

Are you prepared for PSD2?

PSD2 is changing how payments are managed in the European market.

Let's find out how much you know about this directive and how we can help your business prepare for the changes it brings.



July 2019

RangeError

Maximum call stack size exceeded.

JS

app:///main.ios.bundle in ga at line 11:496869

app:///main.ios.bundle in n at line 11:456494

app:///main.ios.bundle in v at line 11:17137

app:///main.ios.bundle at line 11:499897

app:///main.ios.bundle in Nf at line 11:29188

app:///main.ios.bundle in ga at line 11:496869

app:///main.ios.bundle in n at line 11:456494

app:///main.ios.bundle in v at line 11:17137

Mysterious errors

“To be honest the app doesn’t work on our devices. It’s so slow and unresponsive”

“I can no longer use my app because it is now so incredibly slow”



Performance regressions

Guideline 5.1.1 - Legal - Privacy - Data Collection and Storage

We noticed that your app requires users to register or log in to access features that are not account-based.

Specifically, your app requires users to register and log in with personal information before they can browse the menu.

Next Steps

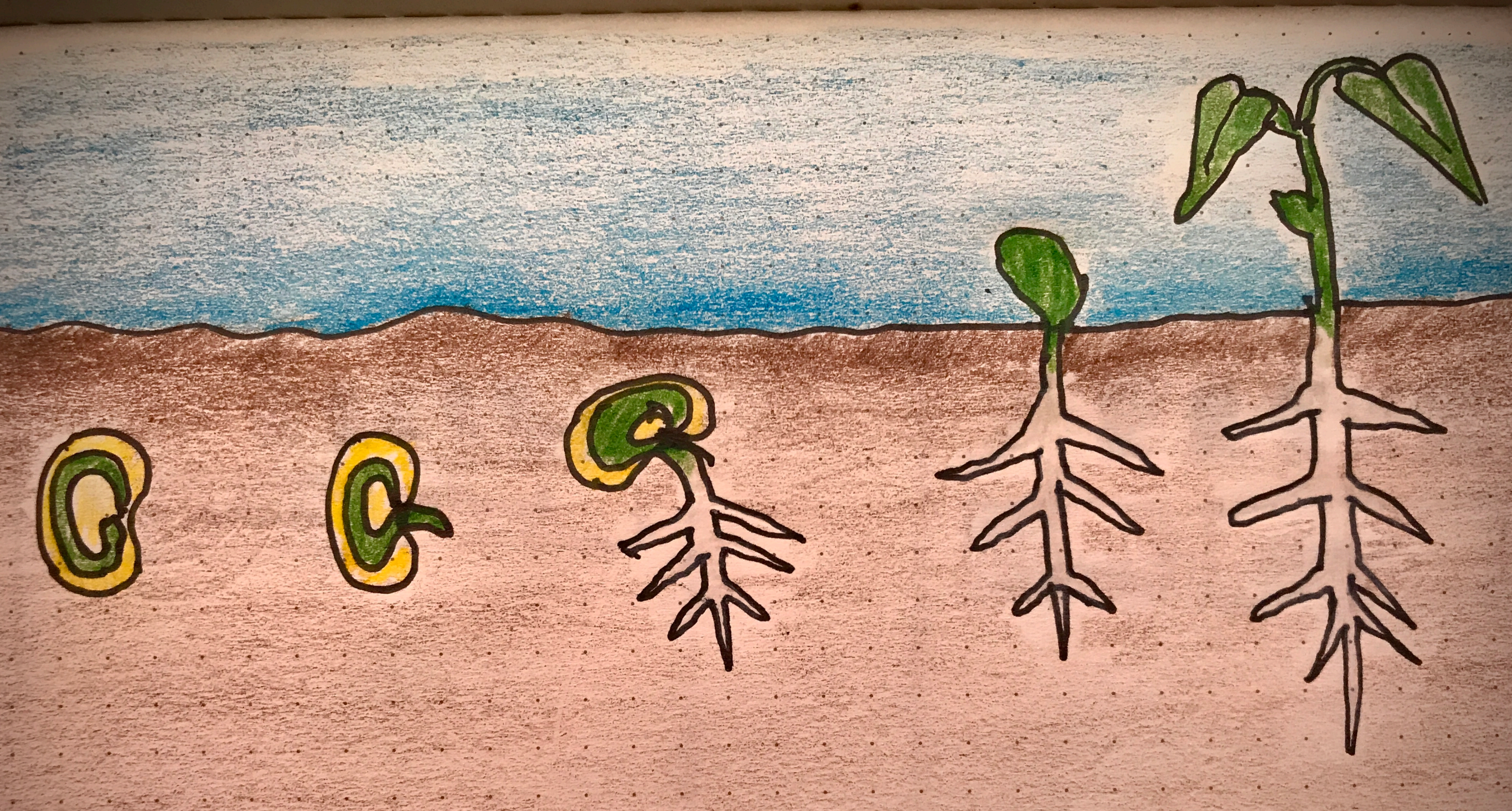
To resolve this issue, please revise your app to let users freely access your app's non-account-based features.

Apps may not require users to enter personal information to function, except when directly relevant to the core functionality of the app or required by law.

You should allow users to freely access your app's non-account-based features. For example, an e-commerce app should let users browse store offerings and other features that are not account-based before being asked to register, or a restaurant app should allow users to explore the menu before placing an order. Registration must then only be required for account-specific features, such as saving items for future reference or placing an order.

