Ionic 2 - Firebase RealtimeDB & Storage

Stephan Gieffers
sg@vs-5.de

Background

- founded in 2011: RealtimeDB and Social Login
  - based on chat Cloud Service offering

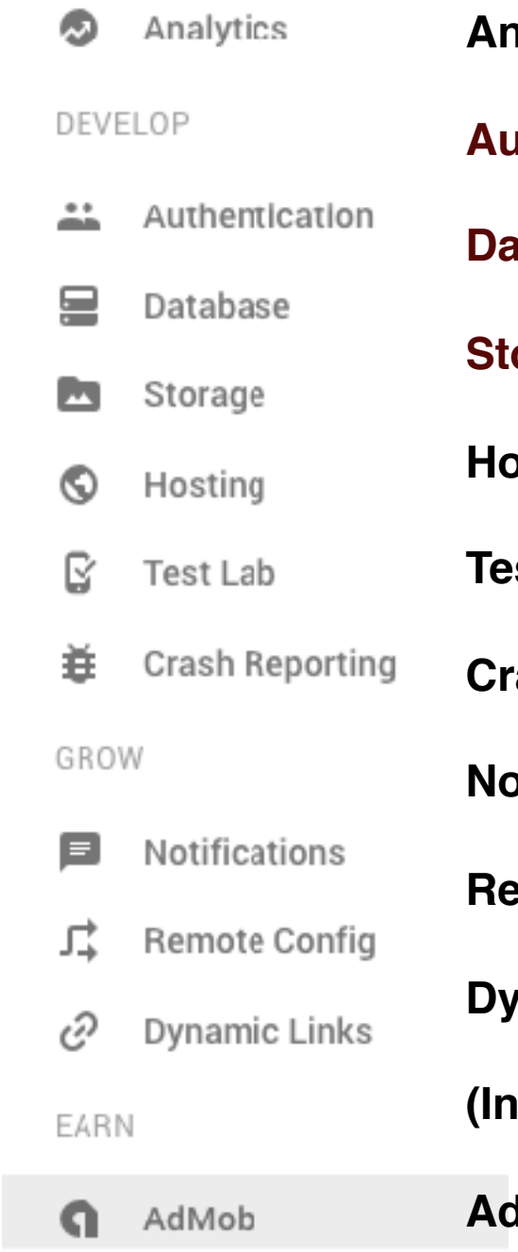
[https://en.wikipedia.org/wiki/Firebase]

- since October 2014 : acquired by Google
  [https://firebase.googleblog.com/2014/10/firebase-is-joining-google.html)]

- since May 2016 (Google I/O): „Firebase++“

  - additional Google services under„Firebase“ brand
  - strong marketing push into native Android and iOS dev camps…
    (but still first class support for JavaScript)

Firebase Features/ Services



Analytics: Google Analytics integration *(native, requires third party cordova plugin)*

Authentication: Social Login (native and web, partly requires third party cordova plugin)

Database: Realtime DB

Storage: Storage and retrieval of files

Hosting: for static webpage only - great for Angular/Ionic

TestLab: Android only - Test (native/ compiled) app on various devices in the cloud

CrashReporting: *native, requires third party cordova plugin*

Notifications: Push *(native, requires third party cordova plugin)* s

Remote Config: Distribute Configuration Parameters *(native, requires third party cordova plugin)*

Dynamic Links: Preserve web links through sign-in process *(native, requires third party cordova plugin)*

(Indexing): setup you website Index to support application links

AdMob: add Advertising to mobile app *(native, requires third party cordova plugin)*

AngularFire

- Firebase provides generic API for JavaScript (Frontend & Node)
- AngularFire is an *official* Google/ Firebase library that wraps the API in „Angular style“:
    - *only* supports Authentication and Realtime Database
    - Angular 1: https://github.com/firebase/angularfire
    - Angular 2 (beta): https://github.com/angular/angularfire2 (use GitHub URL !!!!)
- Using AngularFire for IONIC is optional!
    - Core API can be used directly
    - Both can be used at the same time

