should) do this automatically for you



SETTING UP DYNAMICS LINKS

Domains

Domains: `applinks:a795j.app.goo.gl`

+

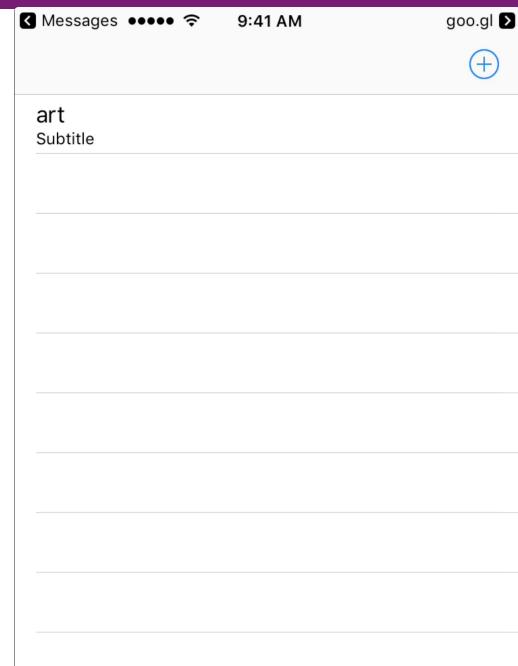
Steps: ✓ Add the Associated Domains entitlement to your entitlements file
✓ Add the Associated Domains feature to your App ID.

- <https://a795j.app.goo.gl>

SETTING UP DYNAMICS LINKS



TEST ON DEVICE



DYNAMIC LINKS NEED TO BE CLICKED ON, THEY DON'T WORK BY TYPING ADDRESS IN TO BROWSER. WORKAROUNDS: SEND AS TEXT MESSAGE, PUT IN NOTE, ETC.

CREATING A DYNAMIC LINK

CREATING A DYNAMIC LINK

- Set up iOS app to handle Universal Links
- Firebase is hosting our associated domain

https://a795j.app.goo.gl/ ⓘ



Dynamic Links are URLs that reduce friction and get users to relevant screens of your app whether or not it is installed

ⓘ [Learn more](#)

[NEW DYNAMIC LINK](#)

CREATING A DYNAMIC LINK

FIREBASE DYNAMIC LINKS

IOS UNIVERSAL LINKS

ANDROID LINKS

CREATING A DYNAMIC LINK

- Set up Dynamic link in the console

Dynamic Links

← Create Dynamic Link ?

1 Set up your Dynamic Link

A Dynamic Link is a deep link into your app that works whether or not your app is installed. On desktop it will go to the deep link url. [Learn More](#)

Deep link URL ?

Example: `https://yourapp.com/welcome`

Dynamic Link name ?

Arts Chat

NEXT

2 Define link behavior for iOS

3 Define link behavior for Android

4 Track a campaign with UTM parameters (optional)

5 Add social meta tags for better sharing (optional)

CREATING A DYNAMIC LINK

```
https://abcde.app.goo.gl/?  
link=http://moviereviews.example.com/reviewID%eD42663  
&isi=48151632&ibi=com.example.moviereviews
```



- All data is passed in through URL parameters
- Allows the links to be "universal"

CREATING A DYNAMIC LINK

- URL contains information that will be read in the iOS app

The screenshot shows the Firebase console interface. On the left, a sidebar lists various services: Overview, Analytics, Authentication, Database, Storage, Hosting, Functions, Test Lab, Crash Reporting, Notifications, Remote Config, Dynamic Links (which is highlighted in blue), and AdMob. The main area is titled "Dynamic Links" and "Create Dynamic Link". It displays a step-by-step guide:

- Set up your Dynamic Link**

A Dynamic Link is a deep link into your app that works whether or not your app is installed. On desktop it will go to the deep link url. [Learn More](#)

Deep link URL:

Dynamic Link name:

NEXT
- Define link behavior for iOS
- Define link behavior for Android
- Track a campaign with UTM parameters (optional)
- Add social meta tags for better sharing (optional)

You can also create Dynamic Links from your app programmatically. [Learn more](#)

CANCEL CREATE DYNAMIC LINK

CREATING A DYNAMIC LINK

- Define the behavior of the link in the selected application



Set up your Dynamic Link

Art Group, Deep link URL: <http://uchicago.mobi/groups/art>



Define link behavior for iOS

- Open the deep link URL in a browser
 Open the deep link in your iOS App

iOS mobi.uchicago.firechat



If your app is not installed, send the user to

- App Store page for your app
 Deep link URL ?
 Custom URL ?

Advanced Settings (optional)

- Open a different app if on iPad
 Add App Store campaign or affiliate parameters
 Use a custom scheme when universal links aren't supported ?

PREVIOUS

NEXT



Define link behavior for Android

Open the deep link URL in a browser

CREATING A DYNAMIC LINK

- Analytics tags
- Mirrors Google Analytics campaign tracking

1 Set up your Dynamic Link
Art Group, Deep link URL: <http://uchicago.mobi/groups/art>

2 Define link behavior for iOS
Link directly in `mobi.uchicago.firebaseio.com`

3 Define link behavior for Android
Open the deep link URL in a browser

4 Track a campaign with UTM parameters (optional)

Track the traffic sources and campaigns that send users to your app

Campaign source (`utm_source`) [?](#) Campaign medium (`utm_medium`) [?](#)

Example: Google Example: cpc

Campaign name (`utm_campaign`) [?](#)

Example: Spring sale

[PREVIOUS](#) [NEXT](#)

5 Add social meta tags for better sharing (optional)

You can also create Dynamic Links from your app programmatically. [Learn more](#)

[CANCEL](#) [CREATE DYNAMIC LINK](#)

CREATING A DYNAMIC LINK

- Social media tags and data



Set up your Dynamic Link

Art Group, Deep link URL: <http://uchicago.mobi/groups/art>



Define link behavior for iOS

Link directly in `mobi.uchicago.firechat`



Define link behavior for Android

Open the deep link URL in a browser

4

Track a campaign with UTM parameters (optional)

5

Add social meta tags for better sharing (optional)

These tags create a preview of your link when shared on social media. These values will override existing social meta tags for your destination link.

Preview title (st) [?](#)

Example: Holiday Promotion

Preview image URL (si) [?](#)

Example: <https://yoururl.com/ima>

Preview description (sd) [?](#)

Example: Get 25% off when you spend \$50 or more!

CREATING A DYNAMIC LINK

Dynamic Links ?

<https://a795j.app.goo.gl/> ? NEW DYNAMIC LINK

Link name	Created ↑	URL	⌚ Clicks (30 days)
Art Group	Apr ...	https://a795j.app.goo.gl/...	0

- Your link 

CREATING A DYNAMIC LINK

Link name

Art Group

Deep link

<http://uchicago.mobi/groups/art>

FALL BACK

iOS app

 <mobi.uchicago.firechat>

FULL LINK

Long Dynamic Link

<https://a795j.app.goo.gl/?link=http://uchicago.mobi/groups/art&isi=284910350&ibi=mobi.uchicago.firechat>

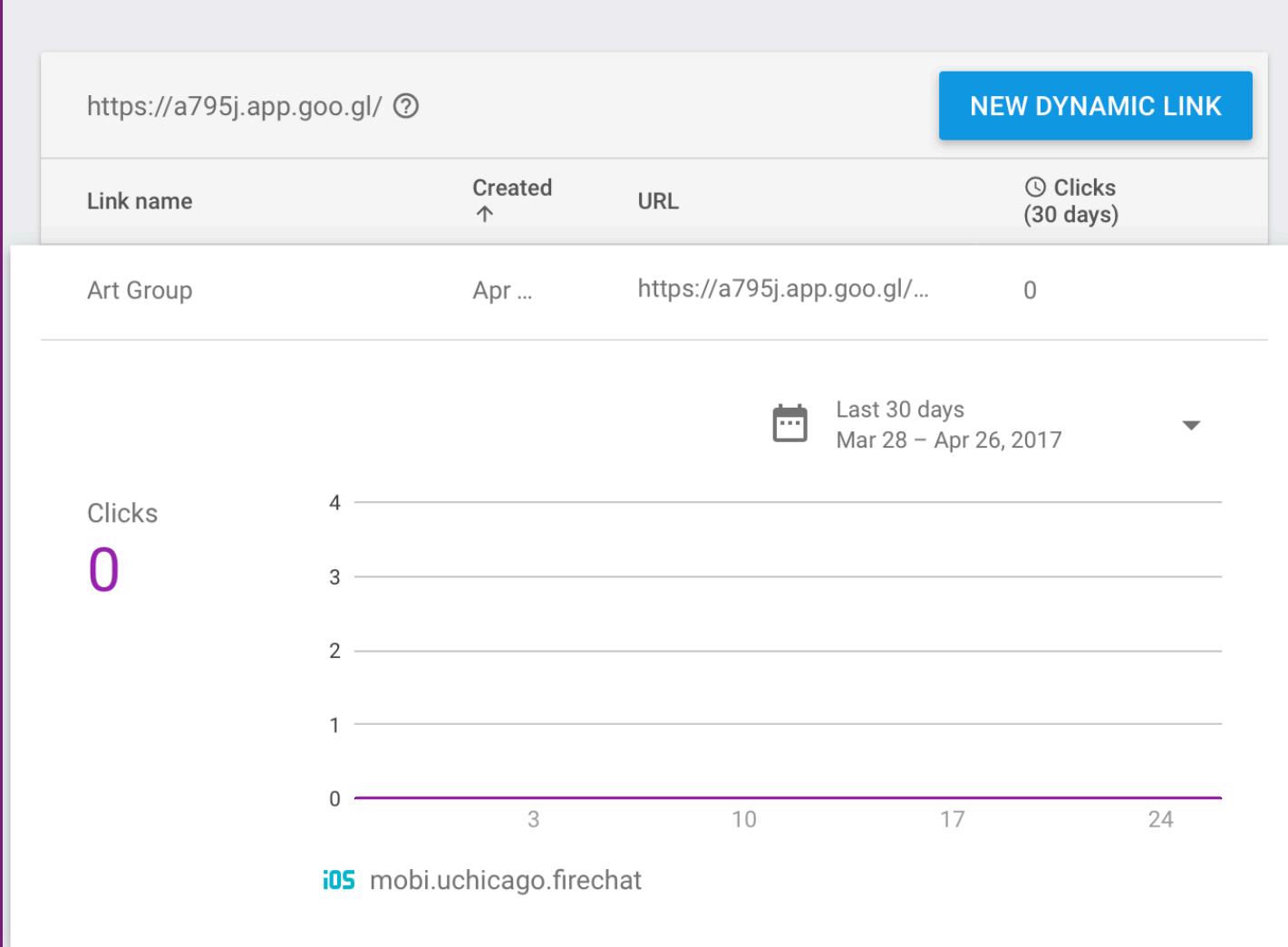
Short Dynamic Link

<https://a795j.app.goo.gl/6SuK>

SHORTENED LINK

CREATING A DYNAMIC LINK

- Analytics for specific link



HANDLING DYNAMIC LINKS

HANDLING DYNAMIC LINKS

`https://abcde.app.goo.gl/tyPW`



Dynamic Links library



FIRDynamicLink
object

HANDLING DYNAMIC LINKS

```
# Uncomment the next line to define a global platform for your project
platform :ios, '9.0'

target 'FireChat' do
  # Comment the next line if you're not using Swift and don't want to use dynamic frameworks
  use_frameworks!

  # Pods for FireChat
  pod 'Firebase/Core'
  pod 'Firebase/Database'

  pod 'Firebase/Auth'
  pod 'GoogleSignIn'

  pod 'Firebase/Crash'

  pod 'Firebase/Storage'
  pod 'Firebase/Invites'

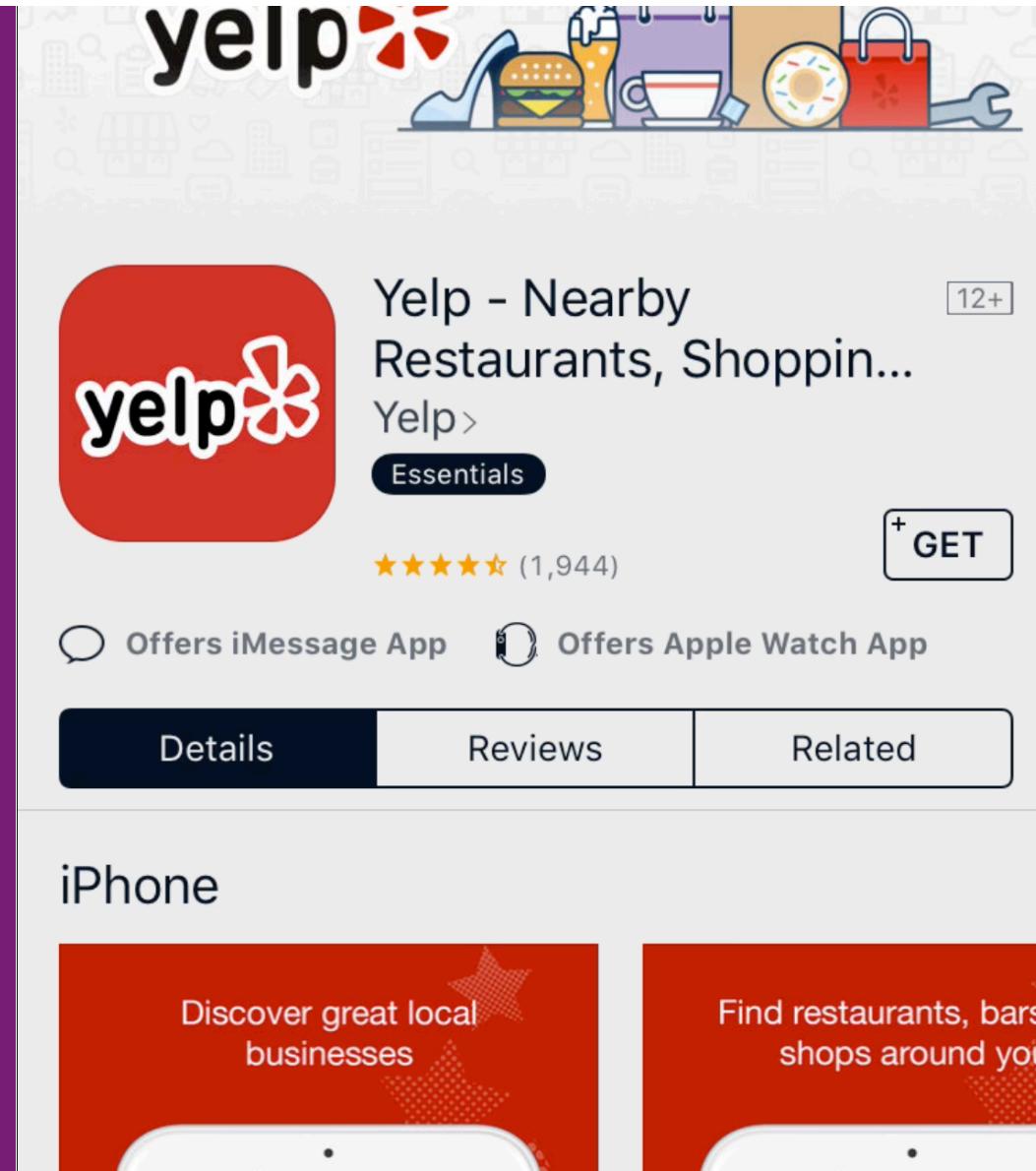
  pod 'Firebase/DynamicLinks'

  pod 'SDWebImage'

end
```

HANDLING DYNAMIC LINKS

- Two different flows
 - Custom URL Scheme
 - First install
 - Will direct to the app store id that you specified
- Deep link
- iOS functions
- Will launch the app and pass the link