REJECTED

A surprise from Apple



Growing mobile apps with
ClojureScript and React Native



The starting point



Roja Buck 15/11 18:22

Thought the conference was cancelled?

Anyway... I do indeed have an anecdote 😊

15/11 18:24

It was my first day at riverford. I sat down next to the MD and after about 10 minutes sorting out my email and such things he turned to me and said "it's great to have you here, I know you probably have a lot of ideas about where to get started but just in case I wrote you a list of things" the list had 60 things on it. the very first of which was "an easy order app." Now we just have the 59 other things to build...



1

“An easy order app”

Sign-in 1

High fidelity sign in flows, initial onboarding timed image overlays, video, placeholder forgot password screen, error messages. Logging in. Busy indicator. Persistence of token to SecureStorage.

User stories: APP-2, APP-18

Next delivery 1

Next delivery screen with product thumbnails, prices, quantities, date, total costs. Placeholder buttons. Blank state. Loading state while fetching next delivery.

User stories: APP-12

Menu 1

Top level menu with dropdown animation and top level category images (no action yet). Error handling if category fetch fails. Logout.

User stories: APP-16

Search 1

Search overlay, product search results appear as you type. Keyboard throttling. Error state. Empty state when nothing typed. No results state. Pop-up and 'Close' animation.

User stories: APP-8, (APP-26)

Product 1

Basic product screen. Image with overlay hero text. Description. Placeholder buttons. Stack and 'Back' navigation. Scroll animation

API dependencies: Cleaned up product details

Add widget 1

How many this week and regularly summary. Set up regular and one off deliveries for an item. Update local state. Wire into product and next delivery screen.

User stories: APP-10, APP-11, APP-13, APP-14, APP-15

Sync 1

Red/Orange/Green Sync status bar. Last change wins syncing. Persistence of order in local storage.

User stories: APP-21

Analytics/error reporting 1

Basic recording of ui events and submission of error logs and breadcrumbs to sentry. Collection of some basic analytics.

Testing 1

Manual test scripts. Preparation of test builds for alpha tests.

Product 2

Pulling content from CMS. Quote. Influencers. Seasonality. Nutrition. Recipes for recipe boxes. Grower info.

User stories: APP-20

API dependencies: whichever product details are required - e.g. nutrition, seasonality

Category 1

Category screen with header image and list view contents. Wire into menu. Loading state and error states. Modular recipe boxes. Offline catalog?

User stories: APP-5, (APP-26)

API dependencies: Categories

Next delivery 2 - errors and warnings

Notification if delivery will not occur due to payment issues.

Min spend. Delivery day exceptions. Discounts.

User stories: APP-17, APP-24, APP-25

iPhone X compatability

iPhone x has a larger screen, but portions of it are not allowed to contain controls. The user interface needs to adapt to these requirements. Apple will not allow releases if they are not compatible with iPhone X

Recommendations 1

"Have you forgotten" recommendations based on local state (things previously ordered but not in this order). Slider with product links.

API dependencies: recommendations

Menu 2

Other menu items and pages, Franchisee info (nice to have)

Notification settings. Privacy/GDPR settings. Send logs.

User Stories: APP-28, APP-30

Box Contents 1

Box contents slider for all box items in next delivery and product screen. Products do not link.

User stories: APP-12

Sign in 2

Forgot password and magic link. Improved (longer) expiry handling. Pulling password from keychain. Screen transitions.

