

Workshop

Universal Apps

@Baconbrix

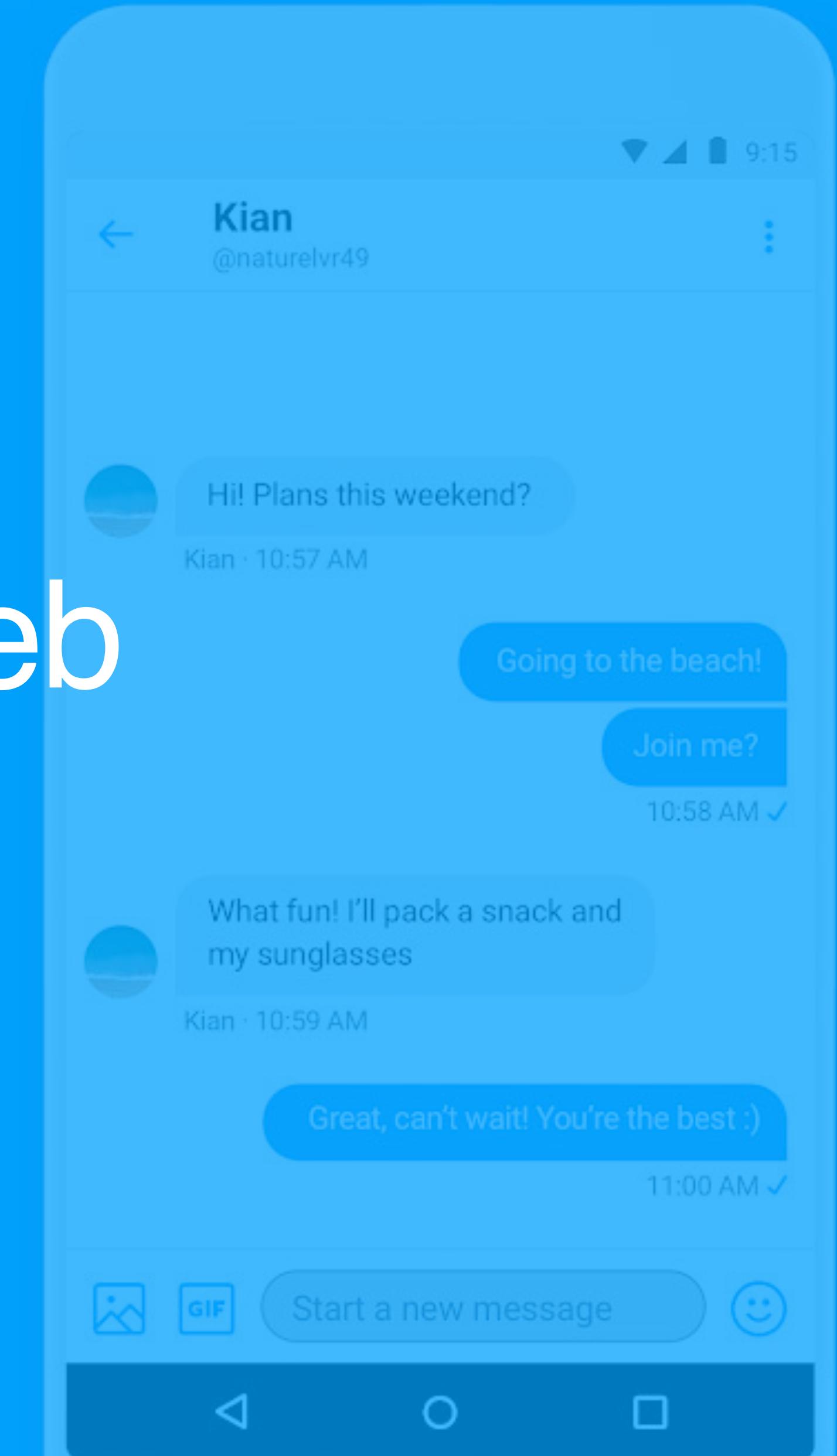
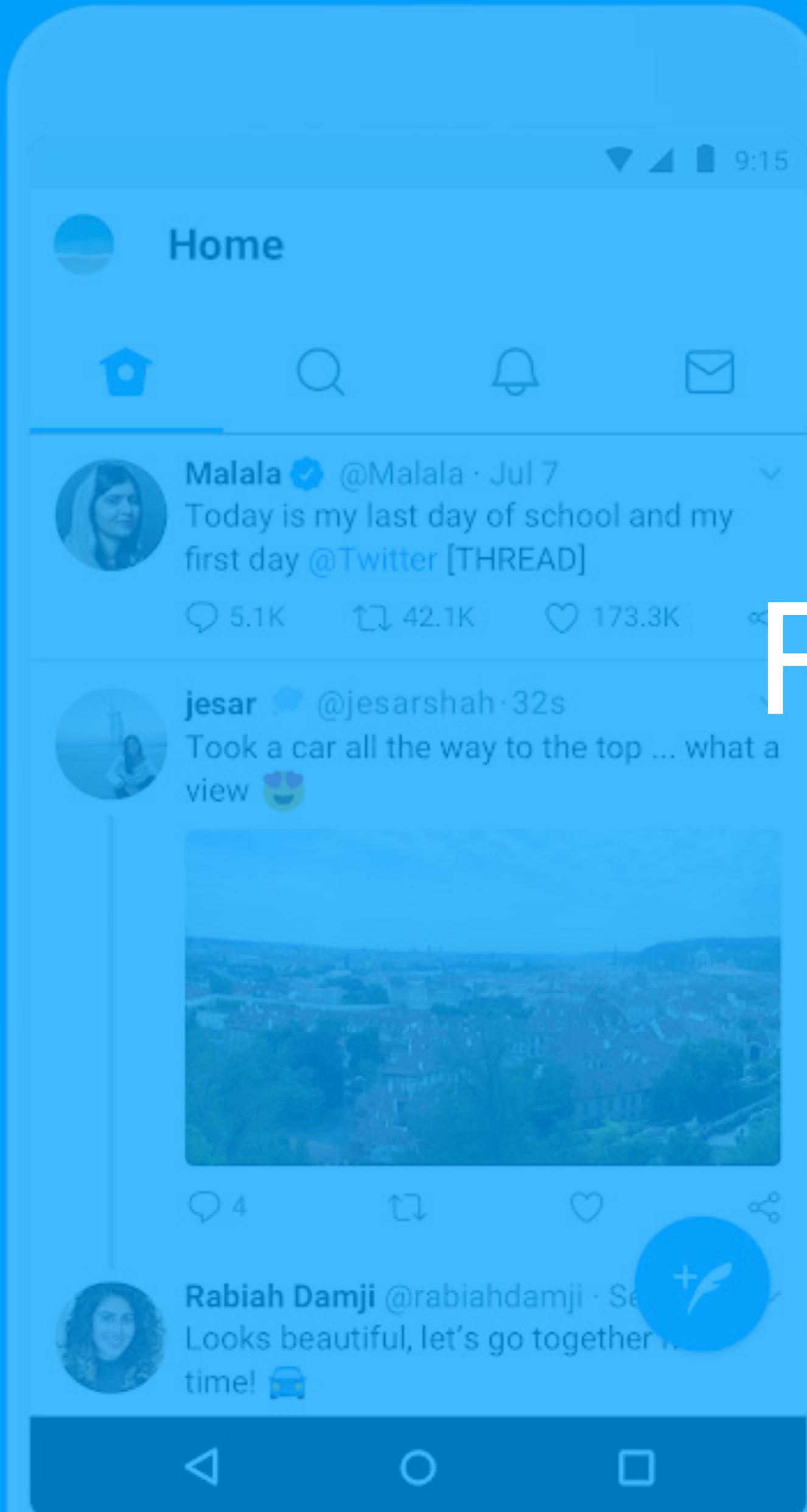
Evan Bacon

LEG
O

Expo

Gaming

React Native for Web



React Native for Web

- High-quality web interfaces
- Write once, render anywhere
- Accessibility
- First-Class I18N
- Powerful CSS-in-JS StyleSheet API

React Native for Web

```
<View /> => <div />  
<Text /> => <span />  
<Image /> => <img />
```

React Native for Web

- Twitter
- Major League Soccer
- Flipkart
- Uber
- The Times
- DataCamp

Workshop

Styles

Styles

```
const styles = StyleSheet.create({  
  container: {  
    flex: 1,  
    justifyContent: 'center'  
  },  
  text: {  
    margin: 10  
  }  
})
```

Styles

margin of 0, 0, 0, 0.

```
<style>
  .marginTop { margin-top: 10px; }
  .marginBottom { margin-bottom: 20px; }
  .margin { margin: 0; }
</style>
<div class='marginTop marginBottom margin'></div>
```

margin of 10, 0, 20, 0.

```
const style = [
  { marginTop: 10 },
  { marginVertical: 20 },
  { margin: 0 }
];
```

```
const Box = () => <View style={style} />
```

React Native for Web

The screenshot shows a web browser window titled "Performance tests" with the URL "necolas.github.io/react-native-web/benchmarks/". The page displays a table of benchmark results comparing various styling libraries. The left column lists the library and test type, the middle column shows the execution time, and the right column shows the standard deviation.