HANDLING DYNAMIC LINKS

```
// Handle dynamic links
@available(iOS 8.0, *)
func application(_ application: UIApplication,
                 continue userActivity: NSUserActivity,
                 restorationHandler: @escaping ([Any]?) -> Void) -> Bool {
    guard let dynamicLinks = FIRDynamicLinks.dynamicLinks() else {
        return false
    }
    let handled = dynamicLinks.handleUniversalLink(userActivity.webpageURL!) { (dynamiclink, error) in
        if let dynamiclink = dynamiclink, let _ = dynamiclink.url {
            self.handleIncomingLink(dynamiclink)
        } else {
            print("Error: \(String(describing: error?.localizedDescription))")
        }
    }
    return handled
}
```

- In App Delegate detect open from dynamic links

HANDLING DYNAMIC LINKS

```
// Open the app from URL callback (ie. mobi.uchicago.firechat) during
// Google authentication OR from an intial install on a deep link
@available(iOS 9.0, *)
func application(_ application: UIApplication,
                  open url: URL,
                  options: [UIApplicationOpenURLOptionsKey : Any]) -> Bool {

    // Try to handle the dynamic link first
    let dynamicLink = FIRDynamicLinks.dynamicLinks()?.dynamicLink(fromCustomSchemeURL: url)
    if dynamicLink != nil {
        // Handle the deep link. For example, show the deep-linked content or
        // apply a promotional offer to the user's account.
        self.handleIncomingLink(dynamicLink!)
        print("深深的 Deep link of first launch or old device")
        return true
    }

    // Handle the Google ID Authentication
    return GIDSignIn.sharedInstance().handle(url,
                                             sourceApplication:options[UIApplicationOpenURLOptionsKey.sourceApplication] as? String,
                                             annotation: [:])
}
```

- In App Delegate, handle a first launch from deep link

HANDLING DYNAMIC LINKS

```
/// Take a dynamic link and parse it into its components
func handleIncomingLink(_ dynamicLink: FIRDynamicLink) {
    print("🔗 DynamicLink: \(dynamicLink)")
    guard let pathComponents = dynamicLink.url?.pathComponents else { return }
    for component in pathComponents {
        print("\t>> component: \(component)")
    }
}
```

```
DynamicLink: <FIRDynamicLink: 0x17024a4a0, url [http://uchicago.mobi/
groups/art], confidence: weak>
>> component: /
>> component: groups
>> component: art
```

DO SOMETHING USEFUL HERE
ADD USER TO GROUP

HANDLING DYNAMIC LINKS

- Link confidence
- Depends on the content

```
matchConfidence = weak  
"20% off coupon"
```

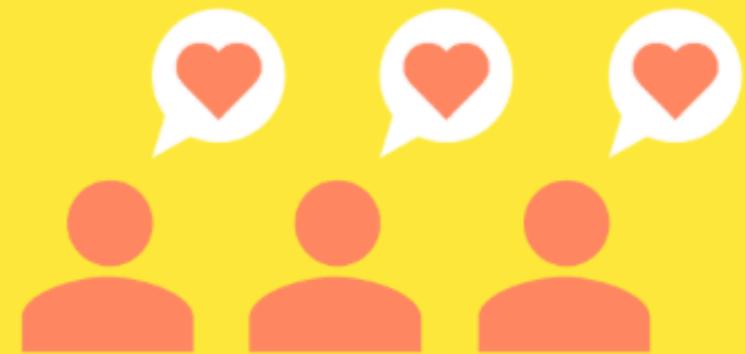
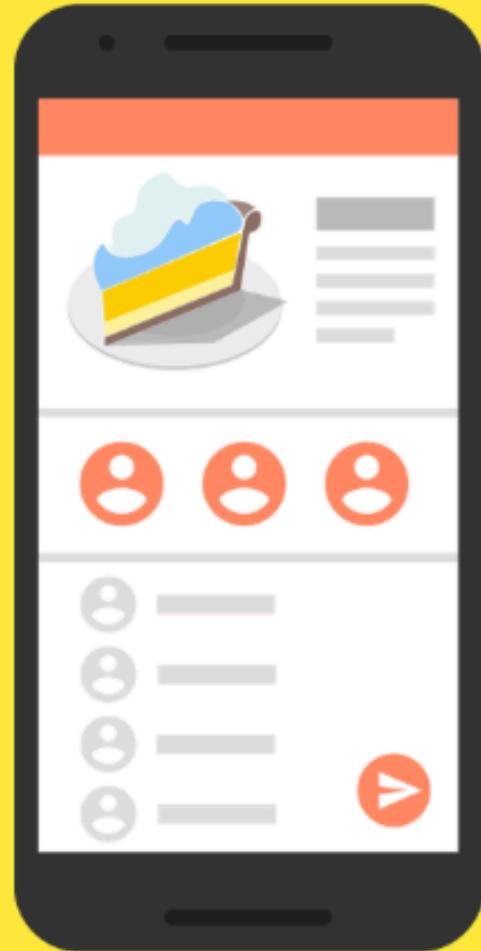
Probably okay

```
matchConfidence = weak  
"Shared by Joan Smith!"
```

FIREBASE INVITES

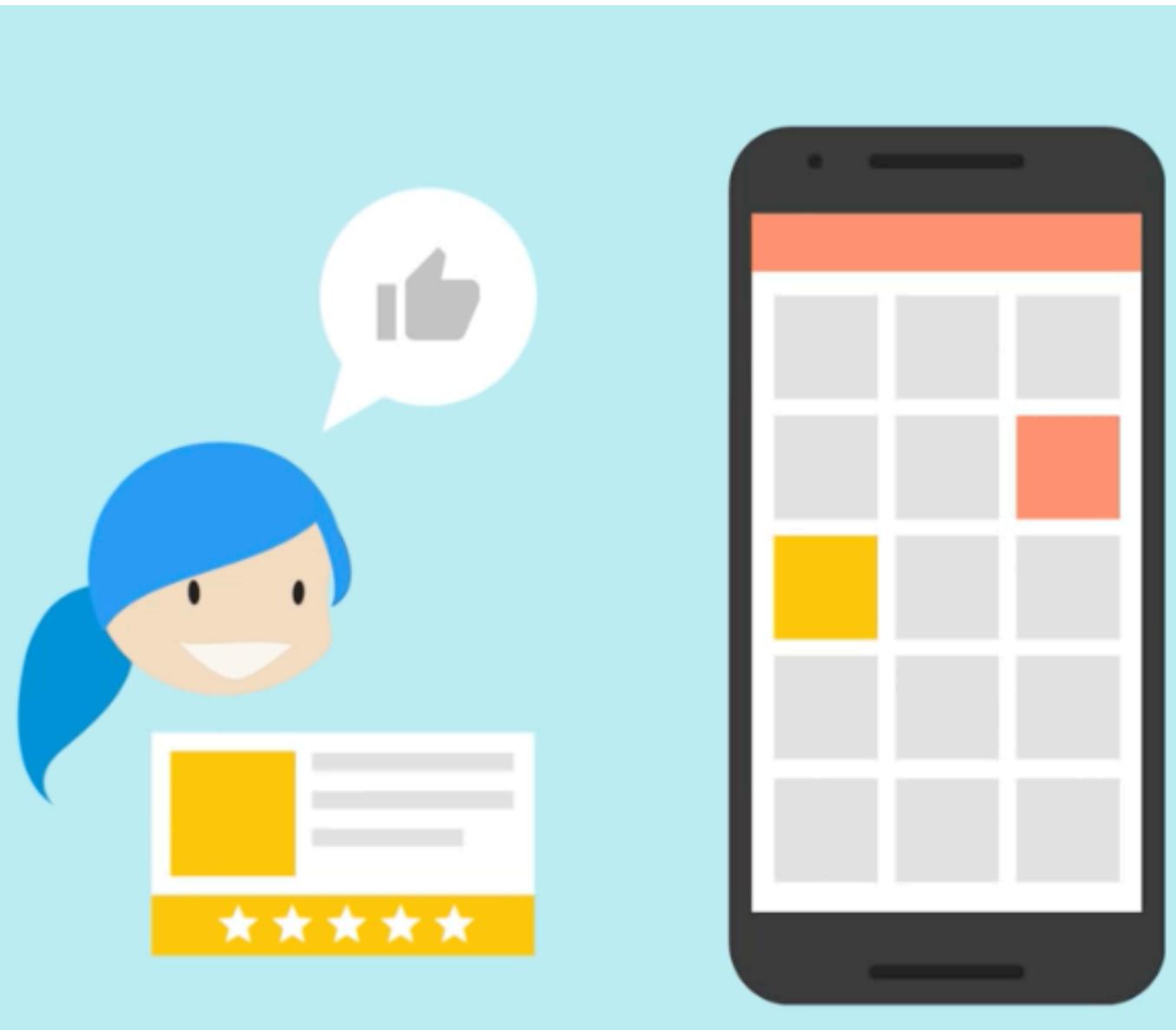
INVITES

- [https://youtu.be/_LkalJCZ_HyM?
list=PLI-K7zZEesYLmOF_07IayrTntevxtbUxDL](https://youtu.be/_LkalJCZ_HyM?list=PLI-K7zZEesYLmOF_07IayrTntevxtbUxDL)



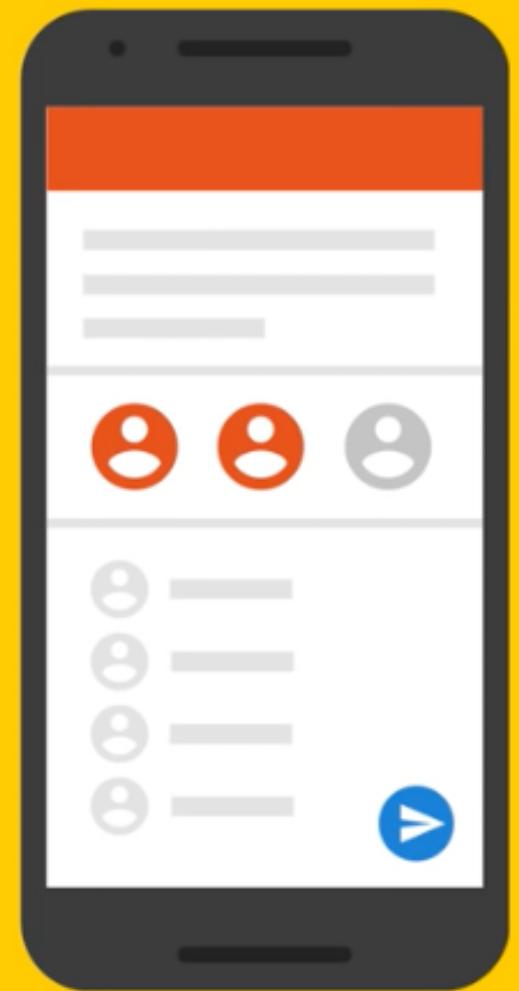
INVITES

- Firebase invites is mechanism for delivering dynamic links
- Built-in UI (material design style)
- Delivered from your authenticated account



INVITES

- Simpler than cut/pasting links in app
- No need to build additional controllers and UI to support a common task
- Tracking/analytics that are more consistent



INVITES

1

Handle links in your app

Enable linking directly to the content you want to share. You handle links on iOS by using custom URL schemes or Universal Links and on Android by using intent filters.

2

Add Share button to your app

When users click a Share button, use the Firebase Invites SDK to set up and open the sharing screen.

3

Handle Dynamic Links in your app

To enable your app to receive invitations, when your app opens, use the Dynamic Links SDK to check if a Dynamic Link was passed to it. If so, get the link from the Dynamic Link data and handle the link as necessary.

INVITES

```
# Uncomment the next line to define a global platform for your project
platform :ios, '9.0'

target 'FireChat' do
  # Comment the next line if you're not using Swift and don't want to use dynamic frameworks
  use_frameworks!

  # Pods for FireChat
  pod 'Firebase/Core'
  pod 'Firebase/Database'

  pod 'Firebase/Auth'
  pod 'GoogleSignIn'

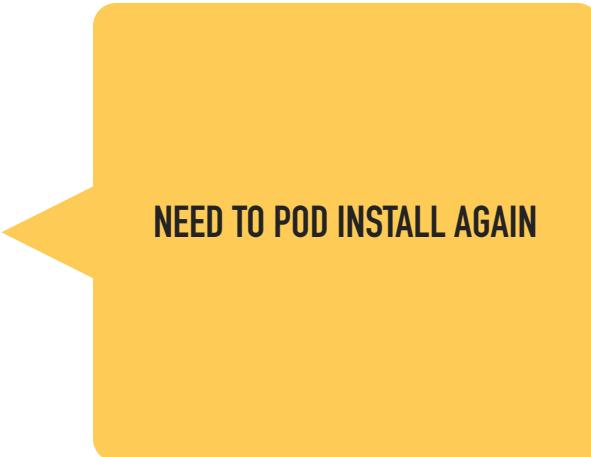
  pod 'Firebase/Crash'

  pod 'Firebase/Storage'
  pod 'Firebase/Invites'

  pod 'Firebase/DynamicLinks'

  pod 'SDWebImage'

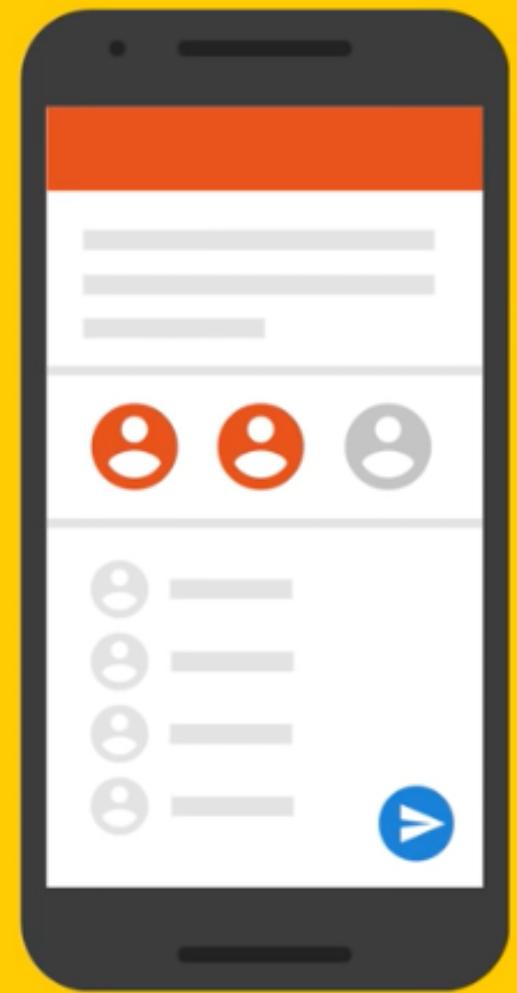
end
```



NEED TO POD INSTALL AGAIN

INVITES

- Users must be authenticated using "Google Sign-in" to use Invites



INVITES

```
// For iOS8 and older
func application(_ application: UIApplication,
                  open url: URL, sourceApplication: String?, annotation: Any) -> Bool {
    if let invite = FIRInvites.handle(url, sourceApplication:sourceApplication, annotation:annotation) as? FIRReceivedInvite {
        let matchType =
            (invite.matchType == .weak) ? "Weak" : "Strong"
        print("Invite received from: \(sourceApplication ?? "") Deeplink: \(invite.deepLink)," +
            "Id: \(invite.inviteId), Type: \(matchType)")
        return true
    }
    return GIDSignIn.sharedInstance().handle(url, sourceApplication: sourceApplication, annotation: annotation)
}
```