User Stories: APP-3, APP-4, APP-23

API dependencies: refresh tokens and expiry

Expo development tools

All of our products and downloads



Client



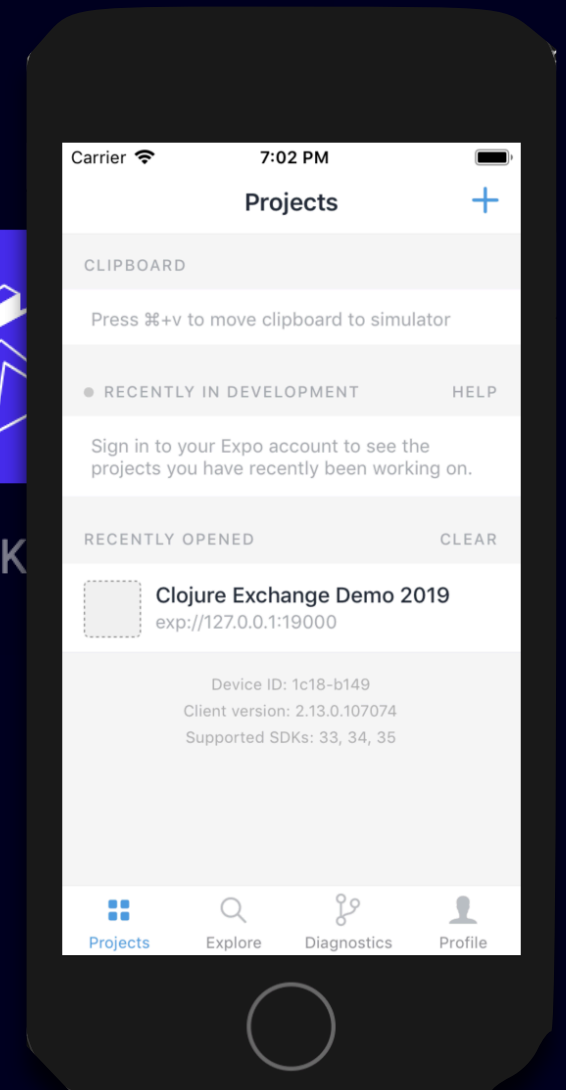
Expo CLI



Snack



SDK



Tip #1: try Expo

1. Stable platform
2. Easily try on device
3. Code push
4. Don't need a mac!
5. Easy CI integration

Expo advantages

```
~/development ▶ yarn global add expo-cli  
success Installed "expo-cli@3.4.1" with binaries:  
  - expo  
  - expo-cli  
✨ Done in 13.89s.
```

```
~/development ▶ expo-cli init
? Choose a template: (Use arrow keys)
  ----- Managed workflow -----
> blank                                a minimal app as clean
as an empty canvas
```


? Choose a template: expo-template-blank
? Please enter a few initial configuration values.

Read more: <https://docs.expo.io/versions/latest/workflow/configuration/> > 0% completed

```
{
  "expo": {
    "name": "<The name of your app visible on  
the home screen>",
    "slug": "<A URL friendly name for your app>"
  }
}
```

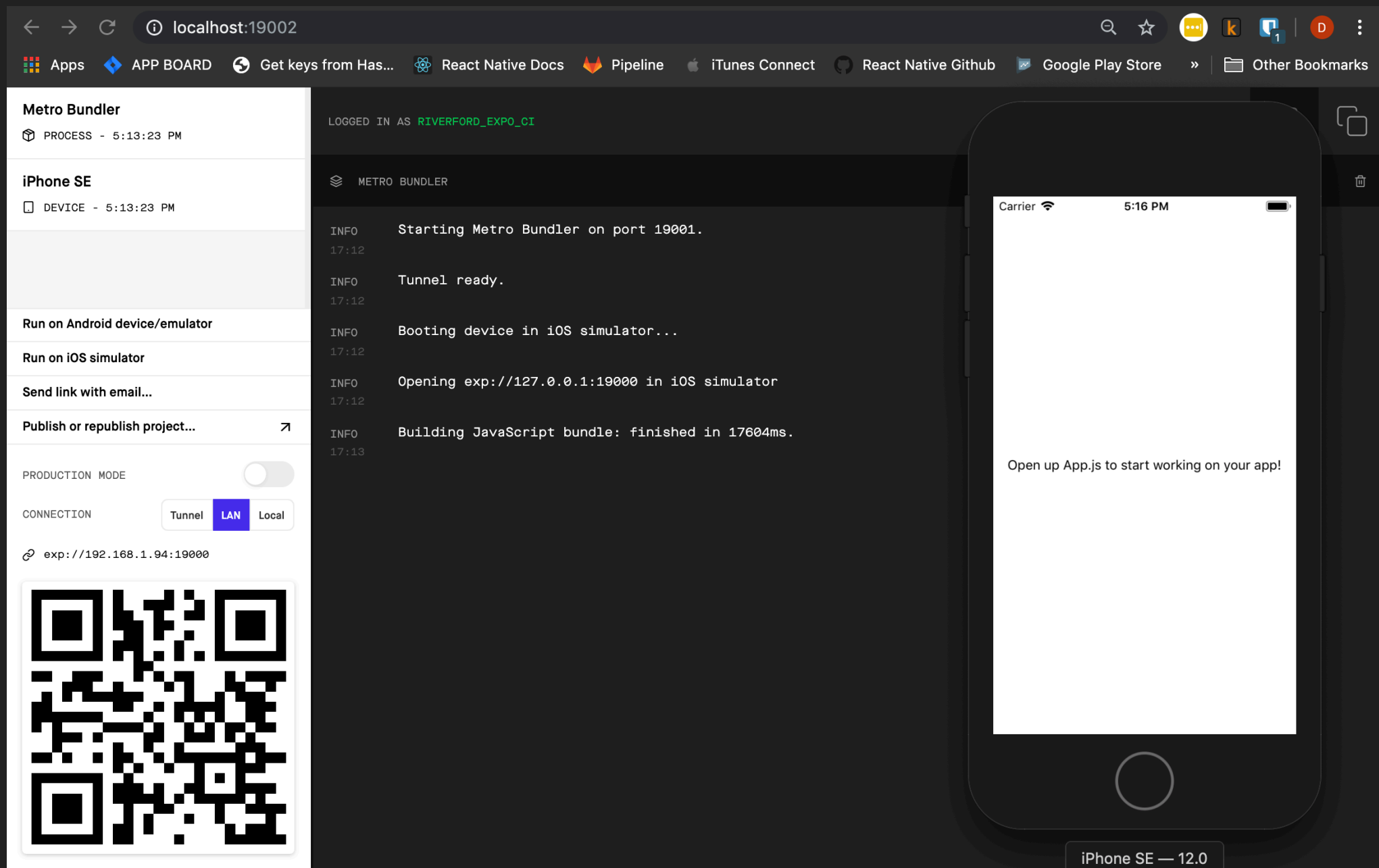
```
/Users/danielneal/deve ${
.
..
.expo-shared
.git
.gitignore
App.js
app.json
assets
babel.config.js
node_modules
package.json
yarn.lock

"expo": {
  "name": "Clojure Exchange Demo 2019",
  "slug": "clojure-exchange-demo-2019",
  "privacy": "public",
  "sdkVersion": "35.0.0",
  "platforms": [
    "ios",
    "android",
    "web"
  ],
  "version": "1.0.0",
  "orientation": "portrait",
  "icon": "./assets/icon.png",
  "splash": {
    "image": "./assets/splash.png",
    "resizeMode": "contain",
    "backgroundColor": "#ffffff"
  },
  "updates": {
    "fallbackToCacheTimeout": 0
  },
  "assetBundlePatterns": [
    "**/*"
  ],
  "ios": {
    "supportsTablet": true
  }
}
```