Library	Benchmark	Time (ms)
react-native-web	Mount deep tree (50)	24.99 ±05.20 ms
emotion@^10.0.9	Mount deep tree (50)	(S/L) 15.93/09.06 ms
inline-styles	Mount deep tree (50)	41.97 ±12.97 ms
react-fela@^10.2.1	Mount deep tree (50)	31.18 ±06.83 ms
react-native-web@0.11.2	Mount deep tree (50)	21.73 ±05.74 ms
react-native-web@0.11.2	Mount wide tree (50)	(S/L) 11.82/09.90 ms
react-native-web@0.11.2	Update dynamic styles (100)	31.80 ±08.19 ms
react-native-web@0.11.2	Mount text tree (50)	17.64 ±06.59 ms
react-native-web	Mount text tree	(S/L) 14.50/03.14 ms
react-native-web	Mount text tree	36.17 ±08.61 ms

Workshop

Weak spots

React Native for Web

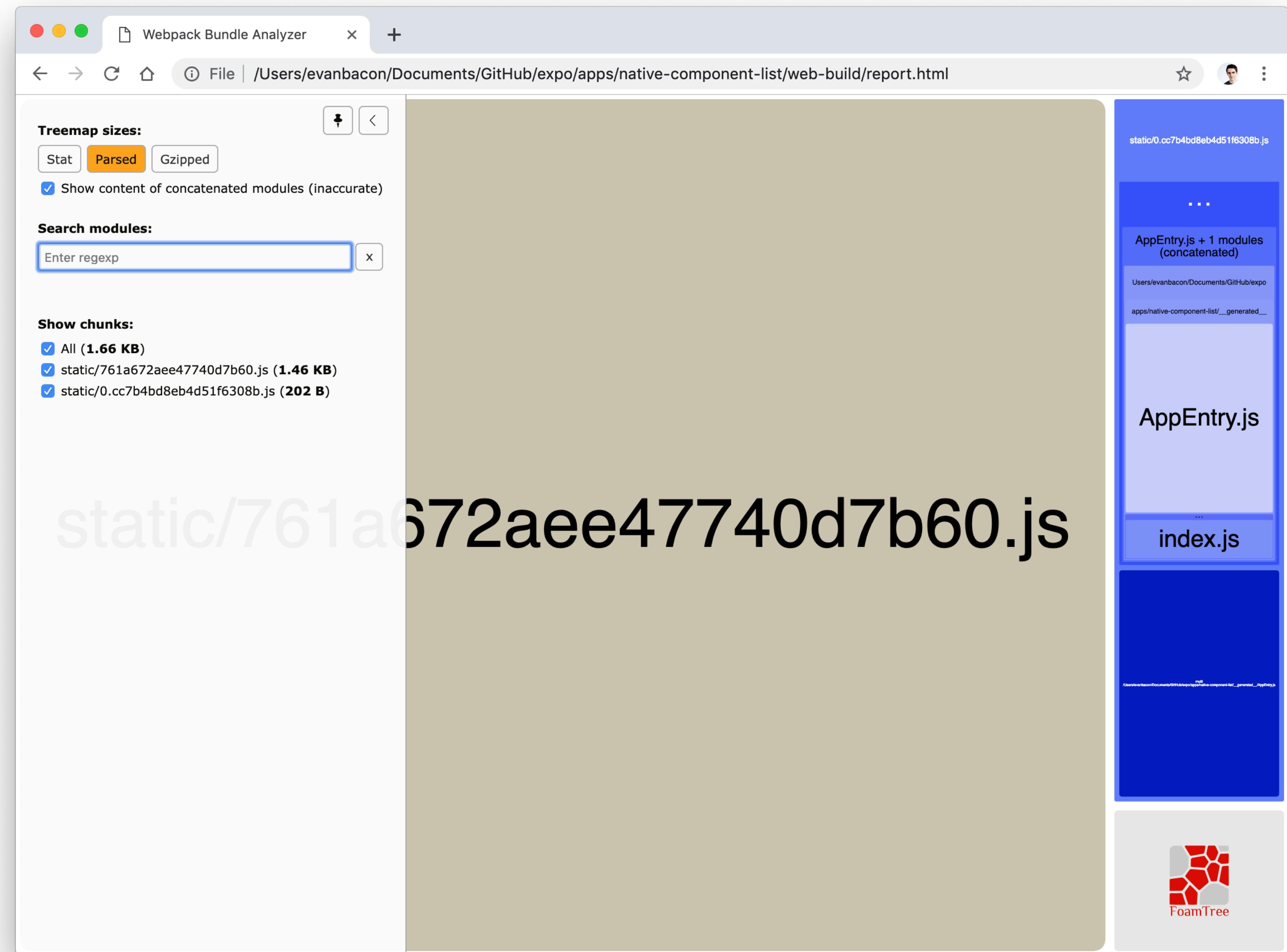
- Tree-Shaking is too fragile
- Use of internals in popular libraries
- CRA too restrictive, ejecting is too complex
- Trouble distinguishing web specific features
- Not enough universal libraries

How it works

```
// Before  
import { View } from 'react-native';  
  
// After  
import View from 'react-native-web/dist/exports/View';
```

React Native for Web

parsed: 1.6KB
GZIP: <200B



React Native for Web

```
// #1. Importing Everything
```

```
import ReactNative from 'react-native';
```

```
import * as ReactNative from 'react-native';
```

React Native for Web

```
// #2. Unsupported Modules
```

```
import { TabBarIOS } from 'react-native';
```

```
// After
```

```
import { TabBarIOS } from 'react-native-web';
```