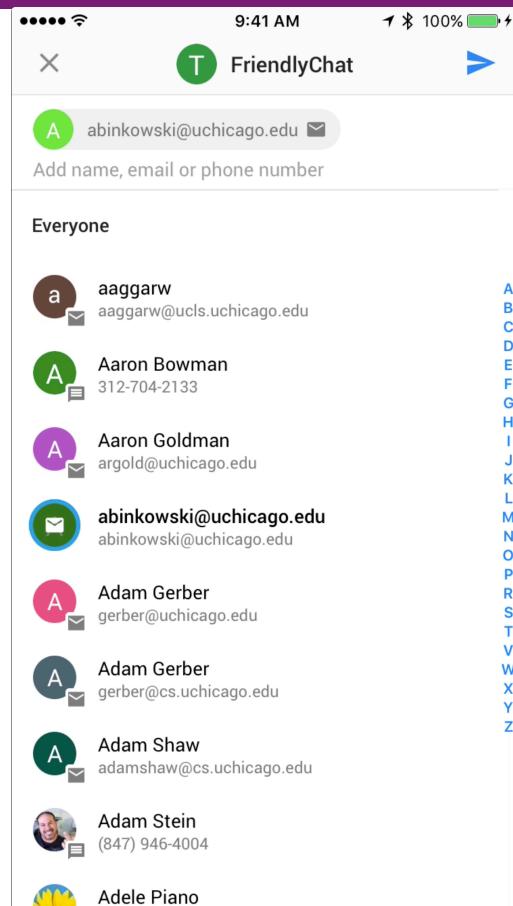
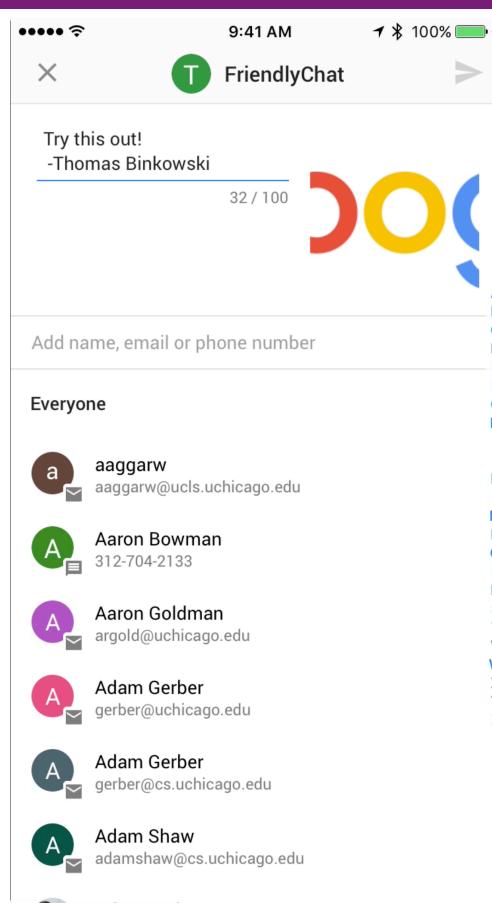
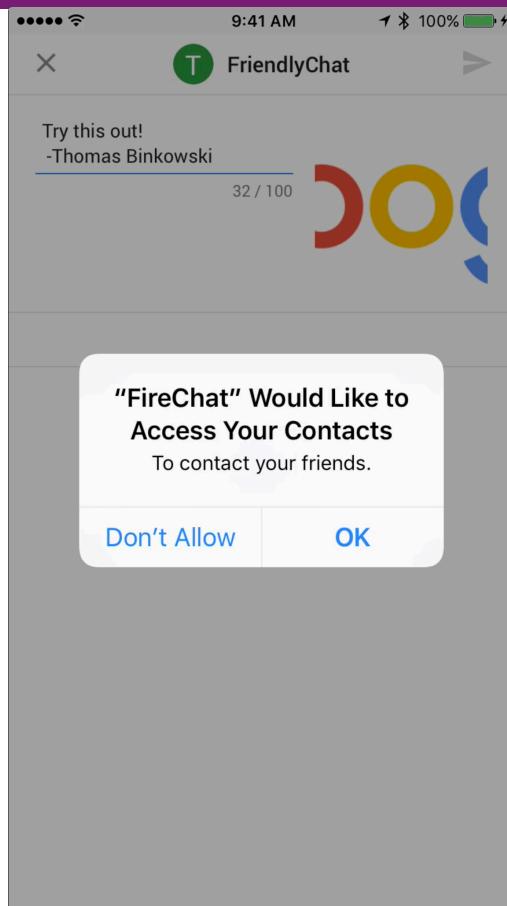
- Handle open from invite similar to dynamic links
- Parse to action

FIREBASE INVITES

SEND AN INVITE

```
if let invite = FIRInvites.inviteDialog() {  
    invite.setInviteDelegate(self as FIRInviteDelegate)  
  
    // NOTE: You must have the App Store ID set in your developer console project  
    // in order for invitations to successfully be sent.  
  
    // A message hint for the dialog. Note this manifests differently depending on the  
    // received invitation type. For example, in an email invite this appears as the subject.  
    invite.setMessage("Try this out!\n -\\" + (FIRAuth.auth()?.currentUser?.displayName ?? "") +")  
    // Title for the dialog, this is what the user sees before sending the invites.  
    invite.setTitle("FriendlyChat")  
    invite.setDeepLink("app_url")  
    invite.setCallToActionText("Install!")  
    invite.setCustomImage(https://www.google.com/images/branding/googlelogo/2x/googlelogo\_color\_272x92dp.png)  
    invite.open()  
}
```

FIREBASE INVITES



FIREBASE INVITES

- It uses your App Store id (hence yelp)
- The install link is customized by you (dynamic link) on mobile device only



The image shows a Google Play store page for the Yelp app. At the top, there's a red square icon with the word "yelp" and a small flower-like logo. Next to it, the text "Yelp - Nearby Restaurants,..." and "Yelp, Inc.". Below this, the large Google logo is displayed. To the right of the logo, the text "App Ratings:" followed by a 5-star rating icon and "(238,305) App Store". A blue "Install!" button is located below the rating. At the bottom of the page, there's a paragraph of text: "Yelp has over 100 million reviews of businesses worldwide and is available for iPhone, iPad, and Apple Watch. Whether you're looking for a pizzeria that just opened or a coffee shop nearby, Yelp is your local guide to finding the perfect place to eat, shop, drink, relax and play...".

FIREBASE INVITES

```
if let invite = FIRInvites.inviteDialog() {  
    invite.setInviteDelegate(self as FIRInviteDelegate)  
  
    // NOTE: You must have the URL scheme defined in your developer console project  
    // in order for invitation links to work.  
    invite.setDeepLink("MY DEEP LINK")  
  
    // A message hint for the dialog. Note this may route differently depending on the  
    // received invitation type. For example, in an email invite this appears as the subject.  
    invite.setMessage("Try this out!\n -\\" + (FIRAuth.auth()?.currentUser?.displayName ?? "")")  
    // Title for the dialog, this is what the user sees before sending the invites.  
    invite.setTitle("Fire Chat 🔥")  
    invite.setDeepLink("https://a795j.app.goo.gl/6SuK")  
    invite.setCallToActionText("Install!")  
    invite.setCustomImage("http://i.telegraph.co.uk/multimedia/archive/02830/cat_2830677b.jpg")  
    invite.open()  
}
```

FIREBASE INVITES

- Customize the appearance, text and link

The screenshot shows the Firebase Invites interface. At the top, there's a header with a close button (X), a user icon (T), the text "Fire Chat" with a fire emoji, and a blue arrow icon. Below the header is a search bar with the placeholder "Add name, email or phone number". A section titled "Everyone" lists several users:

- aaggarw (aaggarw@ucls.uchicago.edu)
- Aaron Bowman (312-704-2133)
- Aaron Goldman (argold@uchicago.edu)
- abinkowski@uchicago.edu (abinkowski@uchicago.edu)
- Adam Gerber (gerber@uchicago.edu)
- Adam Gerber (gerber@cs.uchicago.edu)
- Adam Shaw (adamshaw@cs.uchicago.edu)
- Adam Stein ((847) 946-4004)

On the right side of the interface, there's a vertical column of letters from A to Z, likely representing filters or categories.

FIREBASE INVITES

```
private func inviteFinished(withInvitations invitationIds: [Any], error: Error?) {  
    if let error = error {  
        print("Failed: \(error.localizedDescription)")  
    } else {  
        print("Invitations sent")  
    }  
}
```

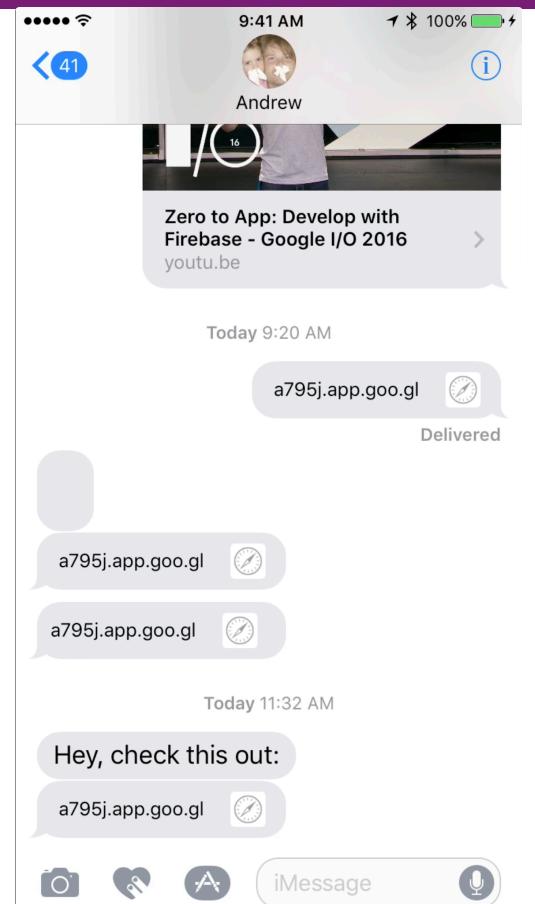
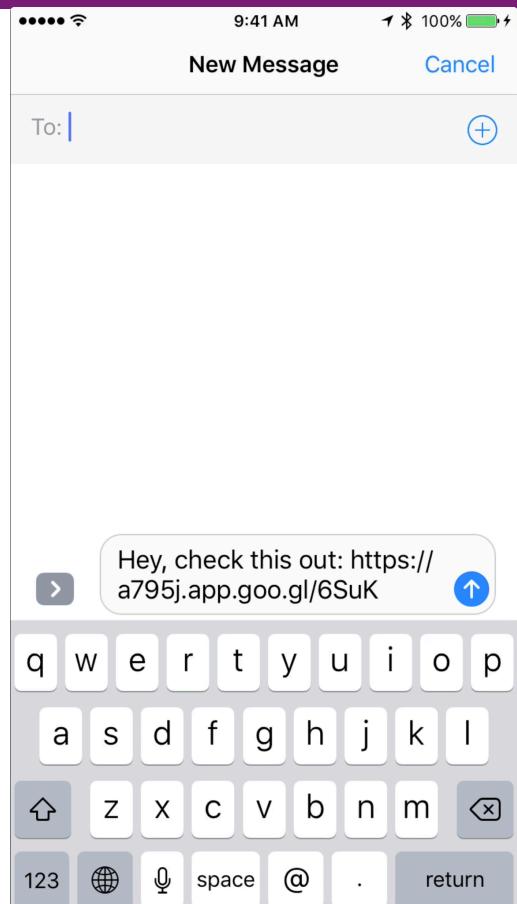
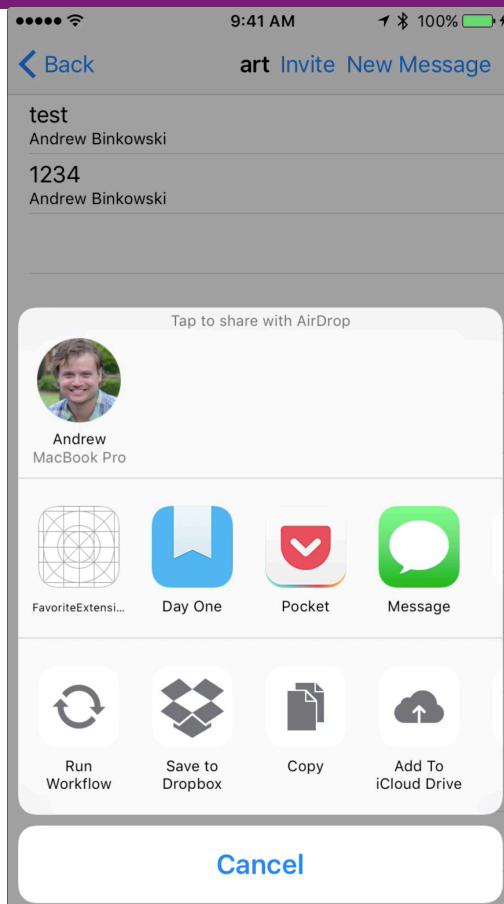
- Callback in case there was a problem

FIREBASE INVITES

```
// Standard UI way
let myDynamicLink = "LINK_TO_SHARE"
let msg = "Hey, check this out: " + myDynamicLink
let shareSheet = UIActivityViewController(activityItems: [ msg ],
                                         applicationActivities: nil)
shareSheet.popoverPresentationController?.sourceView = self.view
self.present(shareSheet, animated: true, completion: nil)
```

- Use standard UIKit to send a message

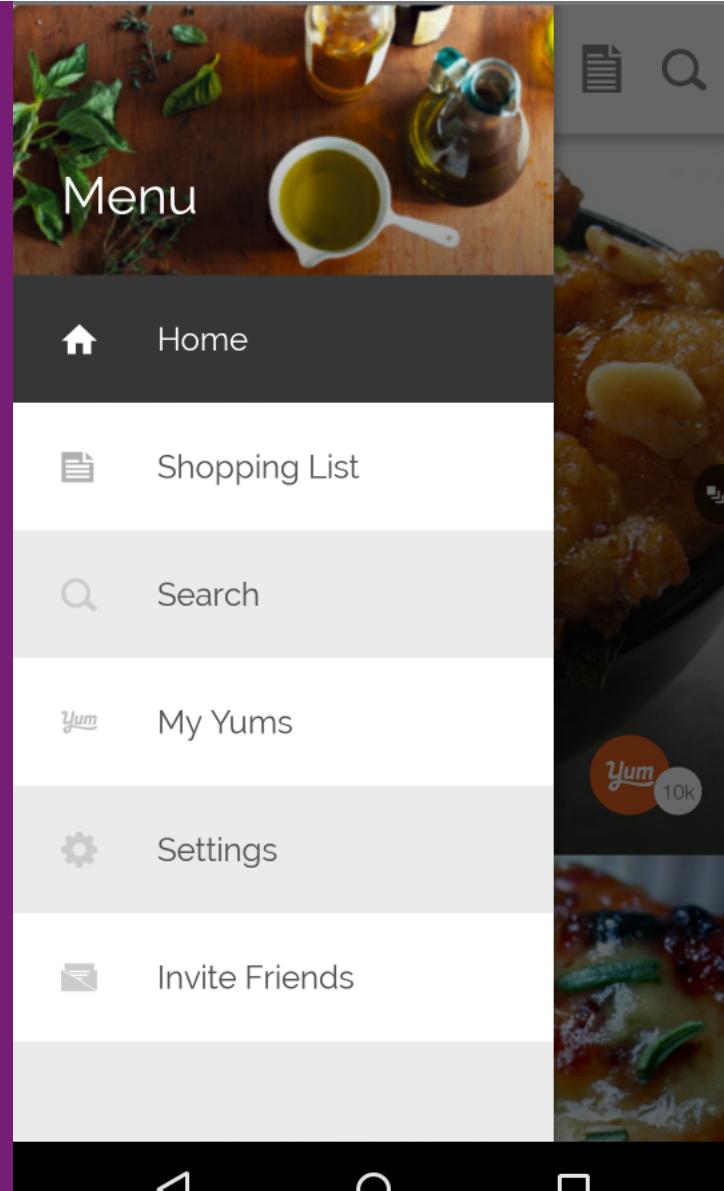
REALTIME DATABASE



BEST PRACTICES

INVITES BEST PRACTICES

- Make the invite and share options easy to find
- Build a custom share sheet
- Keep the message simple



INVITES

BEST PRACTICES

- Customize SMS invitation message from the sender
- Use HTML to customize email invites
 - Include shared content



Deborah Lai

to me

0 minutes ago [Details](#)



Hi,

I just found this great recipe using the free [Yummly iPhone app](#). Yummly is a great way to find recipe inspiration and create shopping lists. You should definitely [try it out](#).