Output

```
~/development/clojure-exchange-demo-2019 (master  
x)★ ▷ expo-cli start --lan  
Starting project at /Users/danielneal/  
development/clojure-exchange-demo-2019  
Expo DevTools is running at http://localhost:  
19002  
Opening DevTools in the browser... (press shift-  
d to disable)  
Starting Metro Bundler on port 19001.
```

Kick-off



The expo devtools and client



Tip #2: Try shadow-cljs

1. Smooth hot reload
2. Better externs support
3. Easy to use npm libs
4. Robust and clear config
5. ***NEW*** shadow inspect

Shadow cljs advantages

```
{:source-paths ["src"]
 :dependencies [[lilactown/hx "0.5.3"]]
 :builds
 {:app
  {:target :react-native
   :init-fn clojure-exchange-demo-2019.entry-point/init
   :output-dir "target"
   :compiler-options {:infer-externs :auto}
   :devtools {:autoload true
              :preloads [shadow.remote.runtime.cljs.browser]}}}}
```

Setting up shadow (1/3)
Add shadow-cljs.edn

```
5 files changed, 38 insertions(+), 20 deletions(-)
modified   .gitignore
deleted    App.js
modified   app.json
@@ -2,6 +2,7 @@
    "expo": {
      "name": "Clojure Exchange Demo 2019",
      "slug": "clojure-exchange-demo-2019",
+     "entryPoint": "./target/index.js",
      "privacy": "public",
      "sdkVersion": "35.0.0",
      "platforms": [
new file   shadow-cljs.edn
new file   src/clojure_exchange_demo_2019/entry_point.cljs
```

Setting up shadow (2/3)

Edit app.json


```
(ns clojure-exchange-demo-2019.entry-point
  (:require
    [shadow.expo :as shadow-expo]
    [hx.react :as hx :refer [defnc]]
    ["react-native" :as rn]))

(defnc App
  []
  [rn/View {:style {:backgroundColor "purple"
                    :alignItems "center"
                    :flexGrow 1
                    :justifyContent "center"}}
   [rn/Text {:style {:color "white"
                     :fontSize 20}}
    "Hello from shadow-cljs"])]

(defn start
  "Entry point for ui, called every hot reload"
  {:dev/after-load true}
  []
  (shadow-expo/render-root (hx/f [App])))

(defn init
  "Initialization function, called once at app start up"
  []
  (start))
```

Hello from shadow-cljs

Setting up shadow cljs
3/3 add the entry point

Design Systems **break as they scale** (either scaling org or scaling product) because new components/variants of a component are introduced. Those variants sometimes (read: often) go undocumented, leading to duplication when that component/variant is needed (and created) again. Even when the component is documented, documenting effectively often means dozens/hundreds of instances to capture all states/variants. Systems like Tachyons et al. approach this problem by instead documenting and limiting **properties** of components. (I like to think of this as “subatomic” design.)

Daniel Eden

Designer at Facebook

Tip #3: Try tachyons

```
(defnc App
  []
  [rn/View {:style {:backgroundColor "#551199"
                    :alignItems "center"
                    :flexGrow 1
                    :justifyContent "center"}}
    [rn/Text {:style {:color "white"
                      :fontSize 20}}
      "Hello Clojure Exchange 2019!"]])

(defnc App2
  []
  [rn/View {:style (s [:bg-brand0 :aic :fg1 :jcc])}
    [rn/Text {:style (s [:ui1 :f4])}
      "Hello Clojure Exchange 2019!"]])
```

What Tachyons looks like

1. Less code
2. Improved consistency
3. Easier modification

Tachyons advantages

Commit 3b28734e authored 20 hours ago by Daniel

Browse files

Options ▾

Replaces the styling on the next delivery screen

parent 1d63330a Pmaster

No related merge requests found

✓ Pipeline #11075 passed with stages ✓ in 20 minutes 48 seconds

Changes 2 Pipelines 1

Showing 2 changed files ▾ with 7 additions and 7 deletions

Expand all

Hide whitespace changes

Inline

Side-by-side

src/riverford/mobile_app_orders/ui/next_delivery.cljs

View file @ 3b28734e

```
... @@ -15,144 +15,17 @@
15 (def expo (js/require "expo"))
16 (def version-number (oget expo "Constants.manifest.version"))
17
18 - (def styles
19 -   (ui/stylesheet
20 -    {:next-delivery.notifications.view {:marginHorizontal 8
21 -                                         :marginVertical 4
22 -                                         :elevation 4
23 -                                         :shadowRadius 4
24 -                                         :shadowOffset {:height 0
25 -                                                         :width 0}
26 -                                         :shadowColor (ui/color :dark-primary)
27 -                                         :borderRadius 4
28 -                                         :shadowOpacity 0.25
29 -                                         :padding 8
30 -                                         :backgroundColor (ui/color :light-sec
31 -                                                             ondary)}}
32 -    :next-delivery.notifications.actions.view {:flexDirection "row"
33 -                                                 :justifyContent "space-between"
34 -                                                 :alignSelf "stretch"
```

```
... @@ -15,144 +15,17 @@
15 (def expo (js/require "expo"))
16 (def version-number (oget expo "Constants.manifest.version"))
17
```



Simultaneous development of
App and API

In growing anything a point is
reached where the part that
is growing hits a wall

Are you prepared for PSD2?