`DatePickerIOS`

`DrawerLayoutAndroid`

`ImageEditor`

`ImageStore`

`InputAccessoryView`

`MaskedViewIOS`

`NavigatorIOS`

`PickerIOS`

`ProgressBarAndroid`

`ProgressViewIOS`

`SegmentedControlIOS`

`SnapshotViewIOS`

`Slider`

`TabBarIOS`

`ToastAndroid`

`ToolbarAndroid`

`ViewPagerAndroid`

`WebView`

`ActionSheetIOS`

`AlertIOS`

`CameraRoll`

`DatePickerAndroid`

`ImagePickerIOS`

`PermissionsAndroid`

`PushNotificationIOS`

`Settings`

`StatusBarIOS`

`Systrace`

`TimePickerAndroid`

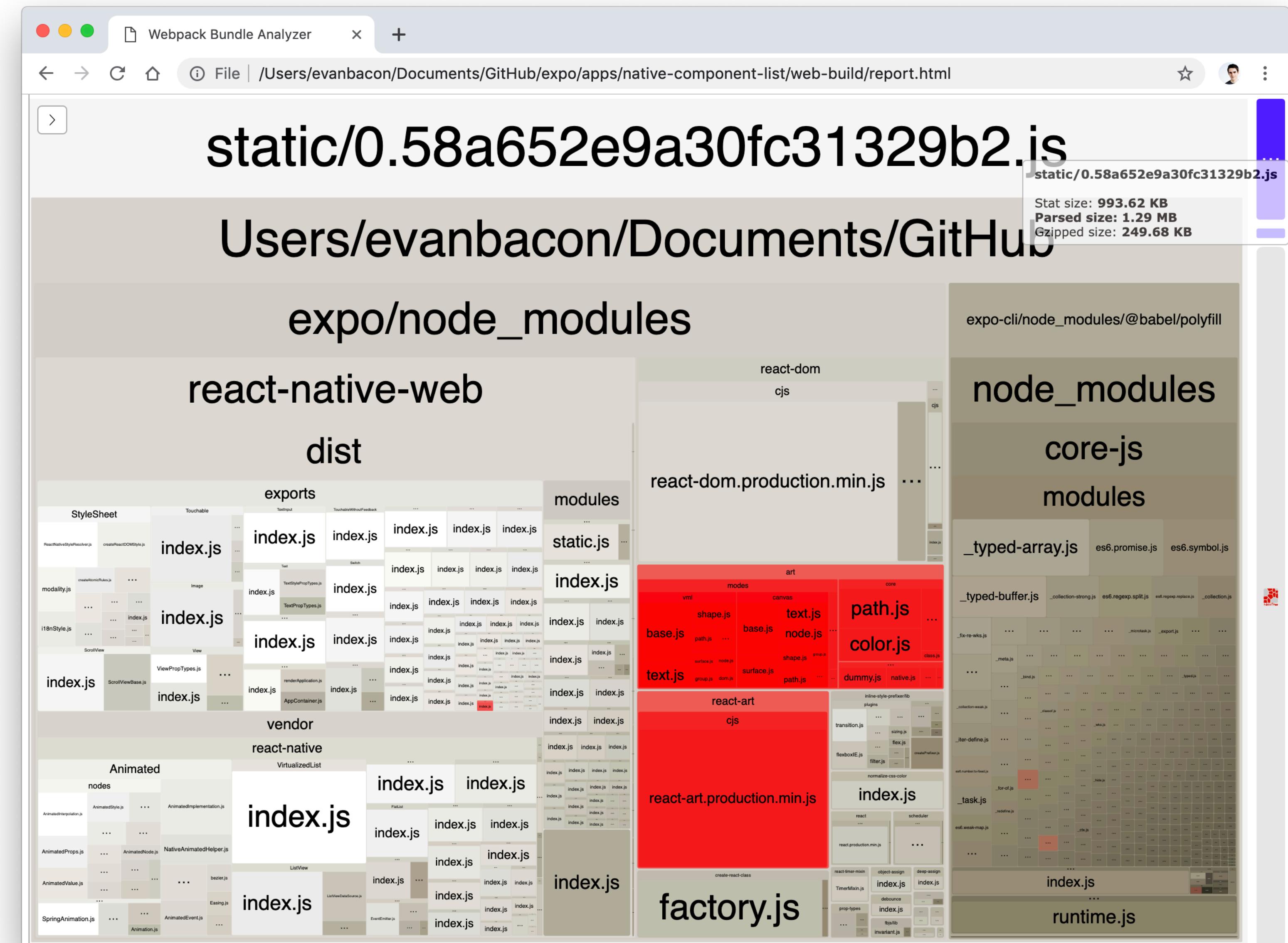
`TVEventHandler`

`VibrationIOS`

`ART`

React Native for Web

**parsed: 1MB
GZIP: ~250KB**



React Native for Web

- Tree-Shaking is too fragile
- Use of internals in popular libraries
- CRA too restrictive, ejecting is too complex
- Trouble distinguishing web specific features
- Not enough universal libraries

React Native for Web

```
import { AnimatedEvent } from 'react-native/Libraries/Animated/src/AnimatedEvent';

// After Webpack alias

import { AnimatedEvent } from 'react-native-web/Libraries/Animated/src/
AnimatedEvent';
```



React Native for Web

- Tree-Shaking is too fragile
- Use of internals in popular libraries
- CRA too restrictive, ejecting is too complex
- Trouble distinguishing web specific features
- Not enough universal libraries

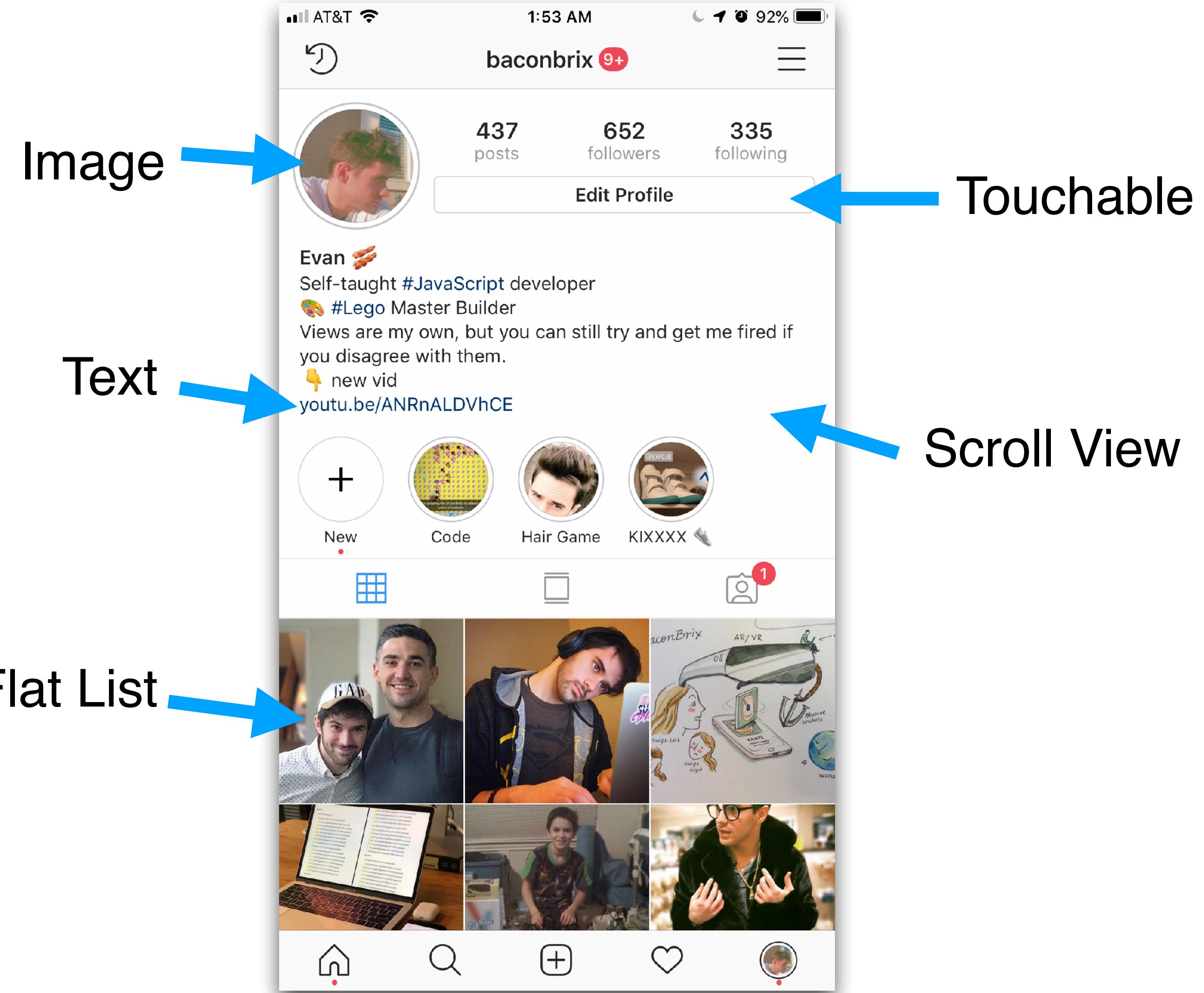
React Native for Web

- Tree-Shaking is too fragile
- Use of internals in popular libraries
- CRA too restrictive, ejecting is too complex
- Trouble distinguishing web specific features
- Not enough universal libraries

React Native for Web

- Tree-Shaking is too fragile
- Use of internals in popular libraries
- CRA too restrictive, ejecting is too complex
- Trouble distinguishing web specific features
- Not enough universal libraries

Primitives



Missing

SVG

View Shot

Camera

Video

Facebook
Gradient



Fonts

Audio

Blur View

Image Picker

Icons

Sharing

Workshop

Expo in the Browser

Expo Web

- Universal workflow for Expo CLI
 - Web support for the Expo SDK
 - Babel preset for multiple bundlers
 - PWA Generation