Before getting into code…

- Firebase 2 to Firebase 3 :
    - Account migration: „one click“, painless, *all „old“ APIs still working* (at least for me ,-)
    - API completely rewritten (but same concepts): easy to migrate, modernized structure, slightly confusing documentation
- Firebase/ RCs: Migration pains seem to be over
    - Forget all the „how to get firebase working with RC0“ posts: Installation, Typescript, etc….: All working as expected out of the box

Database (Realtime DB)

- Once the core of Firebase
- „Cloud based MongoDB on Websockets“:
  - JSON store with simple query, update, subscribe API
  - Rules based security

Code - finally

- setting up firebase (skim)
- using firebase as a poor man’s content mgmt system
- same but using AngularFire
    - extend to store user data…

Firebase RealtimeDB is awesome for

- storing/ updating „web content“
- persisting user settings across devices
  - including A/B testing, config parameters, debug settings, …
- realtime data exchange between users
  - likes, comments, chat, …
- rolling out without managing your own server
  - if you need some server-side action consider node.js to listen on database changes:
    - push notifications, send mail, ….

Caveats

- It is not a relational database!
    - don’t join, but manage redundancy or handle business logic
    - be prepare to roll your own: migration, managing live updates, etc….
- Understand „Rules“ before you deploy

    e.g. Google I/O: https://www.youtube.com/watch?v=PUBnlbjZFAI,
    quick start: https://www.youtube.com/watch?v=eUfSpktxNeQ

    - you are configuring every user’s access to your data - no „shielded by web-server“ layer
    - easy to mess up - follow standard patterns
- Testing can be messy:
    - no up-to-date mock available - use Spy & Stubs - keep Firebase code in providers
    - async, caching, fb running in separate process, … => expect surprises
    - however, code is very maintainable
- Unfortunately: *only partially offline enabled*
    - cache reads & writes on temporary outage (as long as you process is running)
    - no connection on start - no data (solved for Native versions; Web was announced, but than canceled)
    - for „easy“ cases you can build your own - (but beware of IonicStorage)

AngularFire or Firebase „Pure“

- AngularFire is perfect for easy stuff, but
    - you may want to do more complex preprocessing before showing / saving data
    - you will want providers to encapsulate business logic (e.g. „joins“)
    - so eventually => you probably either start using „more advanced“ AngularFire options (aka {preserveSnapshot: true }) or use „pure Firebase“
- As all blackboxes: great if works - doomed if it doesn't

Storage

- Upload and access files via a proper API

more code…..

Good use cases for Firebase Storage

- You are using Firebase anyway - using FB Storage is convenient
    - also integrated well with users of Google Cloud Storage
- If you are not issuing your own authentication tokens (to handle access to storage providers)
- If you need users to do file uploads
- You want to make files available
    - to the unauthenticated public (easy, but S3 may be simpler)
    - to all of your authenticated users (default) - but nobody else
    - only to selected users (i.e. person who uploaded, chat partner etc.): easy, add UserIDs to Storage and use generic rule
    - depending on complex setting in your realtime DB related to user:
      not supported - you will need workarounds (unless you generate your own tokens…)

IONIC challenge is not Firebase Storage but…

- **… that Mobile and Files sucks as such:**
  - Cordova plugins absolute required
    - http://ionicframework.com/docs/v2/native/file/ - File Handling
      - must read (!): github plugin doc to understand different storage locations for iOS and Android
    - http://ionicframework.com/docs/v2/native/transfer/ - File download
    - http://ionicframework.com/docs/v2/native/fileopener/ - Launch native apps
    - there are many more… because there are many challenges…
  - When using files via html: expect to spend some time on iOS/Android differences (checkout „cdvfile“ format)
  - You want to do photos?
    - handling thumbnails vs. actual image
    - integrating (many) local photos images into html may generate memory issues
  - Forget „ionic serve“ - most features only run native

Summary & Outlook & Discussion

- Firebase Realtime DB is a rock solid cloud solution (if you like NoSQL)
- If you really really need files and security, Firebase Storage is a great option
- APIs are very solid - AngularFire provides some shortcuts
- Google is pushing hard - but mostly on Native
- IONIC Announcement on Monday: http://blog.ionic.io/focusing-on-fewer-things/
  - „…and that we could be putting that engineering effort into unique Ionic services around Progressive Web Apps and **web-first realtime databases.**“