PSD2 is changing how payments are managed in the European market.

Let's find out how much you know about this directive and how we can help your business prepare for the changes it brings.



RangeError

Maximum call stack size exceeded.

```
JS
app:///main.ios.bundle in ga at line 11:496869
app:///main.ios.bundle in n at line 11:456494
app:///main.ios.bundle in v at line 11:17137
app:///main.ios.bundle at line 11:499897
app:///main.ios.bundle in Nf at line 11:29188
app:///main.ios.bundle in ga at line 11:496869
app:///main.ios.bundle in n at line 11:456494
app:///main.ios.bundle in v at line 11:17137
```

Customer feedback

"To be honest the app doesn't work on our devices. It's so slow and unresponsive"

"I can no longer use my app because it is now so incredibly slow"



Guideline 5.1.1 - Legal - Privacy - Data Collection and Storage

We noticed that your app requires users to register or log in to access features that are not account-based.

Specifically, your app requires users to register and log in with personal information before they can browse the menu.

Next Steps

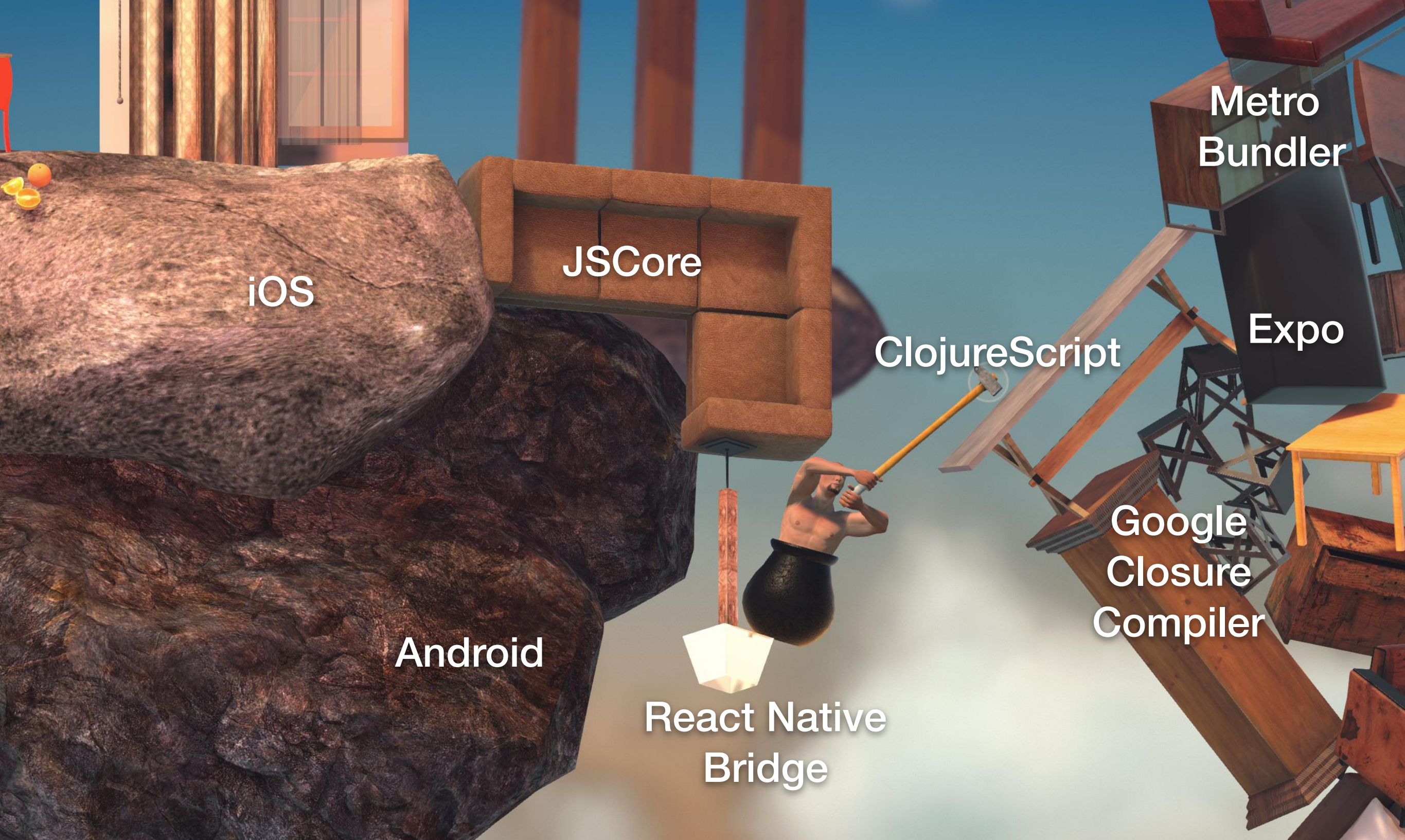
To resolve this issue, please revise your app to let users freely access your app's non-account-based features.