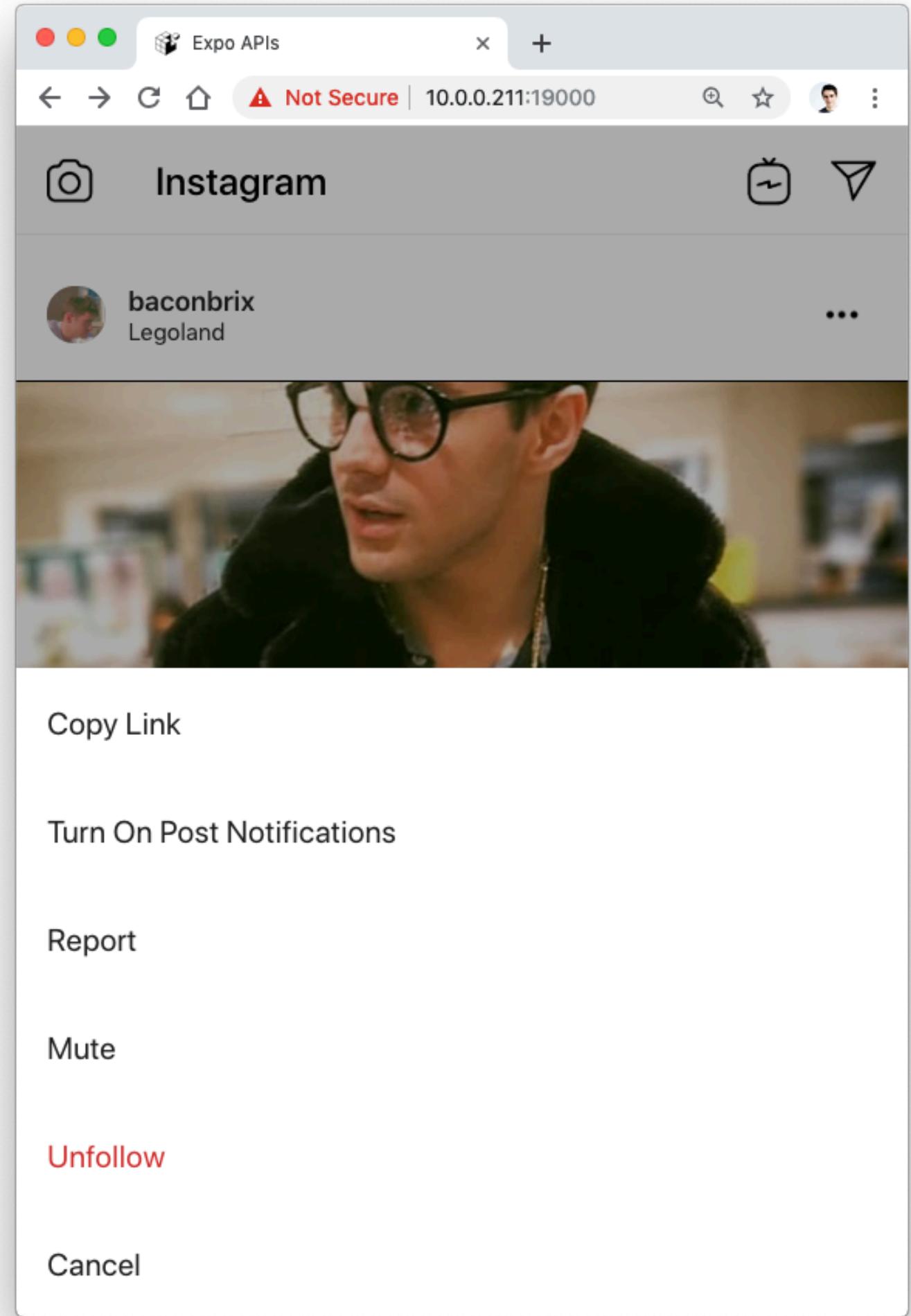
Expo Web

- Universal workflow for Expo CLI
- **Web support for the Expo SDK**
- Babel preset for multiple bundlers
- PWA Generation

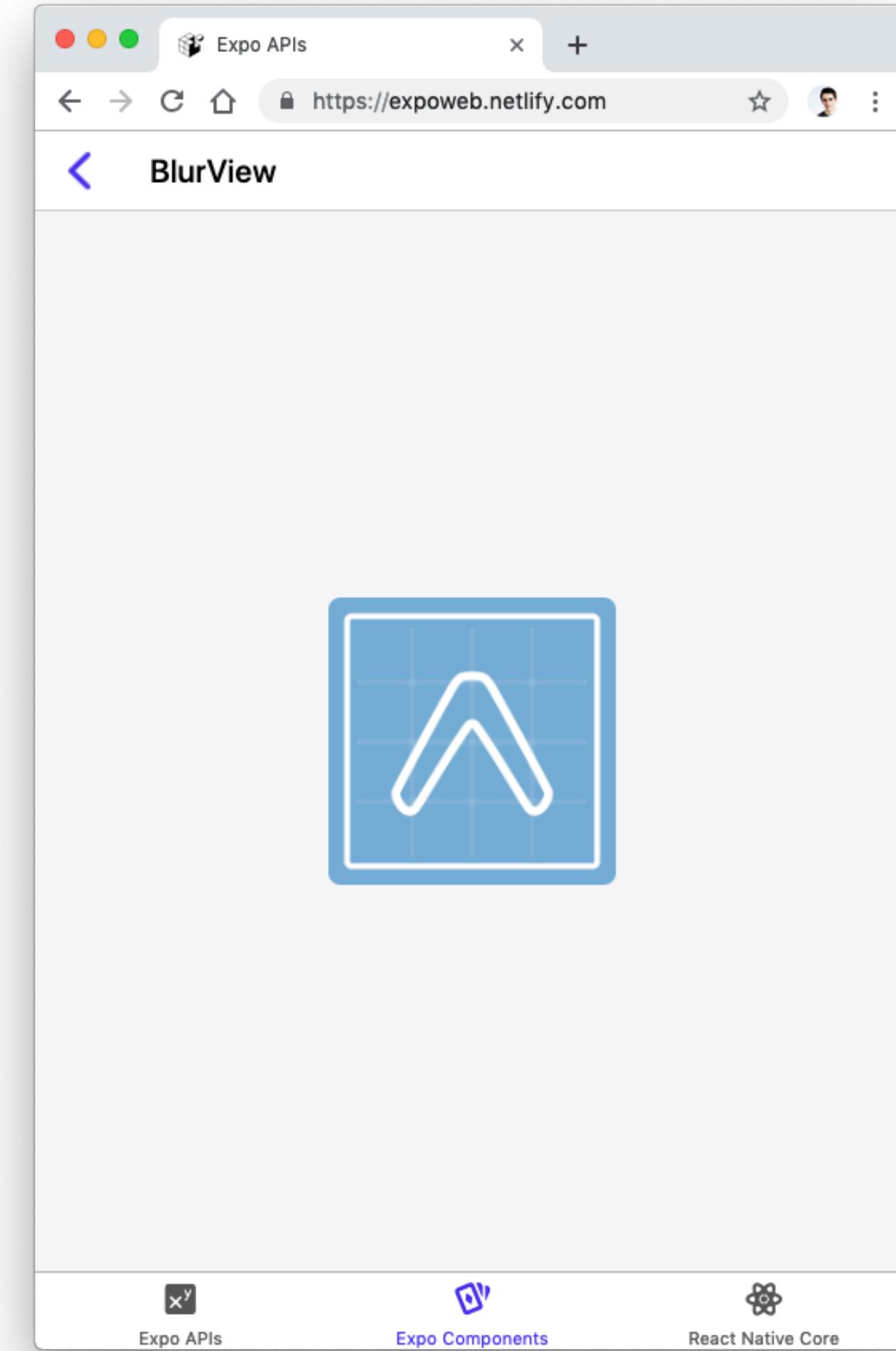
Expo Camera



Fallback



Action Sheet



Blur View

Expo for Web

The screenshot shows a web browser window titled "Native Directory" displaying the URL <https://www.native.directory>. The page lists three Expo modules:

- expo-random**: An Expo universal module for random bytes. It has a "GOOD" quality rating. It supports iOS, Android, Expo, and Web. It was last updated 3 months ago, has 5416 stars, and 24 downloads monthly. It contains 492 issues and a link to its website. Related tags include: [react-native](#)[231], [expo](#)[46], [random](#)[1], [crypto](#)[2], [random-bytes](#)[1], [secure](#)[2], [generator](#)[1], and [bytes](#)[1].
- expo-processing**: Utilities for using Processing.js on Expo. It has a "GOOD" quality rating. It supports iOS, Android, Expo, and Web. It was last updated 3 months ago, has 5416 stars, and 207 downloads monthly. It contains 492 issues and a link to its website. Related tags include: [expo](#)[46], [processing.js](#)[1], [graphics](#)[2], [opengl](#)[1], and [gl](#)[2].
- expo-print**: Provides an API for iOS (AirPrint) and Android printing functionality. It has a "GOOD" quality rating. It supports iOS, Android, Expo, and Web. It was last updated 3 months ago, has 5416 stars, and 140231 downloads monthly. It contains 492 issues and a link to its website. Related tags include: [react-native](#)[231], [expo](#)[46], and [print](#)[1].