Chewy Double Chocolate Peppermint Cookies

Jane



INVITES BEST PRACTICES

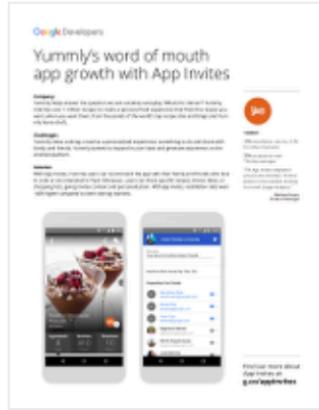
- Case studies

Fabulous



Fabulous is a research-based app incubated in Duke University's Center for Advanced Hindsight. It helps users to embark on a journey to reset their habits, replacing them with healthy rituals, with the goal of improving health and well-being. Learn how Fabulous achieved growth and increased their share links with Firebase Invites.

Yummly



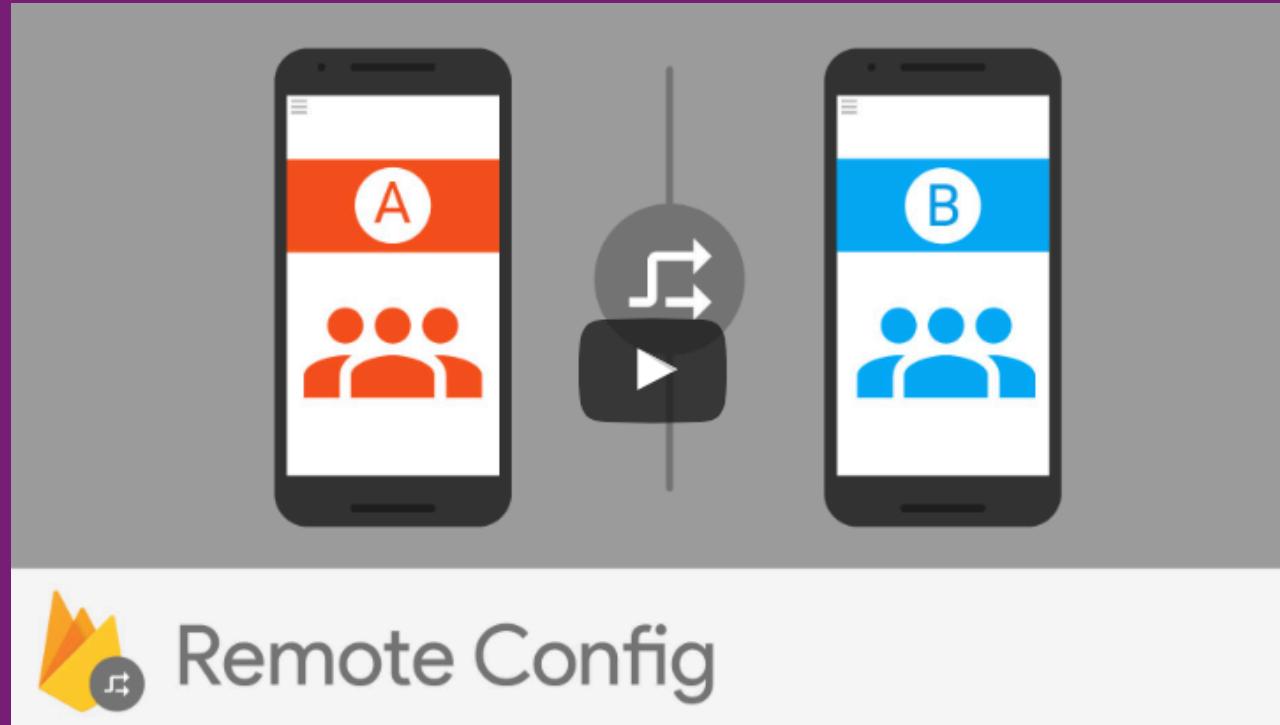
Yummly is a food discovery platform that views cooking a meal as a personalized, shareable experience. Yummly wanted to expand its user base and generate awareness on the Android platform, so they turned to Firebase Invites, and found installation rates to be ~60% higher than with other sharing channels. Learn how Yummly achieved this here.

BREAK TIME



FIREBASE REMOTE CONFIG

REMOTE CONFIG



- https://youtu.be/_CXXVFPO6f0?list=PLI-K7zZEsvYLmOF_07IayrTntevxtbUxDL

REMOTE CONFIG

- Firebase Remote Config is a cloud service that lets you change the behavior and appearance of your app without requiring users to download an app update

Remote Config Server

Your App

Which value is fetched from the server?

1) Highest priority conditional value (if **true** for a given app instance)

2) Server-side default value (if present)



Which value does my app get?

1) Value fetched from the server (if activated)

2) In-app default value (set using **setDefaults**)

3) Static initialized value

REMOTE CONFIG

- Create values that can be changed in the console to make them appear in your app
- There are plenty of ways to do this...this is just less work

Remote Config Server

Your App

Which value is fetched from the server?

1) Highest priority conditional value (if **true** for a given app instance)

2) Server-side default value (if present)



Which value does my app get?

1) Value fetched from the server (if activated)

2) In-app default value (set using **setDefaults**)

3) Static initialized value

REMOTE CONFIG

- Use cases

Quickly roll out changes to your app's userbase

You can make changes to your app's default behavior and appearance by changing service-side parameter values. For example, you could change your app's layout or color theme to support a seasonal promotion, with no need to publish an app update.

Customize your app for segments of your userbase

You can use Remote Config to provide variations on your app's user experience to different segments of your userbase by app version, by Firebase Analytics audience, by language, and more.

Run A/B tests to improve your app

You can use Remote Config random percentile targeting with Firebase Analytics to A/B test improvements to your app across different segments of your userbase so that you can validate improvements before rolling them out to your entire userbase.

REMOTE CONFIG

- Note the following policies
 - Don't use Remote Config to make app updates that should require a user's authorization. This could cause your app to be perceived as untrustworthy.
 - Don't store confidential data in Remote Config parameter keys or parameter values. It is possible to decode any parameter keys or values stored in the Remote Config settings for your project
 - Don't attempt to circumvent the requirements of your app's target platform using Remote Config.

REMOTE CONFIG

```
target 'FireChat' do
  # Comment the next line if you're not using Swift and don't want to use dynamic frameworks
  use_frameworks!

  # Pods for FireChat
  pod 'Firebase/Core'
  pod 'Firebase/Database'

  pod 'Firebase/Auth'
  pod 'GoogleSignIn'

  pod 'Firebase/Crash'

  pod 'Firebase/Storage'
  pod 'Firebase/Invites'

  pod 'Firebase/DynamicLinks'

  pod 'Firebase/RemoteConfig'

  pod 'SDWebImage'

end
```

REMOTE CONFIG

Overview



Analytics

DEVELOP

Authentication

Database

Storage

Hosting

Functions

Test Lab

Crash Reporting

GROW

Remote Config

PARAMETERS



Customize and experiment with app behavior using server-side configuration parameters

[Learn more](#)

ADD YOUR FIRST PARAMETER

REMOTE CONFIG

The screenshot shows the Firebase console interface for managing Remote Config parameters. The top navigation bar includes the Firebase logo, a dropdown for 'FireChat', and links for 'Go to docs' and user profile.

The left sidebar lists various Firebase products: Overview, Analytics, Authentication, Database, Storage, Hosting, Functions, Test Lab, and Crash Reporting. The 'Overview' item is currently selected.

The main content area is titled 'Remote Config' and displays the 'PARAMETERS' tab. A search bar at the top of this section allows searching for parameters, values, and conditions.

A modal dialog is open, prompting for a new parameter. The 'Parameter key' field contains the value 'joke_of_the_day'. The 'Default value' field is set to '(empty string)'. To the right of these fields is a button labeled 'Add value for condition ▾'. At the bottom of the modal are 'CANCEL' and 'ADD PARAMETER' buttons.

Below the modal, another parameter is listed: 'joke_of_the_day' with the value 'Why did the chicken cross the road?'. This value is preceded by a small icon.

REMOTE CONFIG

The screenshot shows the Firebase console interface for managing Remote Config parameters. The top navigation bar includes the Firebase logo, a dropdown for 'FireChat', and links for 'Go to docs', 'More', and a user profile icon.

The left sidebar lists various Firebase services: Overview, Analytics, Authentication, Database, Storage, Hosting, Functions, Test Lab, and Crash Reporting. A gear icon is also present in the sidebar.

The main content area is titled 'Remote Config' and features a 'PARAMETERS' tab. A prominent callout box provides instructions: 'Publish changes when you're ready. They'll be immediately available to your apps and users.' It includes a 'PUBLISH CHANGES' button and a 'DISCARD ALL' button.

A search bar at the bottom of the parameters section allows searching for parameters, values, and conditions. An 'ADD PARAMETER' button is located in the top right corner of this section.

Below the search bar, a parameter is listed with the key 'joke_of_the_day' and the value 'Why did the chicken cross the road?'. To the right of the value is a pencil icon for editing.

REMOTE CONFIG

The screenshot shows the Firebase Remote Config interface. On the left, there's a sidebar with various project services: Overview, Analytics, DEVELOP, Authentication, Database, Storage, Hosting, Functions, Test Lab, and Crash Reporting. Below these are sections for GROW and a settings gear icon. The main area is titled "Remote Config" and has tabs for "PARAMETERS" (selected), "CONDITIONS", and "EXPERIMENTS". A search bar at the top says "Search parameters, values and conditions". On the right, there are buttons for "PUBLISH CHANGES", "DISCARD ALL", and a question mark icon. The "PARAMETERS" section lists two parameters: "joke_of_the_day" with the value "Why did the chicken cross the road?" and "show_joke_emoji" with the value "true".

Overview

Analytics

DEVELOP

Authentication

Database

Storage

Hosting

Functions

Test Lab

Crash Reporting

GROW

Remote Config

PUBLISH CHANGES

DISCARD ALL

?

PARAMETERS

Search parameters, values and conditions

ADD PARAMETER

Parameter	Value
joke_of_the_day	Why did the chicken cross the road?
show_joke_emoji	true

REMOTE CONFIG

- Set default values
 - Programmatically
 - Include a plist
- Fetch latest content
- Activate Fetched values

```
import Foundation
import Firebase

class RemoteConfigManager {

    static let sharedInstance = RemoteConfigManager()
    let remoteConfig = FIRRemoteConfig.remoteConfig()

    private init() {
        activateDebugMode()
        loadDefaultValues()
    }

    func loadDefaultValues() {
        let appDefaults: [String: NSObject] = ["joke_of_the_day": "Default Joke" as NSObject]
        remoteConfig.setDefaults(appDefaults)
    }

    /**
     func fetchCloudValues() {

        let fetchDuration: TimeInterval = 0
        activateDebugMode()
        remoteConfig.fetch(withExpirationDuration: TimeInterval(fetchDuration)) { (status, error) -> Void in

            if status == .success {
                print("Config fetched!")
                self.remoteConfig.activateFetched()

            } else {
                print("Config not fetched")
                print("Error \(error!.localizedDescription)")
            }
            print("Joke of the day: \(self.remoteConfig["joke_of_the_day"].stringValue!)")
        }
    }

    func activateDebugMode() {
        let debugSettings = FIRRemoteConfigSettings(developerModeEnabled: true)
        remoteConfig.configSettings = debugSettings!
    }
}
```

REMOTE CONFIG

- Careful of throttling

```
import Foundation
import Firebase

class RemoteConfigManager {

    static let sharedInstance = RemoteConfigManager()
    let remoteConfig = FIRRemoteConfig.remoteConfig()

    private init() {
        activateDebugMode()
        loadDefaultValues()
    }

    func loadDefaultValues() {
        let appDefaults: [String: NSObject] = ["joke_of_the_day": "Default Joke" as NSObject]
        remoteConfig.setDefaults(appDefaults)
    }

    /**
     func fetchCloudValues() {

        let fetchDuration: TimeInterval = 0
        activateDebugMode()
        remoteConfig.fetch(withExpirationDuration: TimeInterval(fetchDuration)) { (status, error) -> Void in

            if status == .success {
                print("Config fetched!")
                self.remoteConfig.activateFetched()

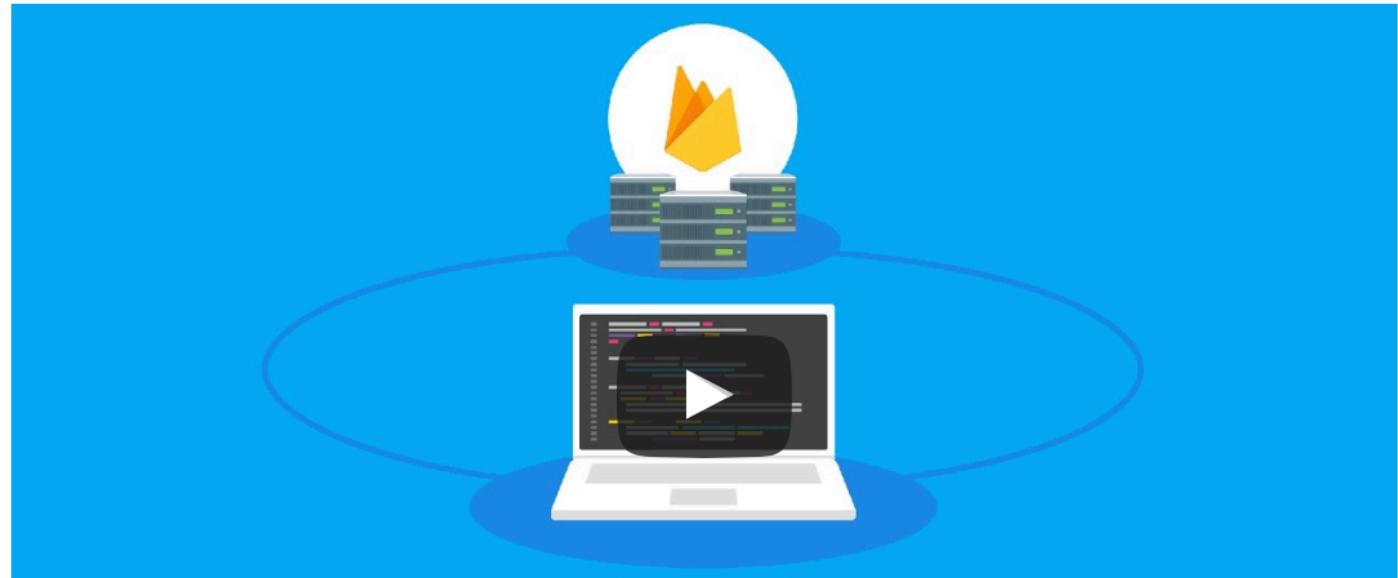
            } else {
                print("Config not fetched")
                print("Error \(error!.localizedDescription)")
            }
            print("Joke of the day: \(self.remoteConfig["joke_of_the_day"].stringValue!)")
        }
    }

    func activateDebugMode() {
        let debugSettings = FIRRemoteConfigSettings(developerModeEnabled: true)
        remoteConfig.configSettings = debugSettings!
    }
}
```

FIREBASE CLOUD FUNCTIONS

🔥 FIREBASE CLOUD FUNCTIONS

- <https://youtu.be/vr0Gfvp5v1A>



Cloud Functions
for Firebase



CLOUD FUNCTIONS

- Integrates with Firebase in different ways
 - Realtime Database Triggers
 - Firebase Authentication Triggers
 - Firebase Analytics Triggers
 - Cloud Storage Triggers
 - Cloud Pub/Sub Triggers
 - HTTP Triggers





CLOUD FUNCTIONS

- Integrates with Firebase in different ways
 - Realtime Database Triggers
 - Firebase Authentication Triggers
 - Firebase Analytics Triggers
 - Cloud Storage Triggers
 - Cloud Pub/Sub Triggers
 - HTTP Triggers