Apps may not require users to enter personal information to function, except when directly relevant to the core functionality of the app or required by law.

You should allow users to freely access your app's non-account-based features. For example, an e-commerce app should let users browse store offerings and other features that are not account-based before being asked to register, or a restaurant app should allow users to explore the menu before placing an order. Registration must then only be required for account-specific features, such as saving items for future reference or placing an order.

REJECTED

July 2019



Question the stack



how to merge two source maps



All

Images

Videos

Maps

News

More

Settings

Tools

About 231,000,000 results (0.81 seconds)

Combine source maps of two compilation steps - Stack Overflow

[https://stackoverflow.com](https://stackoverflow.com/questions/combine-source-maps-of-two-com...) › [questions](#) › [combine-source-maps-of-two-com...](#) ▼

4 answers

Feb 12, 2014 - Try sorcery - it's designed for exactly this purpose (I'm the author, I came here looking for info on related tools). As long as the `.map` files are in ...

How can **sourcemaps** be merged

1 answer Mar 22, 2014

Combine javascript files and their **source maps**

1 answer Aug 8, 2015

Preserving **sourcemaps** when **merging** JS streams (to ...

1 answer Jul 3, 2016

Need to **merge two source** objects into target object ...

1 answer Apr 18, 2019

[More results from stackoverflow.com](#)

jvilk/merge-source-maps: Library, command-line tool ... - GitHub

[https://github.com](https://github.com/jvilk/merge-source-maps) › [jvilk](#) › [merge-source-maps](#) ▼

Source Files **merge-source-maps** will follow the chain of **source maps** to the beginning. For example, let's say that you compiled `foo.ts` to `foo.js`, and minified it to `foo.min.js`. If you want to **combine** `foo.js.map` and `foo.min.js.map`, you should specify `foo.min.js`.

jakobwesthoff/source-map-merger: Merge generated ... - GitHub

[https://github.com](https://github.com/jakobwesthoff/source-map-merger) › [jakobwesthoff](#) › [source-map-merger](#) ▼

Merge generated **Source Maps** to create a map between the initial source and ... **source-map-**

HELP: SEE UNDER GUIDANCE



A rudimentary, but useful tool

The screenshot shows a web browser window with the address bar displaying `github.com/riverford/sourcemap-cljs`. The browser's bookmark bar contains several items: 'Apps', 'APP BOARD', 'Get keys from Has...', 'React Native Docs', 'Pipeline', 'iTunes Connect', and 'React Native Github'. The page title is 'Typical process (with stacktrace)'. The main content area contains the following text:

Follow step 1 as above, but then copy the stacktrace string into a file. Stacktrace will be parsed for `<line>:<column>` pairs.

```
app:///main.ios.bundle at line 11:935705
app:///main.ios.bundle in Mf at line 11:28521
app:///main.ios.bundle in D at line 11:31231
app:///main.ios.bundle in e at line 11:61222
app:///main.ios.bundle in u at line 11:61158
app:///main.ios.bundle in vw at line 11:149747
...
```

1. `pbpaste > stacktrace.txt`
2. `node sourcemap.js --js ~/Downloads/main.ios.map --cljs ~/Downloads/index.js.map --stacktrace stacktrace.txt`

Which should give an output like the following - which is not pretty, but hopefully enough to avoid despair:

```
11:935705
{:source riverford/mobile_app_orders/util.cljs, :line 222, :column 80, :name s}
11:28521
{:source cljs/core.cljs, :line 3901, :column 18, :name a0}
11:31231
{:source cljs/core.cljs, :line 3931, :column 29, :name args}
11:61222
{:source cljs/core.cljs, :line 10789, :column 29, :name args}
```

Available on github:
Riverford/sourcemap-cljs

Digital Digressions by Stuart Sierra

From programming to everything else

[About Me](#)[Writing](#)[Presentations](#)[Software](#)[Contact](#)

Clojure Don'ts: Concat

Posted on April 26, 2015 by Stuart ~~Sierra~~

Welcome to what I hope will be an ongoing series of Clojure *do's* and *don'ts*. I want to demonstrate not just good patterns to use, but also anti-patterns to avoid.

Some of these will be personal preferences, others will be warnings from hard-won experience. I'll try to indicate which is which.

First up: concat

A recommended series

Are you prepared for PSD2?

PSD2 is changing how payments are managed in the European market.

Let's find out how much you know about this directive and how we can help your business prepare for the changes it brings.



RangeError

Maximum call stack size exceeded.

```
JS
app:///main.ios.bundle in ga at line 11:496869
app:///main.ios.bundle in n at line 11:456494
app:///main.ios.bundle in v at line 11:17137
app:///main.ios.bundle at line 11:499897
app:///main.ios.bundle in Nf at line 11:29188
app:///main.ios.bundle in ga at line 11:496869
app:///main.ios.bundle in n at line 11:456494
app:///main.ios.bundle in v at line 11:17137
```

Customer feedback

"To be honest the app doesn't work on our devices. It's so slow and unresponsive"

"I can no longer use my app because it is now so incredibly slow"



Guideline 5.1.1 - Legal - Privacy - Data Collection and Storage

We noticed that your app requires users to register or log in to access features that are not account-based.

Specifically, your app requires users to register and log in with personal information before they can browse the menu.