A sidebar on the right lists various related tags:

- [crashlytics](#)[1]
- [credit-card](#)[1]
- [crop](#)[1]
- [cross-platform](#)[2]
- [crypto](#)[2]
- [css](#)[3]
- [css-modules](#)[2]
- [custom](#)[1]
- [customize-date](#)[1]
- [customize-event](#)[1]
- [d3](#)[1]
- [d3js](#)[1]
- [d3v4](#)[1]
- [data-visualization](#)[1]
- [database](#)[2]
- [date](#)[1]
- [datepicker](#)[1]
- [datepicker-component](#)[1]
- [datepickerios](#)[1]
- [debugging](#)[1]
- [declarative](#)[1]
- [deep-linking](#)[1]
- [demo-app](#)[1]
- [detection](#)[1]

Expo Web

- Universal workflow for Expo CLI
- Web support for the Expo SDK
- Babel preset for multiple bundlers
- PWA Generation

Expo Web

- Universal workflow for Expo CLI
- Web support for the Expo SDK
- Babel preset for multiple bundlers
- PWA Generation

Workshop

Adding Web Support

Expo for Web

```
yarn add expo@next react-native-web react-dom
```

```
yarn add -D babel-preset-expo@^5.1.1
```

Expo for Web

```
expo start
```

```
expo start --web
```

```
expo start --web-only
```

```
expo start --https --no-https
```

```
expo start --dev --no-dev
```

Expo for Web

```
// package.json
{
  "main": "node_modules/expo/AppEntry.js",
  "dependencies": {
    "expo": "^33.0.0-alpha.web.3",
    "react": "16.5.0",
    "react-dom": "^16.8.6",
    "react-native-web": "^0.11.2"
  },
  "devDependencies": {
    "babel-preset-expo": "^5.1.1"
  }
}
```

Expo for Web

```
module.exports = function(api) {  
  api.cache(true);  
  return {  
    presets: ['babel-preset-expo'],  
  };  
};
```

Expo for Web

```
expo-web-demo $ expo start
Starting project at /expo-web-demo
Expo DevTools is running at http://localhost:19003
Press d to open DevTools now, or shift-d to always open it automatically.
Error: Problem validating fields in app.json.
See https://docs.expo.io/versions/v32.0.0/workflow/configuration/
• Field: platforms - should NOT have more than 2 items.
```

Expo Web

- Babel 7 (`babel.config.js`)
- `babel-preset-expo@^5.1.1`
- `expo@next`
- `app.json` exists
- App.js exports a React Component
- `package.json main` is set to
`node_modules/expo/AppEntry`

Exercise: Run this locally

- Hint: it's in beta, feel free to ask for help :)
- - Expo 33 alpha won't run on native until the official release

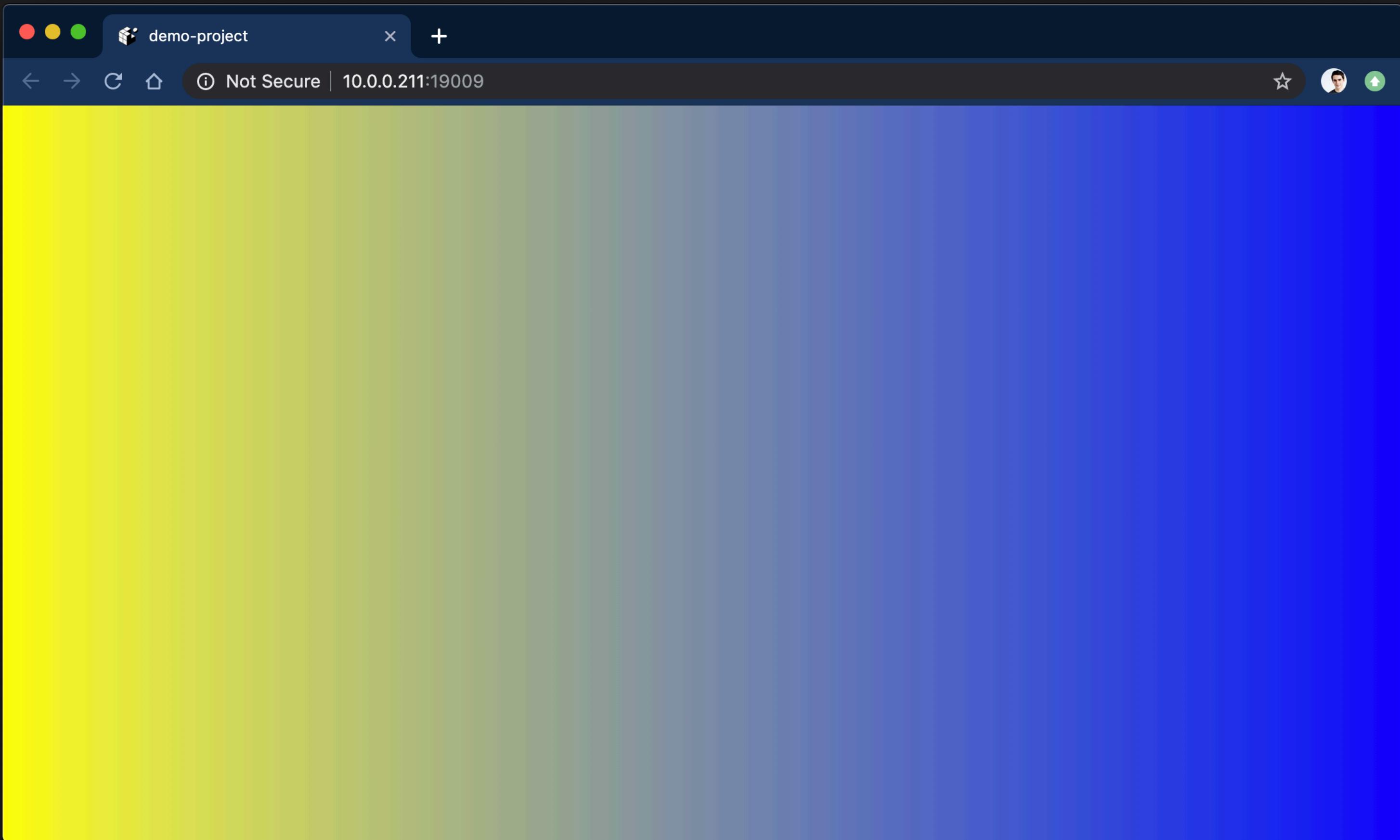
Workshop

Expo Primitives

Expo for Web

```
import { LinearGradient } from 'expo';
import React from 'react';
export default class App extends React.Component {
  render() {
    return (
      <LinearGradient
        colors={['yellow', 'blue']}
        style={{ flex: 1 }}>
        />
    );
  }
}
```