FIREBASE

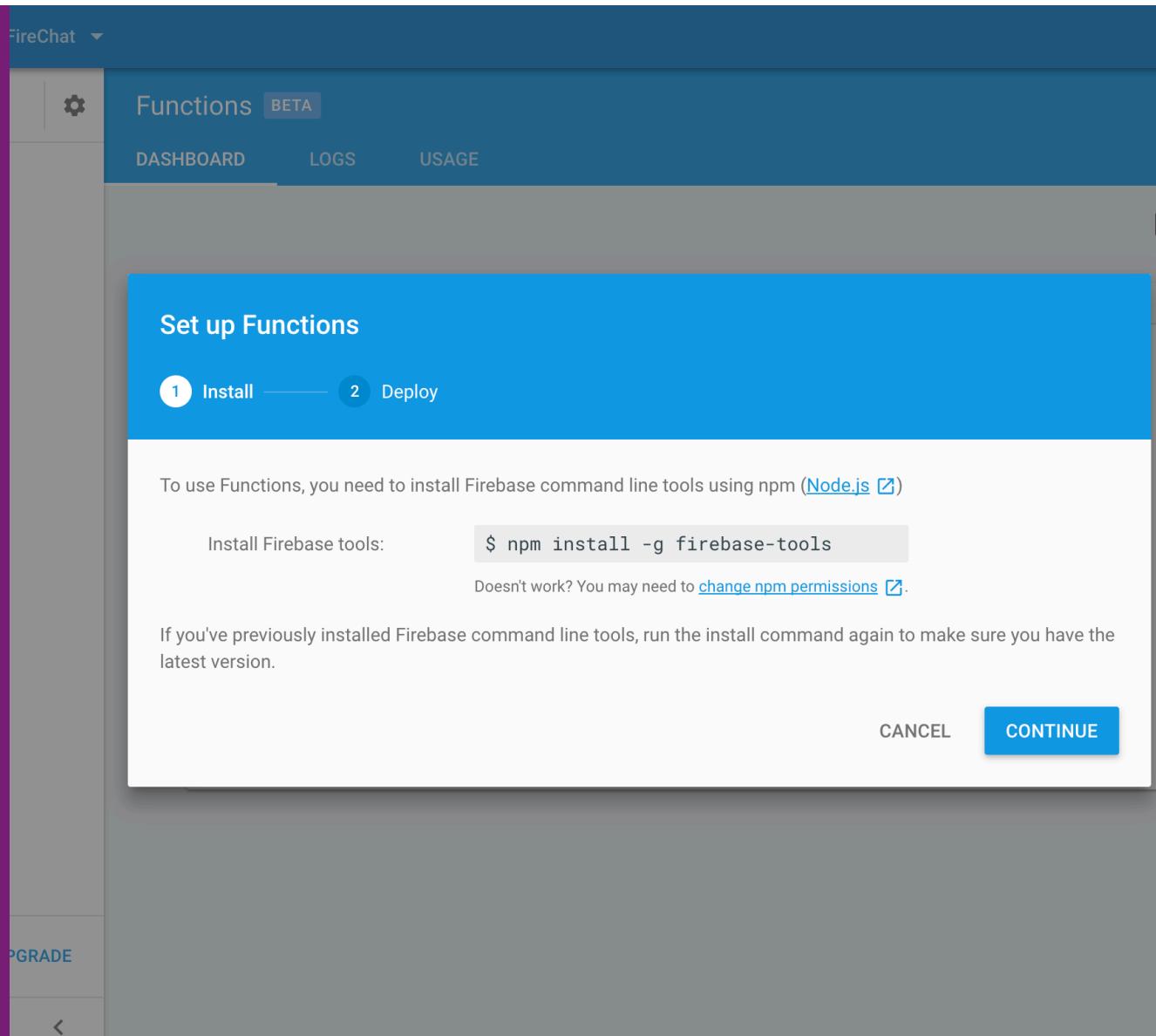
CLOUD FUNCTIONS

```
tabinkowski@Ts-MacBook-Pro ~  
521 % brew install node
```

- Need to install node first

FIREBASE CLOUD FUNCTIONS

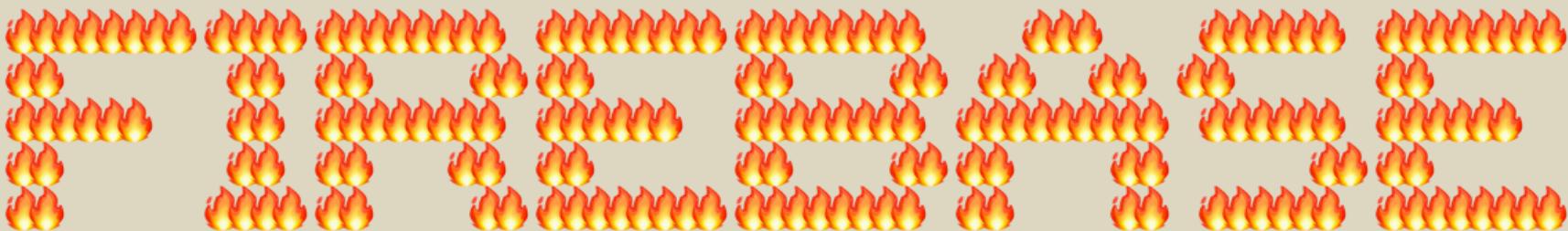
- Use node locally to test and deploy
- Uses express framework



The screenshot shows the Firebase Functions setup interface. At the top, there's a navigation bar with tabs: DASHBOARD, LOGS, and USAGE. On the far right, there are date filters for "Current" (Oct 1 - Oct 6) and "Median run time". Below the navigation, a large blue header box says "Set up Functions" with two numbered steps: "1 Install" and "2 Deploy". A sub-instruction below says: "To use Functions, you need to install Firebase command line tools using npm ([Node.js](#))". It provides a command line input field containing "\$ npm install -g firebase-tools" and a note: "Doesn't work? You may need to [change npm permissions](#)". A note at the bottom states: "If you've previously installed Firebase command line tools, run the install command again to make sure you have the latest version." At the bottom right are "CANCEL" and "CONTINUE" buttons.

🔥 FIREBASE CLOUD FUNCTIONS

```
533 % firebase init functions
```



You're about to initialize a Firebase project in this directory:

/Users/tabinkowski

Before we get started, keep in mind:

- * You are initializing your home directory as a Firebase project

==== Project Setup

FIREBASE CLOUD FUNCTIONS

? Which Firebase CLI features do you want to setup for this folder? Press Space to select features, then Enter to confirm your choices. Functions: Configure and deploy Cloud Functions

==== Project Setup

First, let's associate this project directory with a Firebase project. You can create multiple project aliases by running `firebase use --add`, but for now we'll just set up a default project.

? Select a default Firebase project for this directory: [create a new project]

==== Functions Setup

A `functions` directory will be created in your project with a Node.js package pre-configured. Functions can be deployed with `firebase deploy`.

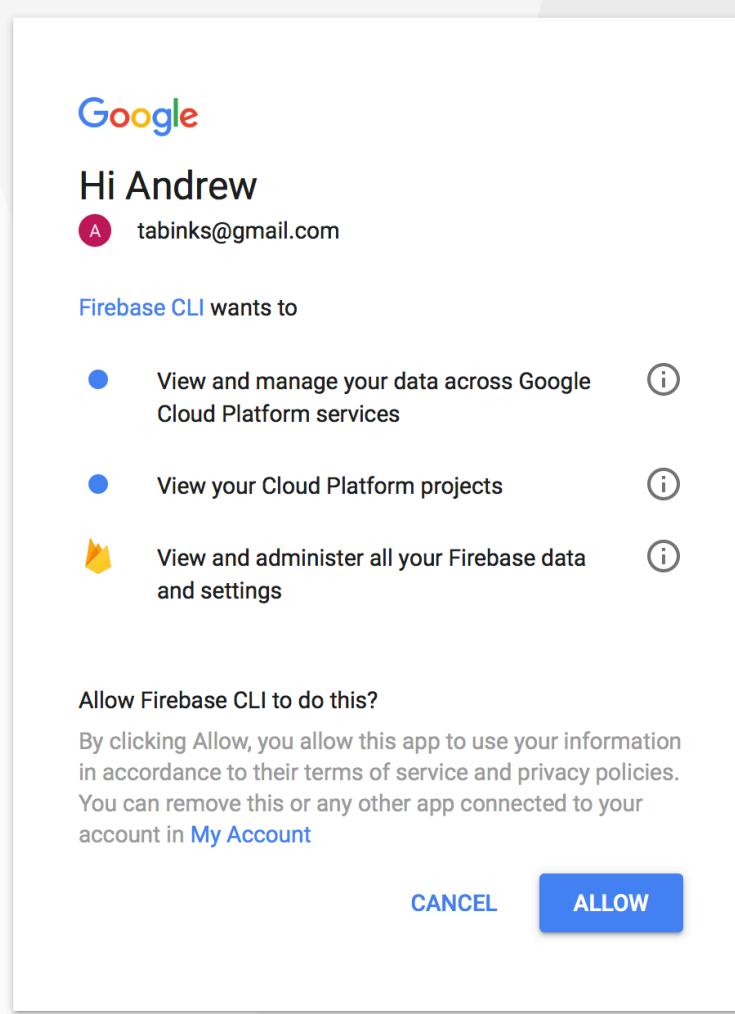
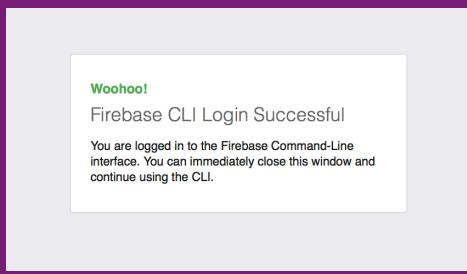
✓ wrote `functions/package.json`

FIREBASE CLOUD FUNCTIONS

```
tabinkowski@Ts-MacBook-Pro ~/Google Drive/g-Teaching/uchicago.cloud/mpcs51033-2017-autumn/mpcs51033-2017-aut [515] % firebase login
? Allow Firebase to collect information? No
[ ] FIREBASE LOGIN
[ ]
Visit this URL on any dev
https://accounts.google.com/o/oauth2/auth?client\_id=50550455887-1grhgmd47bqnekij5i8b5pr03ho849e6.apps.googleusercontent.com&scope=email%20openid%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloudplatformprojects.readonly%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Ffirebase%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform&response\_type=code&state=1041200543&redirect\_uri=http%3A%2F%2Flocalhost%3A9005
Waiting for authentication...
```

🔥 FIREBASE CLOUD FUNCTIONS

- Give Firebase permissions



English (United States) ▾

Help

Privacy

Terms



FIREBASE

CLOUD FUNCTIONS

```
? Which Firebase CLI features do you want to setup for this folder? Press Space to  
select features, then Enter to confirm your choices.
```

```
⚠ You have not selected any features. Continuing will simply associate this folder with a Firebase project. Press Ctrl + C if you want to start over.
```

==== Project Setup

First, let's associate this project directory with a Firebase project. You can create multiple project aliases by running `firebase use --add`, but for now we'll just set up a default project.

```
? Select a default Firebase project for this directory:
```

[don't setup a default project]

Firebase Demo Project (fir-demo-project)

HuntHunt (hunthunt-77d3b)

› FireChat (firechat-66f87)

ChattyCathy (chattycathy-56d4b)

[create a new project]

🔥 FIREBASE CLOUD FUNCTIONS

```
functions — more package.json — 83×24
tabinkowski@Ts-MacBook-Pro ~/Google Drive/g-Teaching/uchicago.cloud/mpcs51033-2017-
autumn/mpcs51033-2017-autumn-code-samples.firebaseio/functions
[548 % ls
index.js      node_modules  package.json
tabinkowski@Ts-MacBook-Pro ~/Google Drive/g-Teaching/uchicago.cloud/mpcs51033-2017-
autumn/mpcs51033-2017-autumn-code-samples.firebaseio/functions
[549 % more package.json
{
  "name": "functions",
  "description": "Cloud Functions for Firebase",
  "dependencies": {
    "firebase-admin": "~5.4.0",
    "firebase-functions": "^0.7.0"
  },
  "private": true
}
package.json (END)
```

FIREBASE CLOUD FUNCTIONS

index.js — ~/Google Drive/g-Teaching/uchicago.cloud/mpcs51033-2017-autumn/mpcs51033-2017-autumn-code-samples.firebaseio/functions

```
index.js          index.js~          x
1 const functions = require('firebase-functions');
2
3 // Create and Deploy Your First Cloud Functions
4 // https://firebase.google.com/docs/functions/write-firebase-functions
5
6 exports.helloWorld = functions.https.onRequest((request, response) => {
7   response.send("Hello from Firebase!");
8 });
9
```

DEFAULT
FUNCTION

😊

+ x index.js ⓘ 0 ⚠ 0 ⓘ 0 2:1

● LF UTF-8 JavaScript 📄 0 files 📂 1 update

FIREBASE CLOUD FUNCTIONS

```
552 % emacs index.js
[tabinkowski@Ts-MacBook-Pro ~/Google Drive/g-Teaching/uchicago.cloud/mpcs51033-2017-]
autumn/mpcs51033-2017-autumn-code-samples.firebaseio/functions
[553 % atom index.js
tabinkowski@Ts-MacBook-Pr
autumn/mpcs51033-2017-aut
[554 % firebase deploy
1

==== Deploying to 'firechat-66f87'...

i  deploying functions
i  functions: ensuring necessary APIs are enabled...
i  runtimeconfig: ensuring necessary APIs are enabled...
✓  runtimeconfig: all necessary APIs are enabled
✓  functions: all necessary APIs are enabled
i  functions: preparing functions directory for uploading...
```



FIREBASE DEPLOY



FIREBASE

CLOUD FUNCTIONS

```
==== Deploying to 'firechat-66f87'...
```

```
i  deploying functions
[i  functions: ensuring necessary APIs are enabled...
i  runtimeconfig: ensuring necessary APIs are enabled...
✓  runtimeconfig: all necessary APIs are enabled
✓  functions: all necessary APIs are enabled
[i  functions: preparing functions directory for uploading...
[i  functions: packaged functions (1.14 KB) for uploading
✓  functions: functions folder uploaded successfully
i  starting release process (may take several minutes)...
i  functions: creating function helloWorld...
✓  functions[helloWorld]: Successful create operation.
✓  functions: all functions deployed successfully!

✓  Deploy complete!
```

Project Console: <https://console.firebaseio.google.com/project/firechat-66f87/overview>

Function URL (helloWorld): <https://us-central1-firechat-66f87.cloudfunctions.net/helloWorld>

tabinkowski@Ts-MacBook-Pro ~/Google Drive/g-Teaching/uchicago.cloud/mpcs51033-2017-autumn/mpcs51033-2017-autumn-code-samples.firebaseio/functions

FIREBASE CLOUD FUNCTIONS

A screenshot of a web browser window. The address bar shows the URL `us-central1-firechat-66f87.cloudfunctions.net`. The page content displays the text "Hello from Firebase!".

Hello from Firebase!

A screenshot of the Firebase Cloud Functions console. The left sidebar shows "Overview", "Analytics", "DEVELOP", "Authentication", "Database", "Storage", and "Hosting". The main area is titled "Functions BETA". It shows a table with one row:

Function	Event	Executions	Median run time
helloWorld	HTTP Request <code>https://us-central1-firechat-66f87.cloudfunctions.net/hel...</code>	0	—

At the bottom right, it says "Current billing period Oct 1 - Oct 31, 2017 (CDT)".

DATABASE TRIGGERS

DATABASE TRIGGERS

- Make updates using the snapshot of Firebase Admin SDK



DATABASE TRIGGERS

- In a typical lifecycle, a Firebase Realtime Database function does the following:
- 1. Waits for changes to a particular database location.
- 2. Triggers when an event occurs and performs its tasks (see [What can I do with Cloud Functions?](#)for examples of use cases).
- 3. Receives an event data object that contains two snapshots of the data stored at the specified path: one with the original data prior to the change, and one with the new data.



