Going Serverless with **Firebase** and Fullstack **ClojureScript**
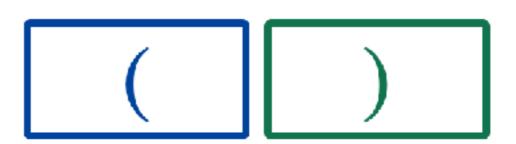
re:Clojure 2019

Agenda

1. show existing mini app

2. Firebase database

    1. **write** to Firebase

    2. **read** from to Firebase

3. **profit**

4. *create https endpoint (bonus)*

mini-app: parentheses

- JavaScript / npm deps

  - **Shadow-cljs** (all-in-one cljs build tool)

  - **Firebase** (app dev platform on Google infra)

- ClojureScript deps

  - **re-frame** (framework to build single page apps)

Write to Firebase

- realtime database

- "parens" path

- custom effect

- push

Read from Firebase

- change init function

- another custom effect

- one way data flow

Profit

- deploy to Firebase hosting

  - **https://parens-2eb63.web.app/**

- pros

  - the sky is the limit

  - free tier

- cons

  - ok, vendor lock-in

Operation	Limit	Description
Simultaneous connections	200,000	A simultaneous connection is equivalent to one mobile device, or server app connected to the database. This isn't the same as the total number of users of your app, bec users don't all connect at once. For example, apps with 10 milli active users usually have fewer than 200,000 simultaneous con Your max simultaneous connections depends on your total user the average time users spend in your app. However, if you need to scale beyond this limit, try using multipl
Simultaneous responses sent from a single	~100,000/second	Responses include simultaneous broadcast and read operation server from a single database at a given time. The limit refers t packets that represent each individual read or broadcast operat push notifications, sent from the database. This guidance is for

bonus: endpoint!

- Firebase Cloud Functions

- Https triggers

- also scale to millions

- https://github.com/multis/parens

```
[
    ")",
    "(",
    ")",
    ")",
    "(",
    ")",
    "(",
    ")",
    "(",
    ")",
    "(",
    ")",
    "(",
    ")",
    "\"
```

Thank you for listening

- i'm Théo / @teawaterwire

- co-founder @ Multis (YC S19)

  - crypto-first business banking

  - fullstack ClojureScript

  - join@multis.co ;)