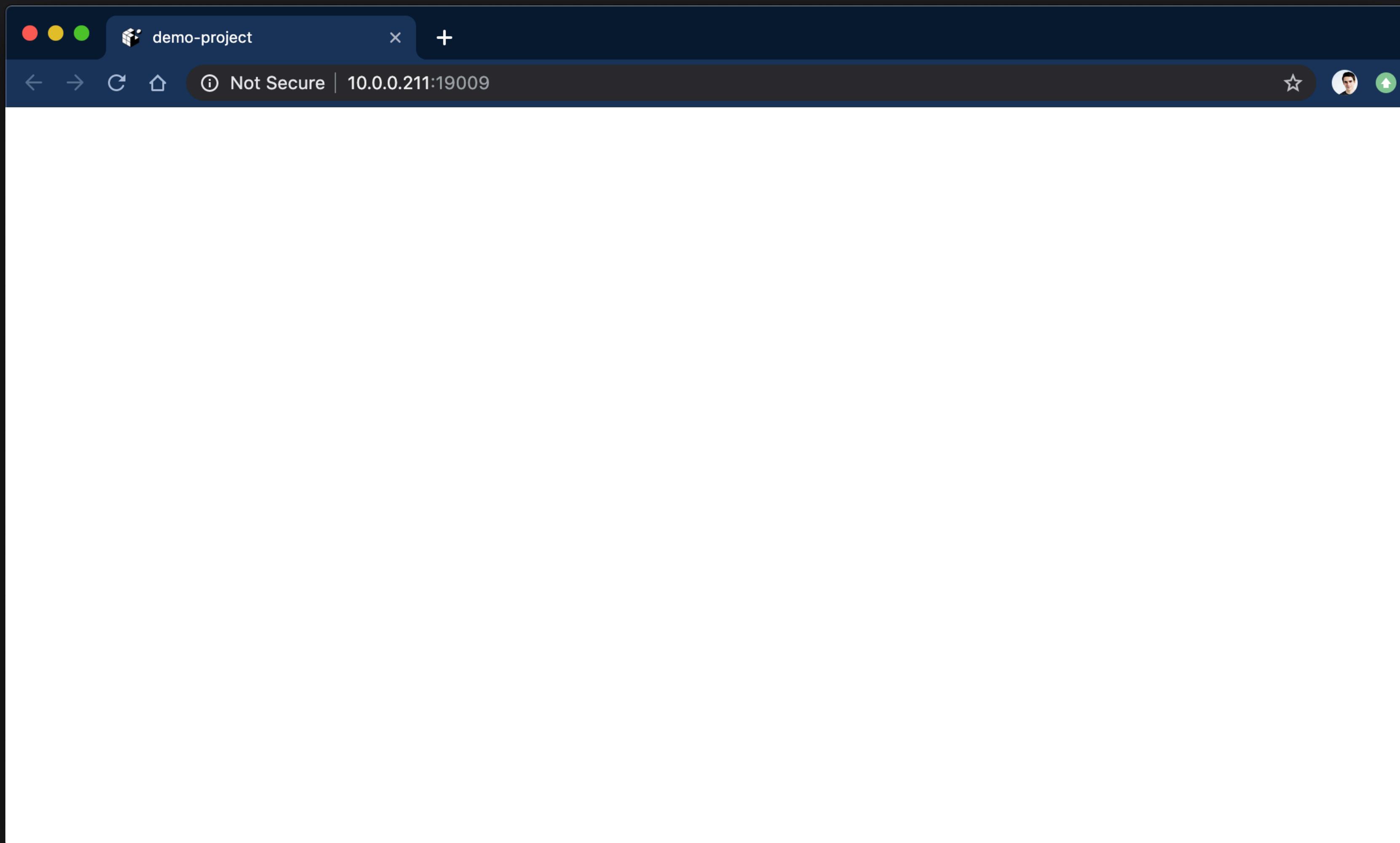
Expo for Web



Expo for Web

```
import { Camera } from 'expo';
import React from 'react';
export default class App extends React.Component {
  render() {
    return (
      <Camera
        style={{ flex: 1 }}>
        />
    );
  }
}
```