DATABASE TRIGGERS

`onWrite()` , which triggers when data is created, destroyed, or changed in the Realtime Database.

`onCreate()` , which triggers when new data is created in the Realtime Database.

`onUpdate()` , which triggers when data is updated in the Realtime Database.

`onDelete()` , which triggers when data is deleted from the Realtime Database.

- Event handlers for triggers

DATABASE TRIGGERS

- Triggers operate when a node path is met
- Try to use the deepest path match you can

```
# Trigger a call at /foo/bar
functions.database.ref('/foo/bar')
```

```
# Both paths will trigger the
# cloud function
/foo/bar
/foo/bar/baz/really深深/path
```

DATABASE TRIGGERS

- Use wildcards to match keys
- Path available at `event.params`

```
# Trigger a call at /foo/bar
functions.database.ref('/foo/{bar}/')

# Wildcards may be called multiple
# times depending on level
{
  "foo": {
    "hello": "world",
    "firebase": "functions"
  }
}
```

DATABASE TRIGGERS

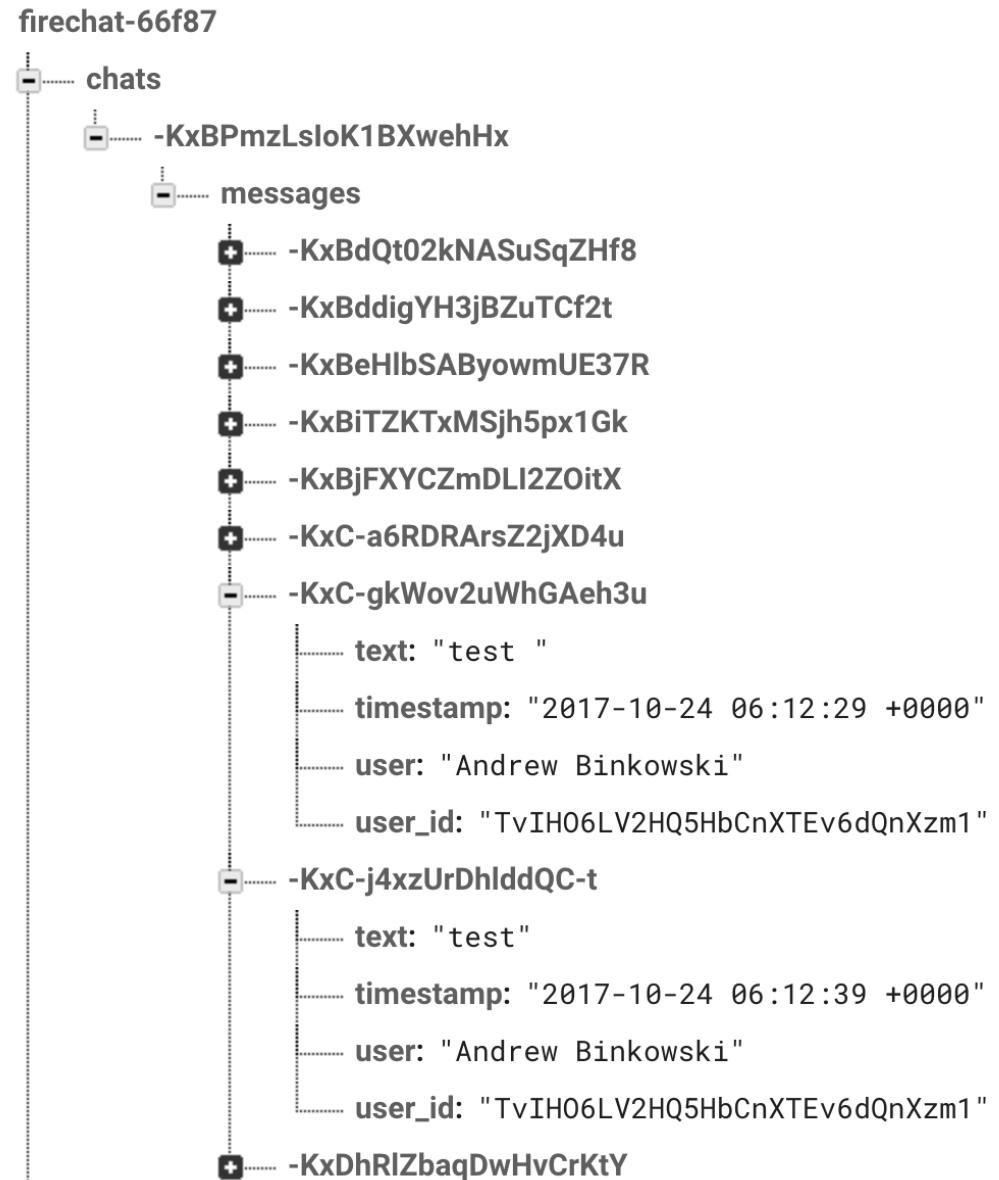
- Use wildcards to match keys
- Path available at `event.params`

```
# Trigger a call at /foo/bar
functions.database.ref('/foo/{bar}/')

# Wildcards may be called multiple
# times depending on level
{
  "foo": {
    "hello": "world",
    "firebase": "functions"
  }
}
```

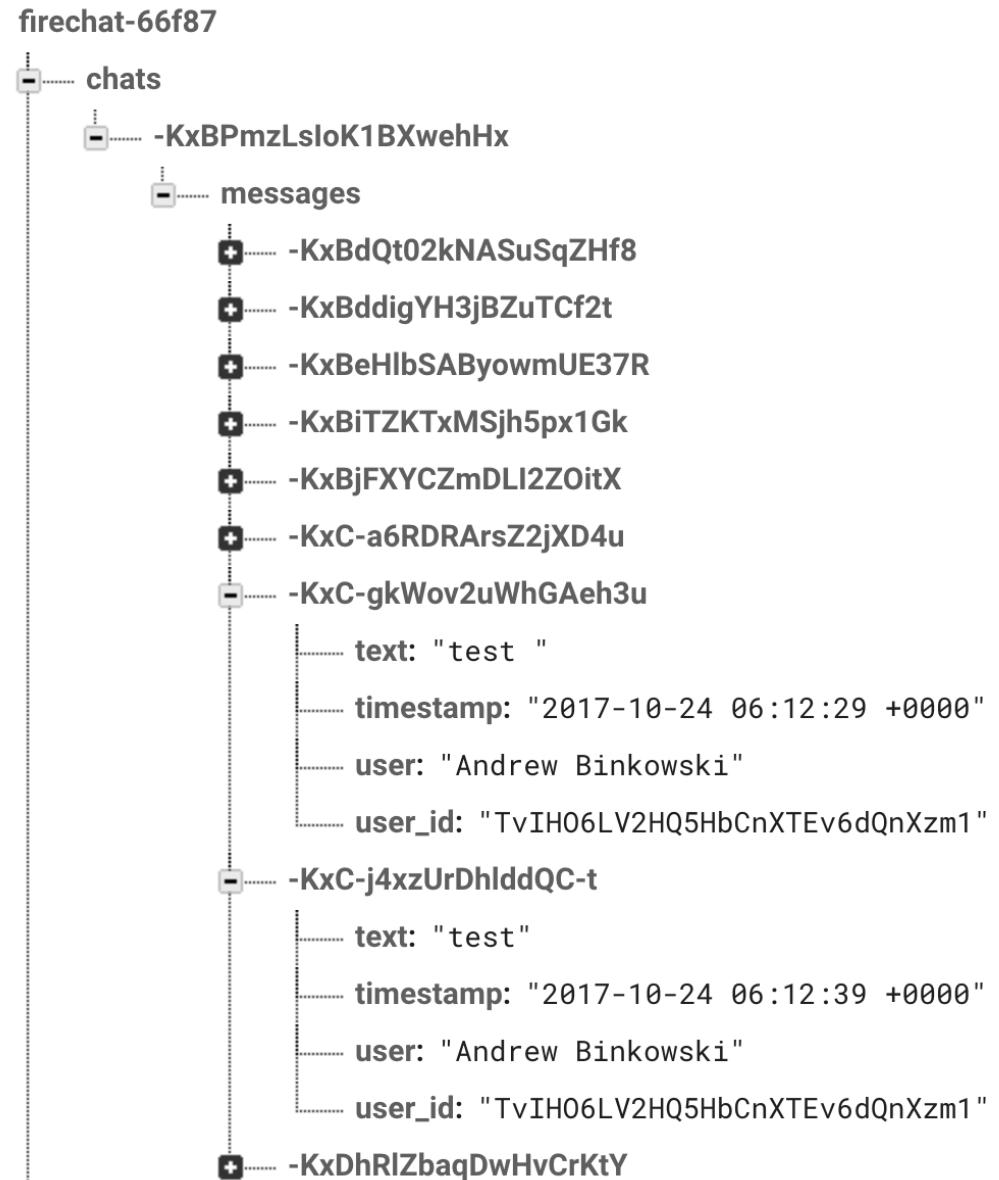
DATABASE TRIGGERS

- Current structure of our data in the realtime database
- Path:/chats/{chatId}/messages/{messageId}/text



DATABASE TRIGGERS

- Current structure of our data in the realtime database
- Path:/chats/{chatId}/messages/{messageId}/text



DATABASE TRIGGERS

```
exports.makeUppercase = functions.database.ref('/chats/{chatId}/messages/{ messageId}/text')  
....onWrite(event => {  
  ....  
  ....// Grab the current value of what was written to the Realtime Database.  
  ....const original = event.data.val();  
  ....  
  ....// Log out so we can debug  
  ....console.log('Uppercasing', event.params.chatId, event.params.messageId, original);  
  ....const uppercase = original.toUpperCase();  
  ....  
  ....// You must return a Promise when performing asynchronous tasks inside a  
  ....// Functions such as writing to the Firebase Realtime Database.  
  ....// Setting an "uppercase" sibling returns a Promise  
  ....const promise = event.data.ref.parent.child('uppercase').set(uppercase);  
  ....return promise  
});
```

DATABASE TRIGGERS

- Current structure of our data in the realtime database
- Path:/chats/{chatId}/messages/{messageId}/text

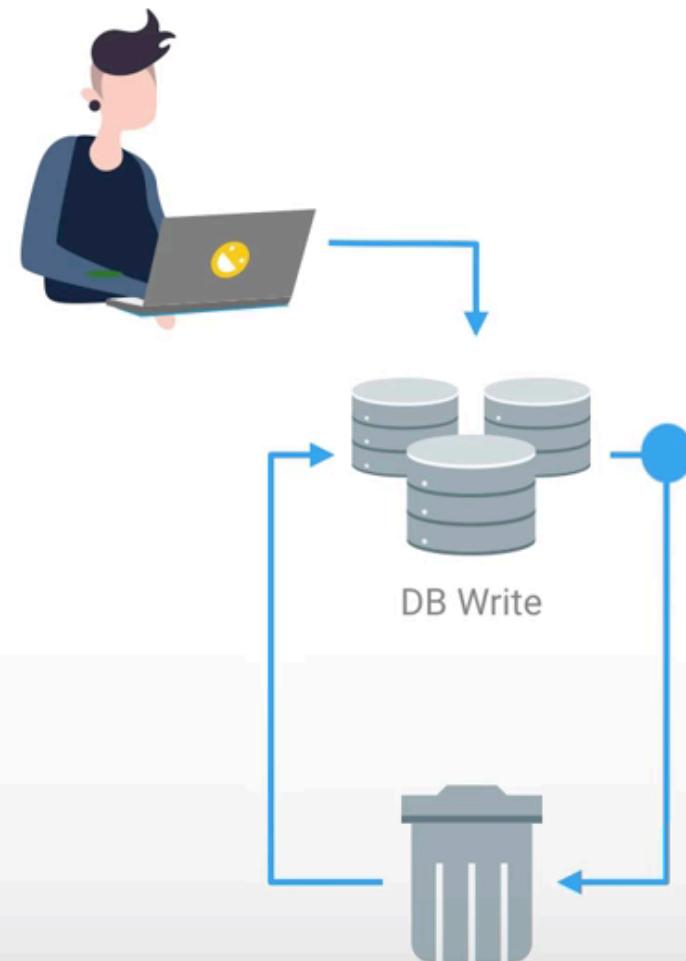


DATABASE TRIGGERS

Time ↓	Level	Function	Event message
Oct 24, 2017			
9:07:...	🚩	makeUp...	▶ Function execution took 1091 ms, finished with status: ...
9:07:...	ℹ️	makeUp...	▼ Uppercasing -KxBPmzLsIoK1BXwehHx -KxDhRlZbaqDwHvCrKtY cloud function test
9:07:...	🚩	makeUp...	▶ Billing account not configured. External network is not... available.
9:07:...	🚩	makeUp...	Function execution started
1:31:...	🚩	addMes...	▶ Function execution took 621 ms, finished with status co...

DATABASE TRIGGERS

- Careful that changes do not trigger infinite loop



DATABASE TRIGGERS

```
exports.makeUppercase = functions.database.ref('/chats/{chatId}/messages/{ messageId}/text')  
....onWrite(event => {  
  ....  
  ....// Grab the current value of what was written to the Realtime Database.  
  ....const original = event.data.value();  
  ....  
  ....// Log out so we can debug  
  ....console.log('Uppercasing', event.params.chatId, event.params.messageId, original);  
  ....const uppercase = original.toUpperCase();  
  ....  
  ....// You must return a Promise when performing asynchronous tasks inside a  
  ....// Functions such as writing to the Firebase Realtime Database.  
  ....// Setting an "uppercase" sibling returns a promise  
  ....const promise = event.data.ref.parent.child('uppercase').set(uppercase);  
  ....return promise  
});
```



