Starting a Cross-Platform Firebase App

Mike Koss
koss@google.com

GDG Seattle Dev-Fest
October 15, 2016

Goals

Using the JavaScript Firebase SDK as the foundation of a cross-platform application.

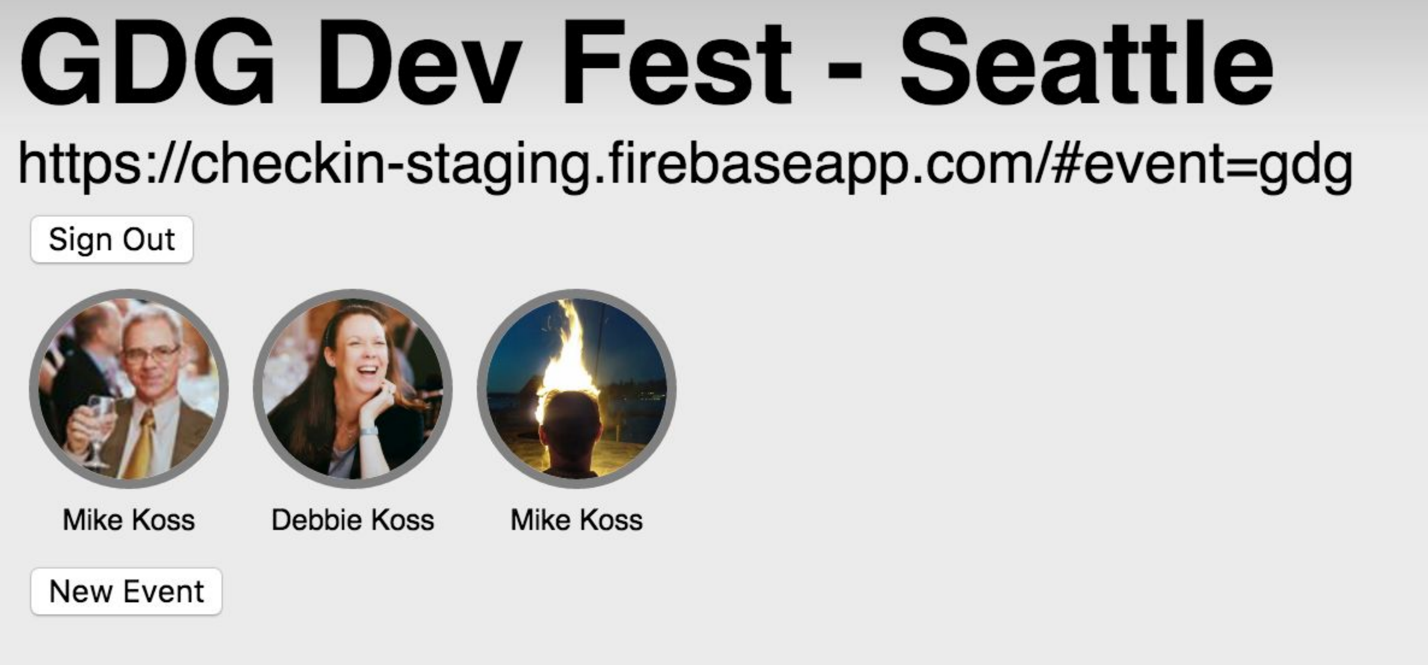
1. Server and Command Line Tools (node.js)
2. Desktop Web
3. Mobile Web
4. Packaged Mobile (Cordova or React Native)

Start with UI-less App

1. Start with the Data Model
    a. Server
    b. Client State
2. Implement UI-less methods and business logic.
3. Expose a stream of client state changes to the UI layer.
4. Can be unit-tested w/o any UI complications.
5. Pluggable into multiple UI platforms or Frameworks

Firebase Checkin - Demo?

https://checkin-staging.firebaseapp.com/#event=gdg



GDG Dev Fest - Seattle

https://checkin-staging.firebaseapp.com/#event=gdg

Walk Through...

1. Firebase Database Basics
    a. Checkin Data Viewer
2. Security Rules
3. Firebase Bolt - Security Rules Compiler
4. (brief detour about TypeScript and tool chain)
5. UI-less Application Structure
    a. Asynchronous Programming
    b. Immutable data structures.
    c. Promises and Streams
6. Unit Tests
7. Hooking up a (web) UI

Code Lab

Speaker Feedback Buttons:

+1 - Liking this talk

-1 - Not so much

Display Green or Red border around the profile image of each person sending Feedback.

Code Lab Steps

1. Install the Repo and configure tools.
2. Create a Firebase App backend (use configure-project script).
3. Update Data Model Rules to add attendee feedback property (checkin.bolt)
4. Add feeback() method to checkin.ts
5. Add +1 and -1 Buttons to UI and call feedback(1) or (-1)
6. Render Profile with unique style for each feedback mode.

Hints: https://github.com/mckoss/firebase-checkin/pull/1/files

Links

https://github.com/mckoss/firebase-checkin

Demo: https://checkin-staging.firebaseapp.com/#event=gdg