Next Steps

To resolve this issue, please revise your app to let users freely access your app's non-account-based features.

Apps may not require users to enter personal information to function, except when directly relevant to the core functionality of the app or required by law.

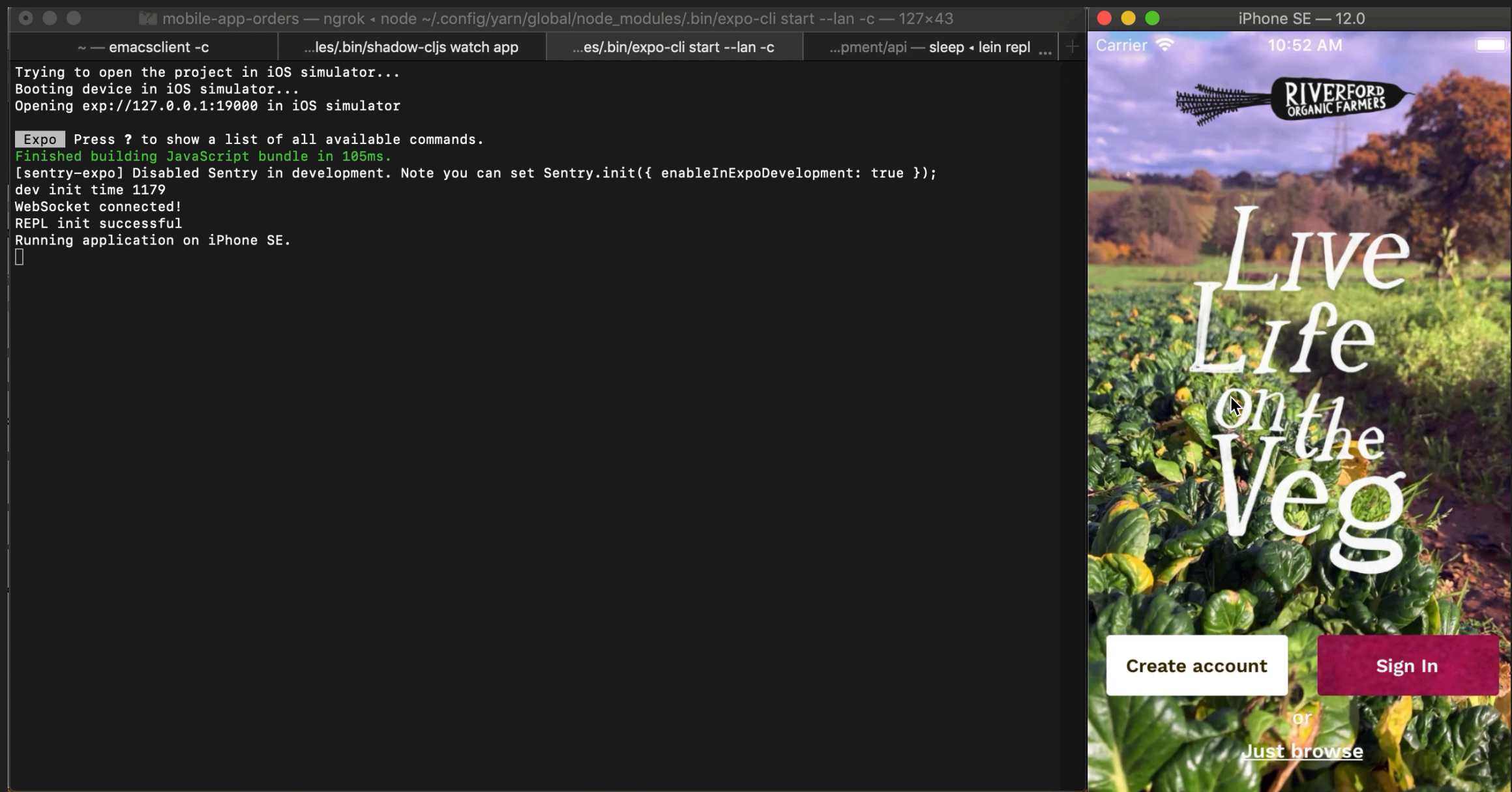
You should allow users to freely access your app's non-account-based features. For example, an e-commerce app should let users browse store offerings and other features that are not account-based before being asked to register, or a restaurant app should allow users to explore the menu before placing an order. Registration must then only be required for account-specific features, such as saving items for future reference or placing an order.

REJECTED

September 2019

1. Using the android systrace tool
2. Spying on the bridge
3. Using the chrome profiler
4. Rudimentary hand tools

4 ways with performance



Using the chrome profiler

```
(def debug (atom []))

(defn regular-items
  "A block for regular items and pre-orders"
  []
  (let [items @(subscribe [:next-delivery/regular-items])
        pre-orders @(subscribe [:next-delivery/pre-orders])
        slot @(subscribe [:next-delivery/next-editable-slot])
        paused? @(subscribe [:next-delivery/paused? slot])]
    (println "Re-rendering")
    (swap! debug conj {:timestamp (js/Date.)
                       :items items})))
```

A rudimentary, but useful tool

```
(extend-type Decimal
  IEquiv
  (-equiv [x o]
    (and (= (type x) Decimal)
          (= (type o) Decimal)
          (ocall x "equals" o))))
```

Fixing the custom Decimal

Are you prepared for PSD2?