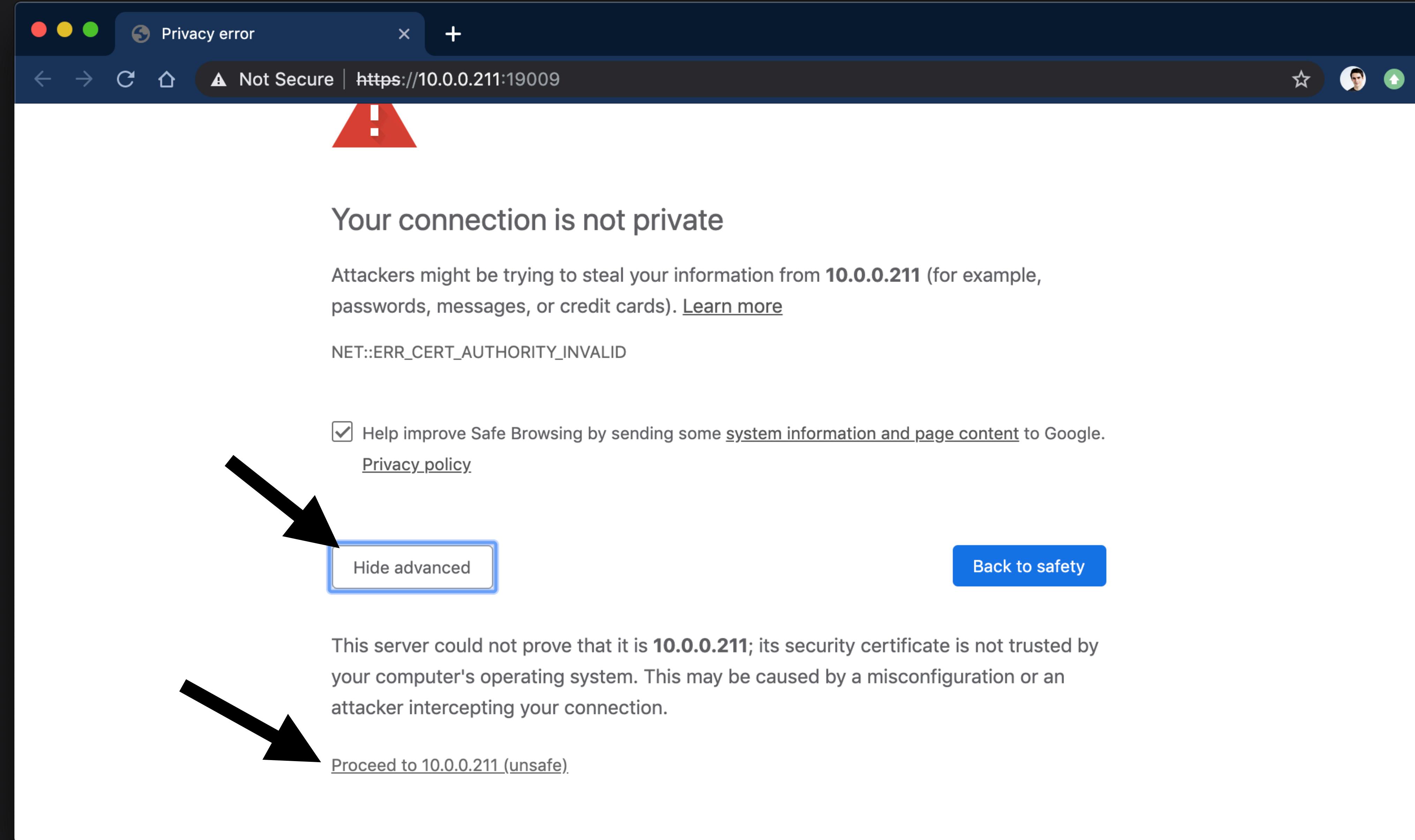
Expo for Web

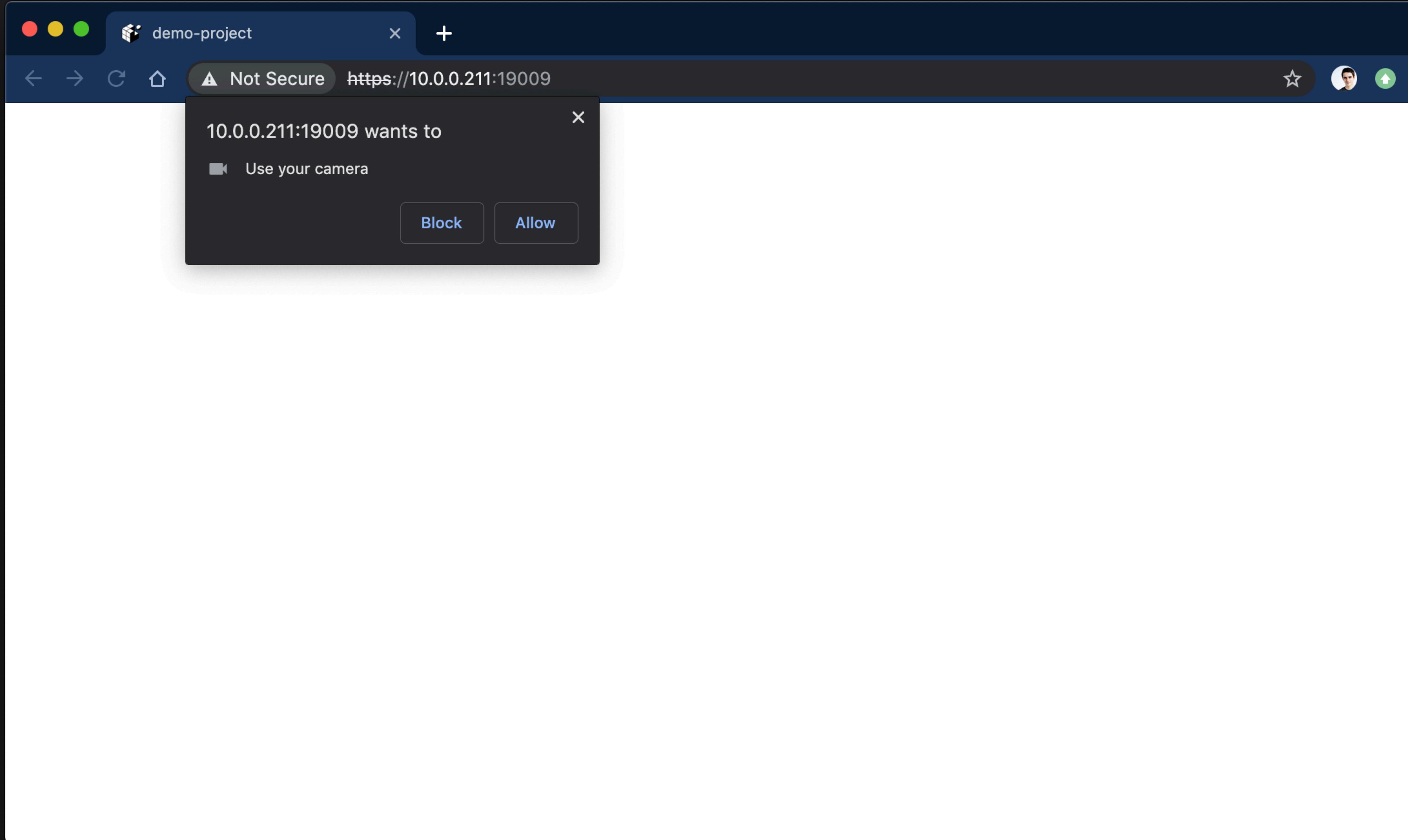


Expo for Web

```
expo start --web --https
```







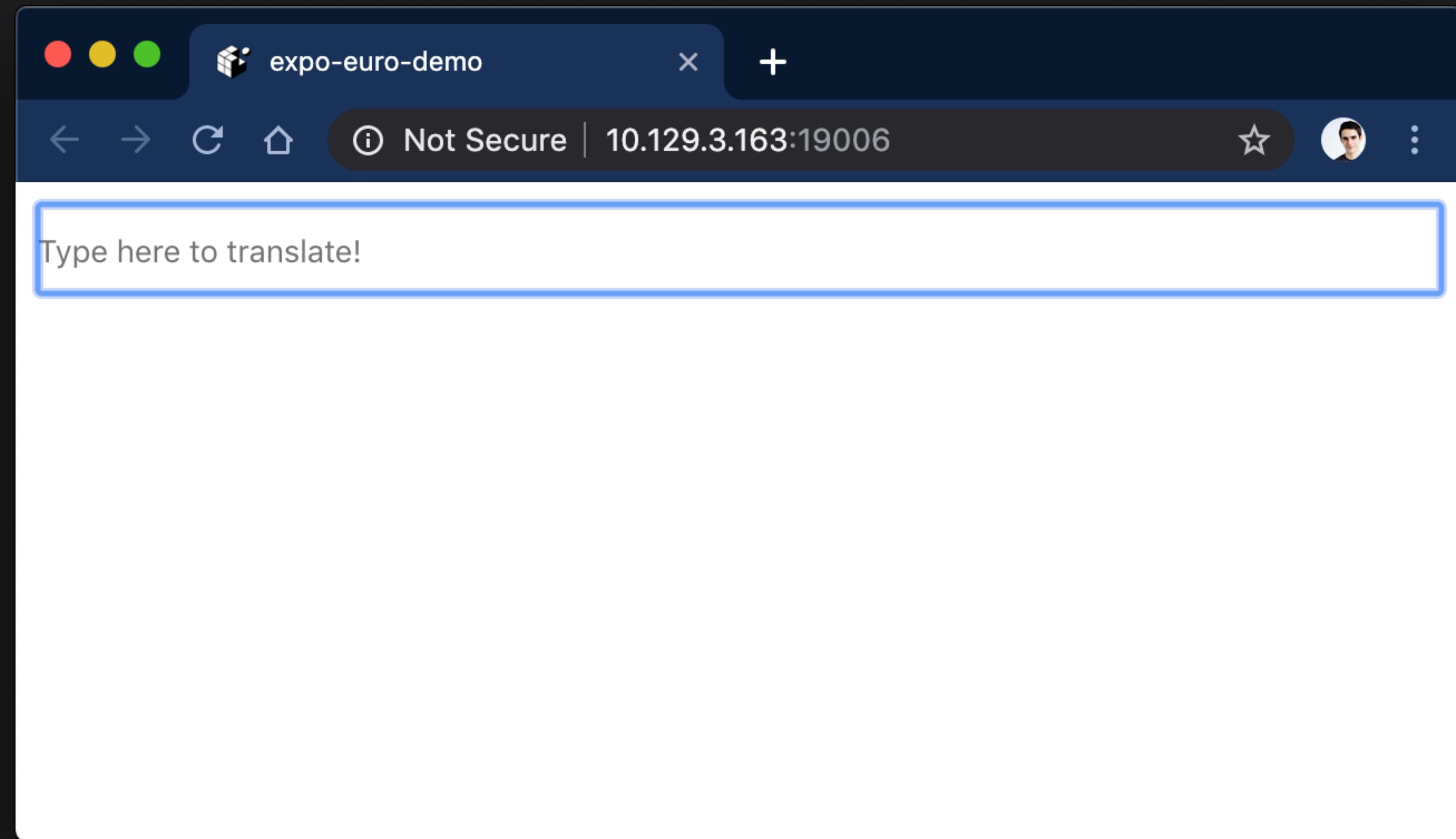
Exercise: Use the Camera

- You'll need to run this locally because code sandbox support isn't complete
- Camera works very differently based on platform.
- Bonus: Find which platform doesn't work 😊

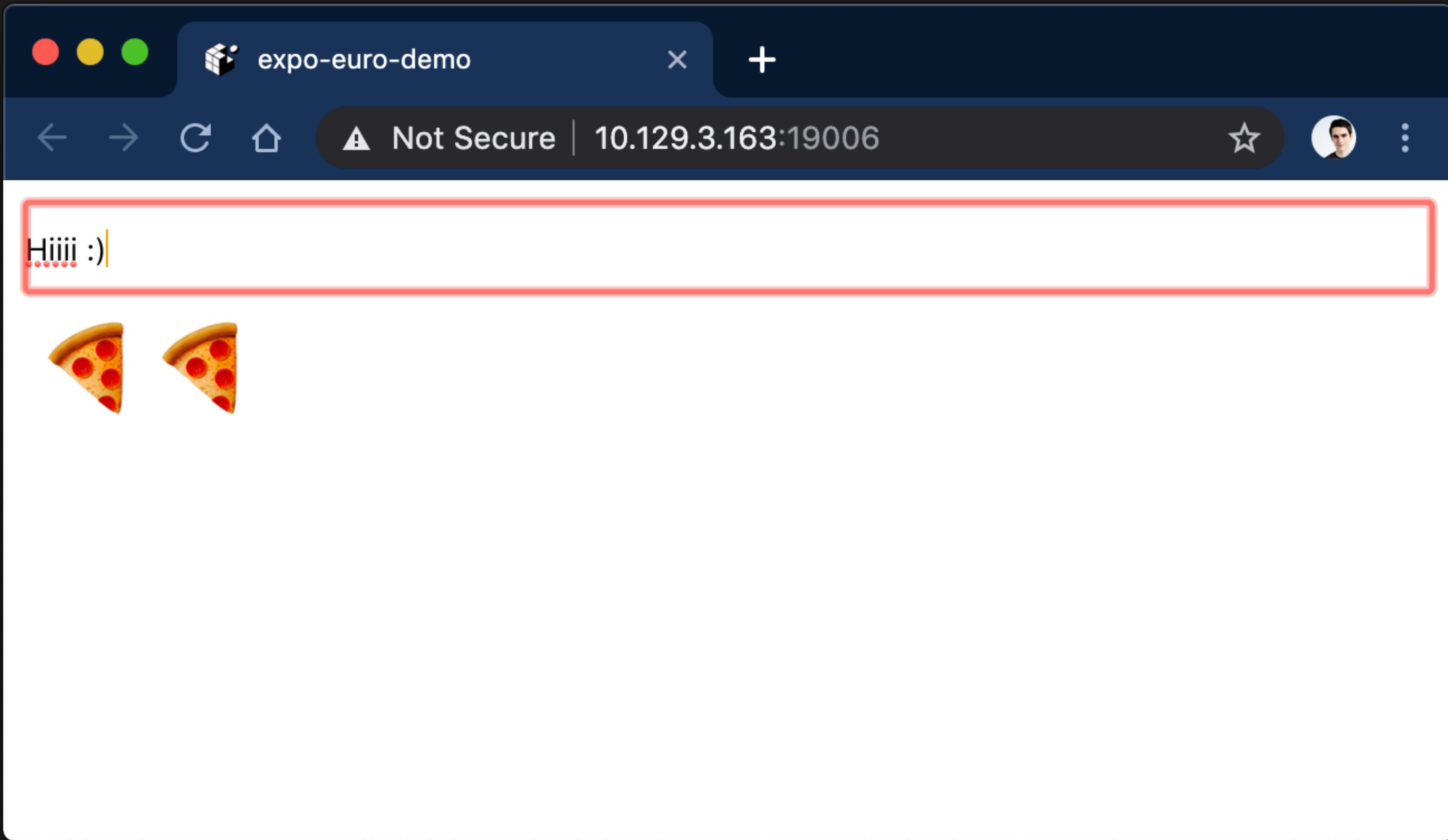
Workshop

React Native Primitives

```
<TextInput  
  style={styles.input}  
  placeholder="Type here boiii!"  
  onChangeText={text => this.setState({ text })}  
/>
```



```
input: {  
  caretColor: 'orange',  
  outlineColor: 'red',  
}
```