DATABASE TRIGGERS

- Authentication trigger
- Welcome email via gmail API

```
// Adds a message that welcomes new users.
exports.addWelcomeMessages = functions.auth.user().onDelete(event => {
  const user = event.data; // The Firebase user.
  const email = user.email; // The email of the user.
  const displayName = user.displayName; // The display name of the user.
  return sendWelcomeEmail(email, displayName);
});

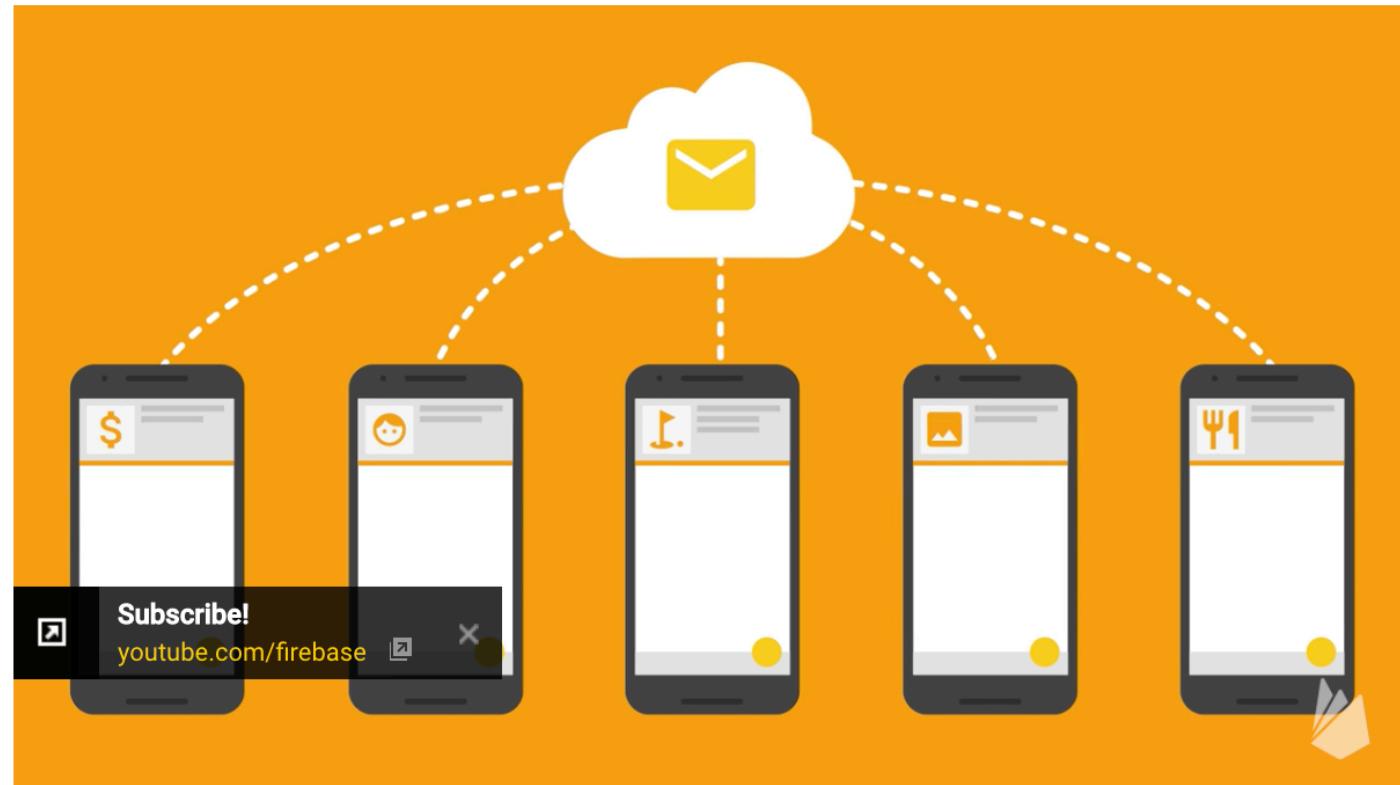
// Sends a welcome email to the given user.
function sendWelcomeEmail(email, displayName) {
  const mailOptions = {
    from: `${APP_NAME} <noreply@firebase.com>`,
    to: email
  };

  // The user subscribed to the newsletter.
  mailOptions.subject = `Welcome to ${APP_NAME}!`;
  mailOptions.text = `Hey ${displayName} || ''! Welcome to ${APP_NAME}.`;
  return mailTransport.sendMail(mailOptions).then(() => {
    console.log('New welcome email sent to:', email);
  });
}
```

FIREBASE CLOUD MESSAGING

FIREBASE CLOUD MESSAGING

- https://youtu.be/sioEY4tWmLI?list=PLI-K7zZEesYLmOF_07layrTntevxtbUxDL



SETTING UP FOR APNS

SETTING UP APNS

- Messaging uses native APNS services to deliver messages
 - Default

Notifications in Your App

Apple Push Notification Service

APNs Overview

Creating the Remote Notification Payload

Communicating with APNs

Payload Key Reference

Legacy Information

Revision History

Communicating with APNs

The APNs provider API lets you send remote notification requests to APNs. APNs then conveys notifications to your app on iOS, tvOS, and macOS devices, and to Apple Watch via iOS.

The provider API is based on the HTTP/2 network protocol. Each interaction starts with a POST request, from your provider, that contains a JSON payload and a device token. APNs forwards the notification payload to your app on the specific user device identified by the request's included device token.

A *provider* is a server, that you deploy and manage, that you configure to work with APNs.

Provider Authentication Tokens

To securely connect to APNs, you can use provider authentication tokens or provider certificates. This section describes connections using tokens.

The provider API supports the JSON Web Token (JWT) specification, letting you pass statements and metadata, called *claims*, to APNs, along with each push notification. For details, refer to the specification at <https://tools.ietf.org/html/rfc7519>. For additional information about JWT, along with a list of available libraries for generating signed JWTs, see <https://jwt.io>

A provider authentication token is a JSON object that you construct, whose header must include:

- The encryption algorithm (*alg*) you use to encrypt the token
- A 10-character *key identifier* (*kid*) key, obtained from [your developer account](#)

The claims payload of the token must include:

- The *issuer* (*iss*) registered claim key, whose value is your 10-character Team ID, obtained from [your developer account](#)
- The *issued at* (*iat*) registered claim key, whose value indicates the time at which the token was generated, in terms of the number of seconds since Epoch, in UTC

After you create the token, you must sign it with a private key. You must then encrypt the token using the Elliptic Curve Digital Signature Algorithm (ECDSA) with the P-256 curve and the SHA-256 hash algorithm. Specify the value `ES256` in the algorithm header key (*alg*). For information on how to configure your token, see

SETTING UP APNS

- 1. Set up app in Xcode
- 2. Create app in developer portal
- 3. Add certificates for authentication
- 4. Upload this certificate to provider (eg. Firebase)

Programming Guide

Communicating with APNs

The APNs provider API lets you send remote notification requests to APNs. APNs forwards your app's notifications to the user's device.

The provider API is based on the HTTP/2 network protocol. Each interaction starts with a POST request from your app to your provider, that contains a JSON payload and a device token. APNs forwards the notification to the user's device, and your app receives a response from APNs.

A *provider* is a server, that you deploy and manage, that you configure to work with your app.

Provider Authentication Tokens

To securely connect to APNs, you can use provider authentication tokens or private keys. A provider authentication token is a JSON object that describes connections using tokens.

The provider API supports the JSON Web Token (JWT) specification, letting you add metadata, called *claims*, to APNs, along with each push notification. For details about the claims supported by APNs, see <https://tools.ietf.org/html/rfc7519>. For additional information about JWT, along with instructions for generating signed JWTs, see <https://jwt.io>.

A provider authentication token is a JSON object that you construct, whose header includes:

- The encryption algorithm (*alg*) you use to encrypt the token
- A 10-character *key identifier* (*kid*) key, obtained from [your developer account](#)

The claims payload of the token must include:

- The *issuer* (*iss*) registered claim key, whose value is your 10-character [provider identifier](#)
- The *issued at* (*iat*) registered claim key, whose value indicates the time the token was generated, in terms of the number of seconds since Epoch, in UTC

After you create the token, you must sign it with a private key. You must then encode the token using the `base64url` encoding scheme and use the `POST` method to send the token to APNs. APNs uses the Elliptic Curve Digital Signature Algorithm (ECDSA) with the P-256 curve and the SHA-256 hash function to verify the token. Specify the value `ES256` in the algorithm header key (*alg*). For information on how to generate a private key, see [Configure push notifications](#) in Xcode Help.

FIREBASE CLOUD MESSAGING

Finished running FireChat on T. Andrew Binkowski's iPhone 6 ⚠ 4 ! 1

Identity and Type

Name: FireChat
Location: Relative to Group
Full Path: /Users/tabinkowski/Google Drive/g-Teaching/uchicago.cloud/mpcs51033-2017-autumn/mpcs51033-2017-autumn-code-samples.firebaseio/FireChat2/FireChat.xcodeproj

Project Document

Project Format: Xcode 3.2-compatible
Organization: T. Andrew Binkowski
Class Prefix:

Capabilities

Category	Feature	Status
TARGETS	iCloud	OFF
	Push Notifications	ON
	Game Center	OFF
	Wallet	OFF
	Siri	OFF
	Apple Pay	OFF
	In-App Purchase	OFF
	Maps	OFF
FireChat	Personal VPN	OFF

Identity and Type

Name: FireChat
Location: Relative to Group
Full Path: /Users/tabinkowski/Google Drive/g-Teaching/uchicago.cloud/mpcs51033-2017-autumn/mpcs51033-2017-autumn-code-samples.firebaseio/FireChat2/FireChat.xcodeproj

Project Document

Project Format: Xcode 3.2-compatible
Organization: T. Andrew Binkowski
Class Prefix:

Capabilities

Category	Feature	Status
TARGETS	iCloud	OFF
	Push Notifications	ON
	Game Center	OFF
	Wallet	OFF
	Siri	OFF
	Apple Pay	OFF
	In-App Purchase	OFF
	Maps	OFF
FireChat	Personal VPN	OFF

Identity and Type

Name: FireChat
Location: Relative to Group
Full Path: /Users/tabinkowski/Google Drive/g-Teaching/uchicago.cloud/mpcs51033-2017-autumn/mpcs51033-2017-autumn-code-samples.firebaseio/FireChat2/FireChat.xcodeproj

Project Document

Project Format: Xcode 3.2-compatible
Organization: T. Andrew Binkowski
Class Prefix:

Capabilities

Category	Feature	Status
TARGETS	iCloud	OFF
	Push Notifications	ON
	Game Center	OFF
	Wallet	OFF
	Siri	OFF
	Apple Pay	OFF
	In-App Purchase	OFF
	Maps	OFF
FireChat	Personal VPN	OFF

FIREBASE CLOUD MESSAGING

	Name	Type	Expires
<input checked="" type="checkbox"/> Pending	Alice Chang (Alice's MacBook Air)	iOS Development	Jun 07, 2018
<input checked="" type="checkbox"/> Development	Andrew Binkowski (T.'s MacBook Pro)	iOS Development	Mar 20, 2018

Pending

Development

Production

Keys

All

Identifiers

App IDs

Pass Type IDs

iCloud Containers

App Groups

Devices

All

Apple TV

Apple Watch

FIREBASE CLOUD MESSAGING

iOS, tvOS, watchOS ▾

 Certificates

- All
- Pending
- Development
- Production

 Keys

- All

 Identifiers

- App IDs
- Pass Type IDs
- iCloud Containers
- App Groups

Add iOS Certificate

Select Type Request Generate Download

 What type of certificate do you need?

Development

- iOS App Development**
Sign development versions of your iOS app.
- Apple Push Notification service SSL (Sandbox)**
Establish connectivity between your notification server and the Apple Push Notification service sandbox environment to deliver remote notifications to your app. A separate certificate is required for each app you develop.

FIREBASE CLOUD MESSAGING

- Pending
- Development
- Production

 **Keys**

- All

 **Identifiers**

- App IDs
- Pass Type IDs
- iCloud Containers
- App Groups

 **Devices**

- All
- Apple TV
- Apple Watch
- iPad



Which App ID would you like to use?

All App IDs that you want to enable for remote notifications require their own Apple Push Notification service SSL certificate. The App ID-specific SSL certificate allows your server to connect to the Apple Push Notification service. Note that only explicit App IDs with a specific Bundle Identifier can be used to create an Apple Push Notification service SSL certificate.

Select an App ID for your Apple Push Notification service SSL Certificate (Sandbox)

App ID:

FIREBASE CLOUD MESSAGING

- Pending
- Development
- Production

Keys

- All

Identifiers

- App IDs
- Pass Type IDs
- iCloud Containers
- App Groups

Devices

- All
- Apple TV
- Apple Watch
- iPad



About Creating a Certificate Signing Request (CSR)

To manually generate a Certificate, you need a Certificate Signing Request (CSR) file from your Mac. To create a CSR file, follow the instructions below to create one using Keychain Access.

Create a CSR file.

In the Applications folder on your Mac, open the Utilities folder and launch Keychain Access.

Within the Keychain Access drop down menu, select Keychain Access > Certificate Assistant > Request a Certificate from a Certificate Authority.

- In the Certificate Information window, enter the following information:
 - In the User Email Address field, enter your email address.
 - In the Common Name field, create a name for your private key (e.g., John Doe Dev Key).
 - The CA Email Address field should be left empty.
 - In the "Request is" group, select the "Saved to disk" option.
- Click Continue within Keychain Access to complete the CSR generating process.

FIREBASE CLOUD MESSAGING

Continue to request a certificate from the CA.

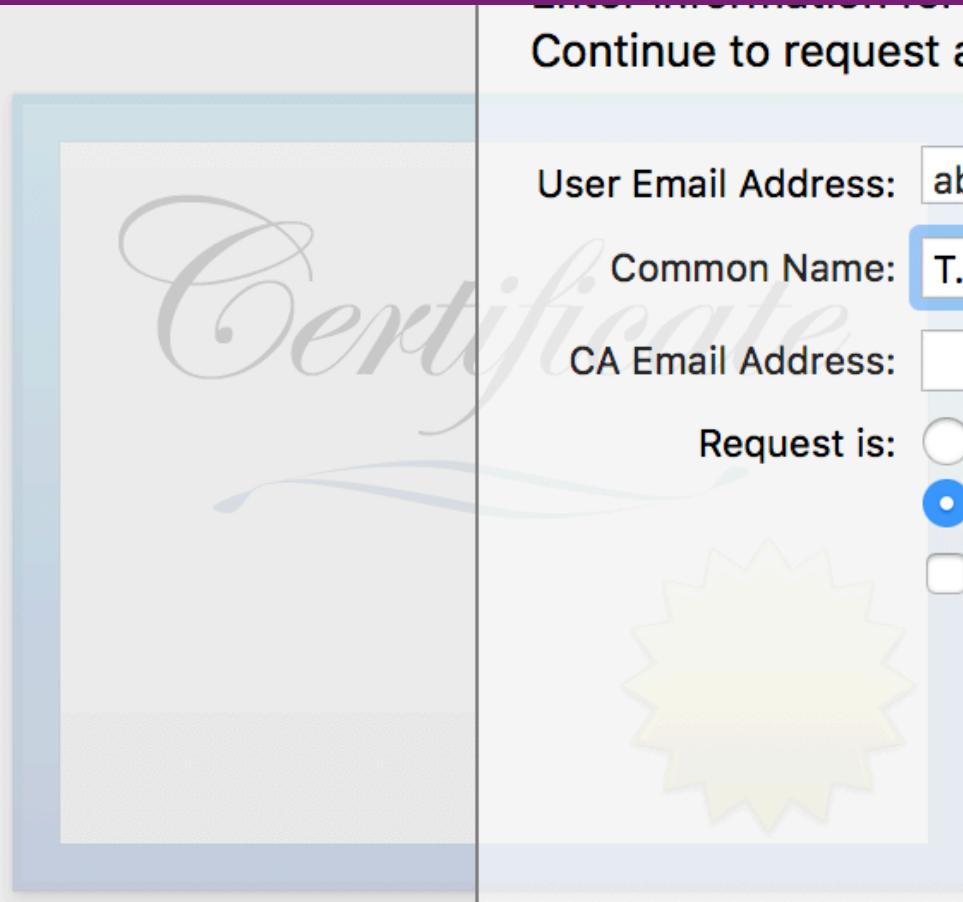
User Email Address: abinkowski@uchicago.edu

Common Name: T. Andrew Binkowski Dev Key

CA Email Address:

Request is:

- Emailed to the CA
- Saved to disk
- Let me specify key pair information



FIREBASE CLOUD MESSAGING

- Pending
- Development
- Production

Keys

- All

Identifiers

- App IDs
- Pass Type IDs
- iCloud Containers
- App Groups

Devices

- All
- Apple TV
- Apple Watch
- iPad



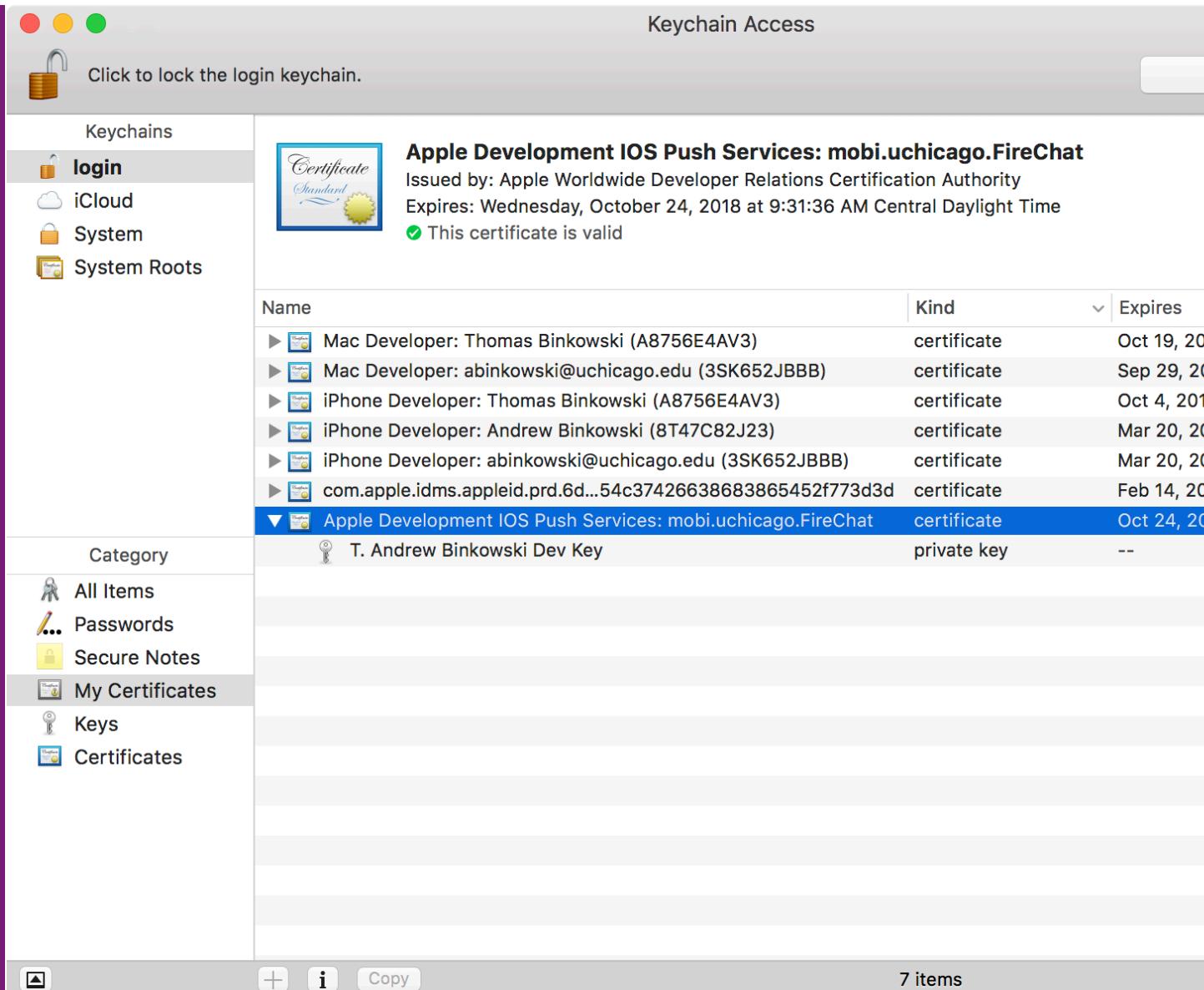
Generate your certificate.

When your CSR file is created, a public and private key pair is automatically generated. Your private key is stored on your computer. On a Mac, it is stored in the login Keychain by default and can be viewed in the Keychain Access app under the "Keys" category. Your requested certificate is the public half of your key pair.

Upload CSR file.
Select .certSigningRequest file saved on your Mac.

FIREBASE CLOUD MESSAGING

- Certificate will live in Keychain, you have to export it as .p12 files



SET UP FIREBASE TO NOTIFICATIONS

FIREBASE CLOUD MESSAGING

```
# Uncomment the next line to define a global platform for your project
# platform :ios, '9.0'

target 'FireChatAPNS' do
  # Comment the next line if you're not using Swift and don't want to use dynamic
  # frameworks
  use_frameworks!

  # Pods for FireChatAPNS
pod 'Firebase/Core'
pod 'Firebase/Messaging'
end
```

FIREBASE CLOUD MESSAGING

Storage

Hosting

Functions

Test Lab

Crash Reporting

Performance

GROW

Notifications

Remote Config

Dynamic Links

EARN

iOS apps



mobi.uchicago.firechat

Not configured for Cloud ...

Firebase Cloud Messaging can use either an APNs authentication key or APNs certificate to connect with APNs

APNs Authentication Key



Configuration with auth keys is recommended as they are the more current method for sending notifications to iOS

File

Key ID

App ID
prefix

No APNs auth key

UPLOAD

FIREBASE CLOUD MESSAGING

 Database

 Storage

 Hosting

 Functions

 Test Lab

 Crash Reporting

 Performance

GROW

 Notifications

 Remote Config

 Dynamic Links



Manage notification campaigns and send messages to engage the right users at the right moment

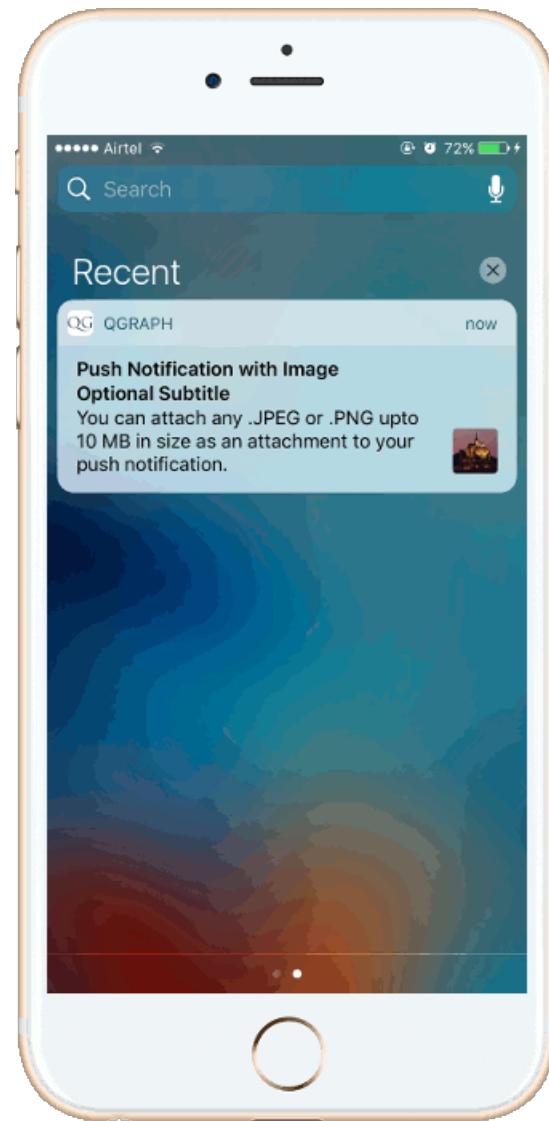
 [Learn more](#)  [View the docs](#)

 SEND YOUR FIRST MESSAGE

HANDLE NOTIFICATIONS

HANDLE NOTIFICATIONS

- There is a lot of boilerplate code to handle all the different notification code paths
- Nothing to do with Firebase



HANDLE NOTIFICATIONS

```
func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions: [UIApplicationLaunchOptionsKey: Any]?) -> Bool {

    // For iOS 10 display notification (sent via APNS)
    if #available(iOS 10.0, *) {
        UNUserNotificationCenter.current().delegate = self

        // Set up types of notifications preferences for users
        let authOptions: UNAuthorizationOptions = [.alert, .badge, .sound]
        UNUserNotificationCenter.current().requestAuthorization(
            options: authOptions,
            completionHandler: {_, _ in })
    } else {
        let settings: UIUserNotificationSettings = UIUserNotificationSettings(types: [.alert, .badge, .sound],
            categories: nil)
        application.registerUserNotificationSettings(settings)
    }

    // Register with APNS
    application.registerForRemoteNotifications()

    return true
}
```

REGISTER A
DEVICE WITH
APNS

HANDLE NOTIFICATIONS

- Notifications handled differently in different states

```
//  
// MARK: - Handle the Notifications  
  
/// If you are receiving a notification message while your app is in the background,  
/// this callback will not be fired till the user taps on the notification launching  
/// the application.  
func application(_ application: UIApplication, didReceiveRemoteNotification userInfo: [AnyHashable: Any]) {  
  
    print(userInfo)  
}  
  
/// If you are receiving a notification message while your app is in the background,  
/// this callback will not be fired till the user taps on the notification launching the  
/// application.  
func application(_ application: UIApplication,  
                 didReceiveRemoteNotification userInfo: [AnyHashable: Any],  
                 fetchCompletionHandler completionHandler: @escaping (UIBackgroundFetchResult) -> Void) {  
    print(userInfo)  
  
    // This is required to let iOS know you are down handling the notification (not much time)  
    completionHandler(UIBackgroundFetchResult.newData)  
}  
  
/// Callback from APNS that will be save in Firebase  
func application(_ application: UIApplication,  
                 didRegisterForRemoteNotificationsWithDeviceToken deviceToken: Data) {  
    Messaging.messaging().apnsToken = deviceToken  
}
```

HANDLE NOTIFICATIONS

- Notifications handled differently in different states

```
//  
// MARK: - User Notification Extensions  
  
  
@available(iOS 10, *)  
extension AppDelegate : UNUserNotificationCenterDelegate {  
  
    /// Receive displayed notifications for iOS 10 devices if the device is in the  
    /// foreground  
    func userNotificationCenter(_ center: UNUserNotificationCenter,  
                               willPresent notification: UNNotification,  
                               withCompletionHandler completionHandler: @escaping (UNNotificationPresentationOptions) -> Void) {  
        let userInfo = notification.request.content.userInfo  
  
        // Print full message.  
        print(userInfo)  
  
        // Change this to your preferred presentation option  
        completionHandler([.alert])  
    }  
  
    ///  
    func userNotificationCenter(_ center: UNUserNotificationCenter,  
                               didReceive response: UNNotificationResponse,  
                               withCompletionHandler completionHandler: @escaping () -> Void) {  
        let userInfo = response.notification.request.content.userInfo  
        // Print full message.  
        print(userInfo)  
  
        completionHandler()  
    }  
}
```

SEND FIREBASE NOTIFICATIONS

SEND FIREBASE NOTIFICATIONS

- Send from console
- Different options for sending

The screenshot shows the Firebase Notifications dashboard under the 'Notifications' tab. The left sidebar lists various services: Overview, Analytics, Authentication, Database, Storage, Hosting, Functions, Test Lab, Crash Reporting, Performance, GROW, Notifications (which is selected and highlighted in blue), Remote Config, and Dynamic Links. The main area displays a table of notifications with the following data:

Message	Status	Delivery date	Platform
This is a test with the new Firebase ...	✓ Completed	Oct 24, 2017 10:54 A...	iOS
This is pretty cool.	✓ Completed	Oct 24, 2017 10:52 A...	iOS
teste	✓ Completed	Oct 24, 2017 10:51 A...	iOS
9	✓ Completed	Oct 24, 2017 10:50 A...	iOS
8	✓ Completed	Oct 24, 2017 10:50 A...	iOS
6	✓ Completed	Oct 24, 2017 10:48 A...	iOS
6	✓ Completed	Oct 24, 2017 10:47 A...	iOS

SEND FIREBASE NOTIFICATIONS

- If you want to target specific users, you need to manage that as part of your database

Message text

Message label (optional) ?

Delivery date ?

Send Now ▾

Target

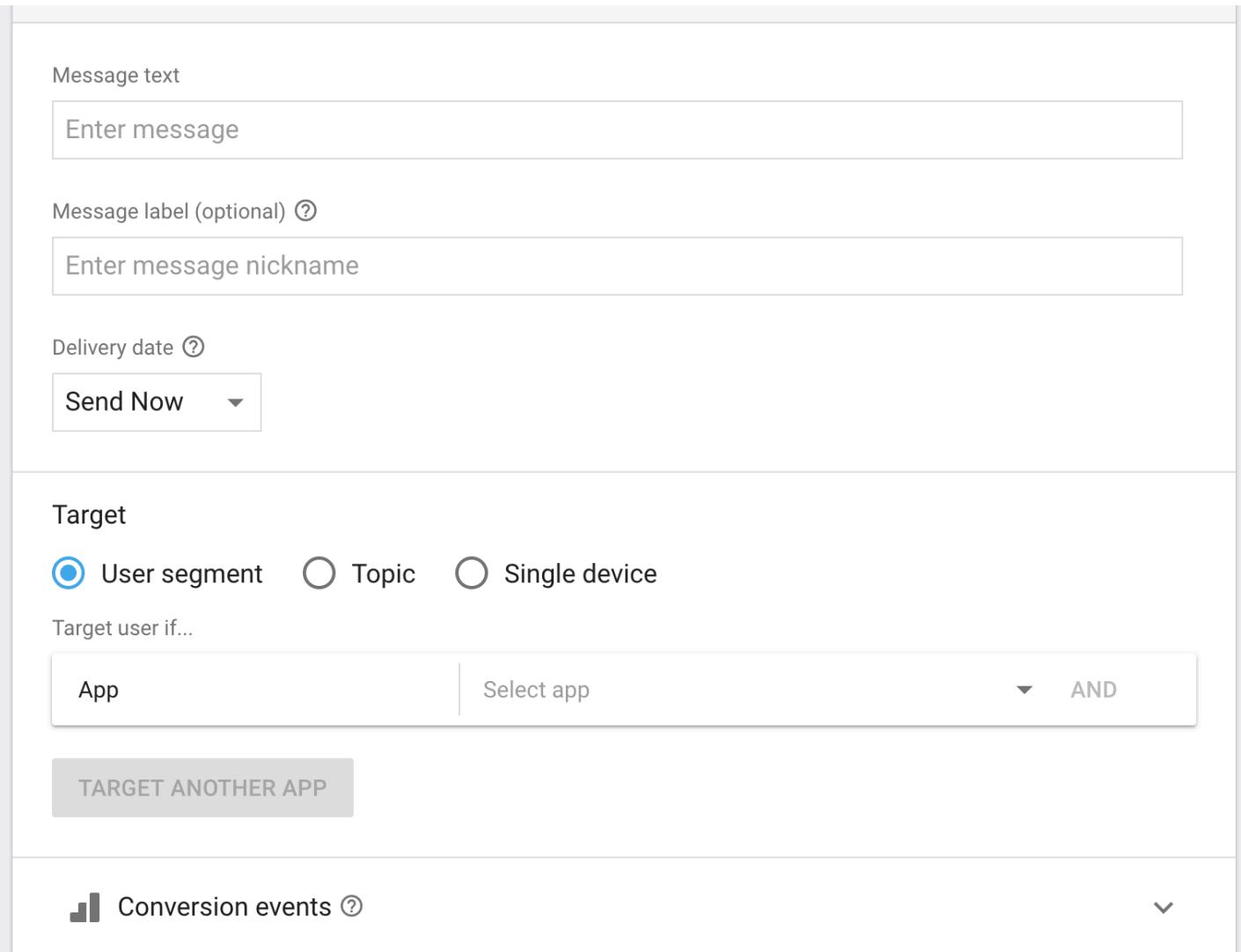
User segment Topic Single device

Target user if...

App Select app ▾ AND

TARGET ANOTHER APP

Conversion events ? ▾



FIREBASE CLOUD MESSAGING

- <https://www.youtube.com/watch?v=A1SDBIViRtE&t=6s>
- How does it work behind the scenes

Method Swizzling

```
func swizzled_application(_ application: UIApplication,  
didRegisterForRemoteNotificationsWithDeviceToken deviceToken: Data) {  
  
    // Receive the APNs deviceToken  
    // Send it off to FCM  
    // Get back an FCM deviceToken  
}
```

Understanding Firebase Cloud Messaging on iOS - Firecasts

13,141 views

122

4

SHARE

≡+

...



Firebase

Published on Jan 9, 2017

SUBSCRIBED 89K



ASSIGNMENT 5

ASSIGNMENT 4

Assignment 4 By Invitation Only: An Exclusive Chat App

In this assignment, you will create a group chat application that takes advantage of many features available through the Firebase platform. The application will incorporate common functionality of group chat applications to allow users to send messages to each other.

Overall Application Behavior

All users will have to be authenticated with Google authentication for the application.



THE UNIVERSITY OF
CHICAGO



MPCS 51033 • AUTUMN 2017 • SESSION 5

BACKENDS FOR MOBILE APPLICATIONS