PSD2 is changing how payments are managed in the European market.

Let's find out how much you know about this directive and how we can help your business prepare for the changes it brings.



Customer feedback

"To be honest the app doesn't work on our devices. It's so slow and unresponsive"

"I can no longer use my app because it is now so incredibly slow"



Guideline 5.1.1 - Legal - Privacy - Data Collection and Storage

We noticed that your app requires users to register or log in to access features that are not account-based.

Specifically, your app requires users to register and log in with personal information before they can browse the menu.

Next Steps

To resolve this issue, please revise your app to let users freely access your app's non-account-based features.

Apps may not require users to enter personal information to function, except when directly relevant to the core functionality of the app or required by law.

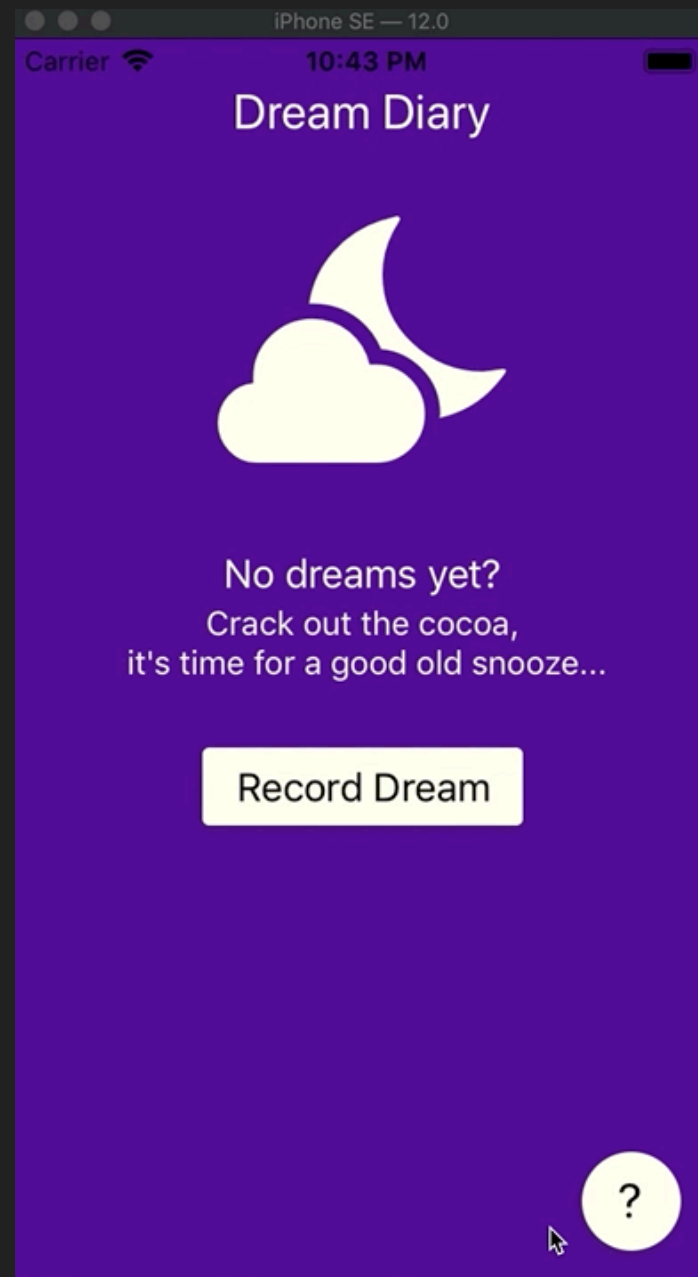
You should allow users to freely access your app's non-account-based features. For example, an e-commerce app should let users browse store offerings and other features that are not account-based before being asked to register, or a restaurant app should allow users to explore the menu before placing an order. Registration must then only be required for account-specific features, such as saving items for future reference or placing an order.

REJECTED

September 2019



Phases of growth



Useful bits:

Expo

Shadow CLJS

Tachyons

React hooks

Local storage

Keyboard Avoiding

Navigation

Available on GitHub

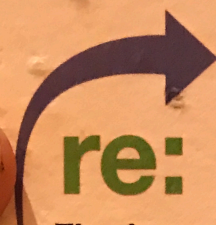
Riverford/clojure-exchange-demo-2019

1. Access to hardware features
2. Marketing + brand awareness
3. Rapid prototyping of ideas

Why build an app at all?

```
Emacs Prelude - ~/development/clojure-exchange-demo-2019/src/clojure_exchange_demo_2019/ideas.cljs
;; 1. buoyancy - app for helping ideas rise
;; 2. constellation - dot to dot icebreaker
;; 3. food chain - meal planner and fridge management system x
;; 4. dream diary - with 16 dream questions
;; 5. power ranges - track your strength in different rep ranges
;; 6. resonance - talking back app for focusing x
;; 7. arrows - a multiway dictionary for meaning
;; 8. focusing map to the body
;; 9. amino acid profile tracker for vegans
;; 10. language learning app
;; 11. ferment tracker - track
;; 12. supermarket sweep - put in the aisles in your supermarket and your shopping list and get the pick order
```

Seed bomb?



re:

Thank you for attending re:Clojure 2019

Your name badge is printed on biodegradable seed paper



Plant me



Water me



Wait for me to grow



cognitect

droit



uSwitch



MANNING



GRIFFIN



twilio