Exercise: use outline props on Text or Text Input

- CSS: outline-color, outline-width, outline-style, outline-offset
- Hint: Camel case: outline-width = outlineWidth
- Sandbox: bit.do/rn-web-outline-ex
- All web styles: tiny.cc/pka16y

Workshop

Accessibility Role

Accessibility Role

```
import { Text, Linking } from 'react-native';

const Link = ({ url }) => (
  <Text
    onPress={() => {
      Linking.openURL(url);
    }}
  >
    Open Link
  </Text>
);
```

Accessibility Role

```
<div  
  dir="auto"  
  data-focusable="true"  
  tabindex="0"  
  class="...>  
  Open Link  
</div>
```

Accessibility Role

- No link preview
- No “Copy Link Address”
- Cannot drag on mobile
- No Peek-and-Pop

Accessibility Role

```
import { Text, Linking } from 'react-native';

const Link = ({ url }) => (
  <Text
    + accessibilityRole="link"
    + href={url}
    onPress={() => {
      Linking.openURL(url);
    }}
  >
    Open Link
  </Text>
);
```

Accessibility Role

```
<a  
    dir="auto"  
    role="link"  
    href="#">  
    data-focusable="true"  
    class="...>  
        Open Link  
</a>
```

Accessibility Role

- VoiceOver on iOS
- TalkBack on Android
- Screen Readers on Web

Accessibility Role

```
<View accessibilityRole="article" /> => <article role="article" />
<View accessibilityRole="banner" /> => <header role="banner" />
<Text accessibilityRole="label" /> => <label />
<Text accessibilityRole="link" /> => <a role="link" />
<View accessibilityRole="main" /> => <main role="main" />
```

Accessibility Role

```
<Text accessibilityRole="heading" /> => <h1 />  
<Text accessibilityRole="heading" aria-level="3" /> => <h3 />
```

Accessibility Role

```
const Button = () => (
  <TouchableHighlight
    accessibilityRole="button"
    onPress={() => {}}>
    <Text>Press me!</Text>
  </TouchableHighlight>
);
```

Exercise: Create a universal Link component

- Sandbox: bit.do/rn-web-link-ex
- Bonus: Inspect element, be confused by DOM, install react-developer-tools

Workshop

Pseudo Classes

```
button {  
  color: red;  
}
```

```
button:hover {  
  color: blue;  
}
```

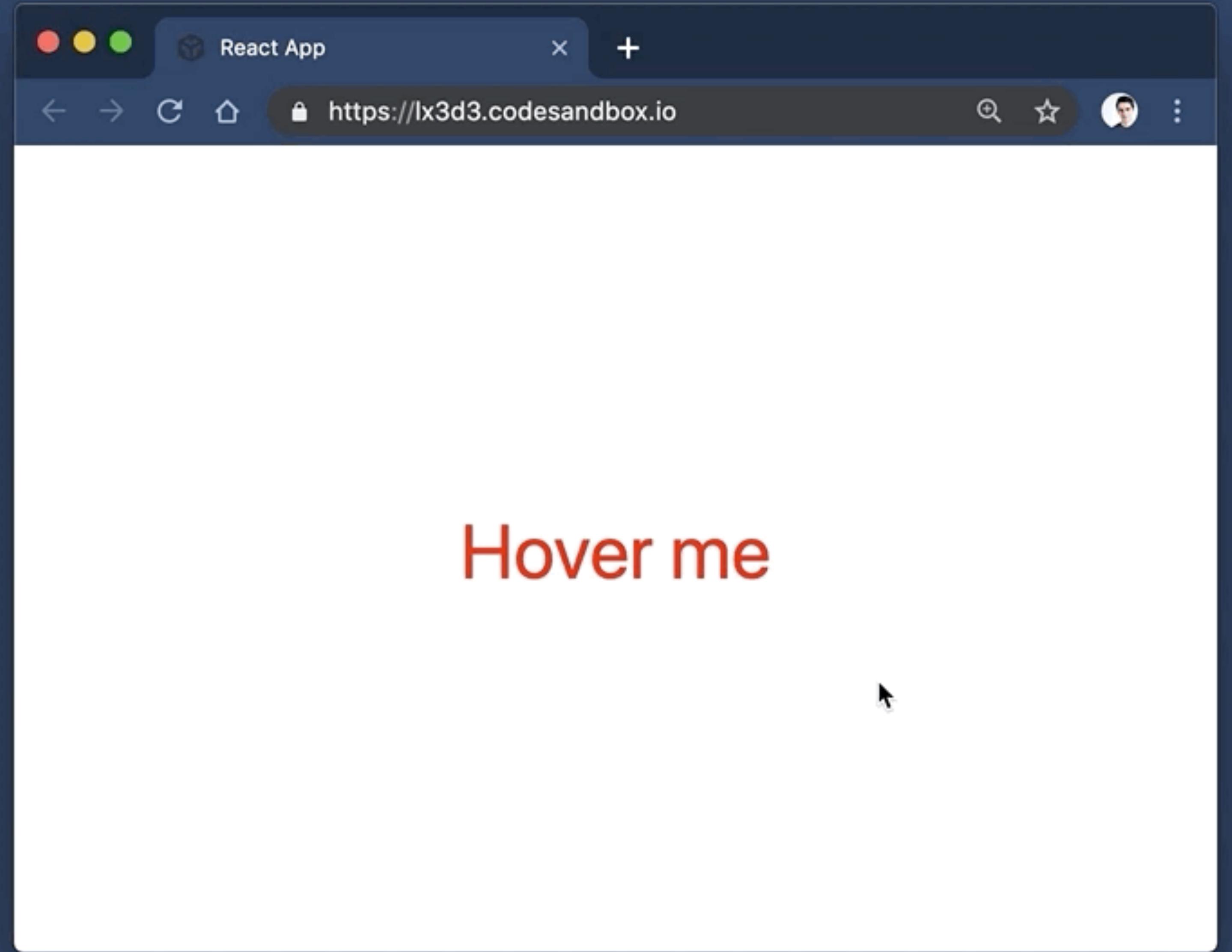
```
return React.cloneElement(  
  React.Children.only(child), {  
    onMouseEnter() {  
      this.setState({ isHovered: true })  
    },  
    onMouseLeave() {  
      this.setState({ isHovered: false })  
    },  
  });
```

```
yarn add @expo/style-utils
```



```
import React from 'react';
import { StyleSheet, Text, View } from 'react-native';
import { Hoverable } from '@expo/style-utils';

export default class App extends React.Component {
  render() {
    return (
      <View style={styles.container}>
        <Hoverable>
          {isHovering => (
            <Text
              style={{
                color: isHovering ? 'blue' : 'red',
              }}
            >
              Hover me
            </Text>
          )}
      </View>
    );
  }
}
```



Exercise: use hover in your project

- Sandbox: bit.do/rn-web-hover-ex
- Bonus: Use hooks somehow I guess

Workshop

Animations

- Layout Animation
- Animated API
- Animatable

```
export default class App extends Component {
  state = { isExpanded: false };

  componentWillMount() {
    LayoutAnimation.easeInEaseOut();
  }

  render() {
    const { isExpanded } = this.state;

    return (
      <View
        onPress={() => this.setState({ isExpanded: !isExpanded })}
        style={{ transform: [{ scale: isExpanded ? 10 : 1 }] }}
      />
    );
  }
}
```





A Expo

CSS Transition

```
componentWillUpdate() {
  LayoutAnimation.easeInEaseOut();
}

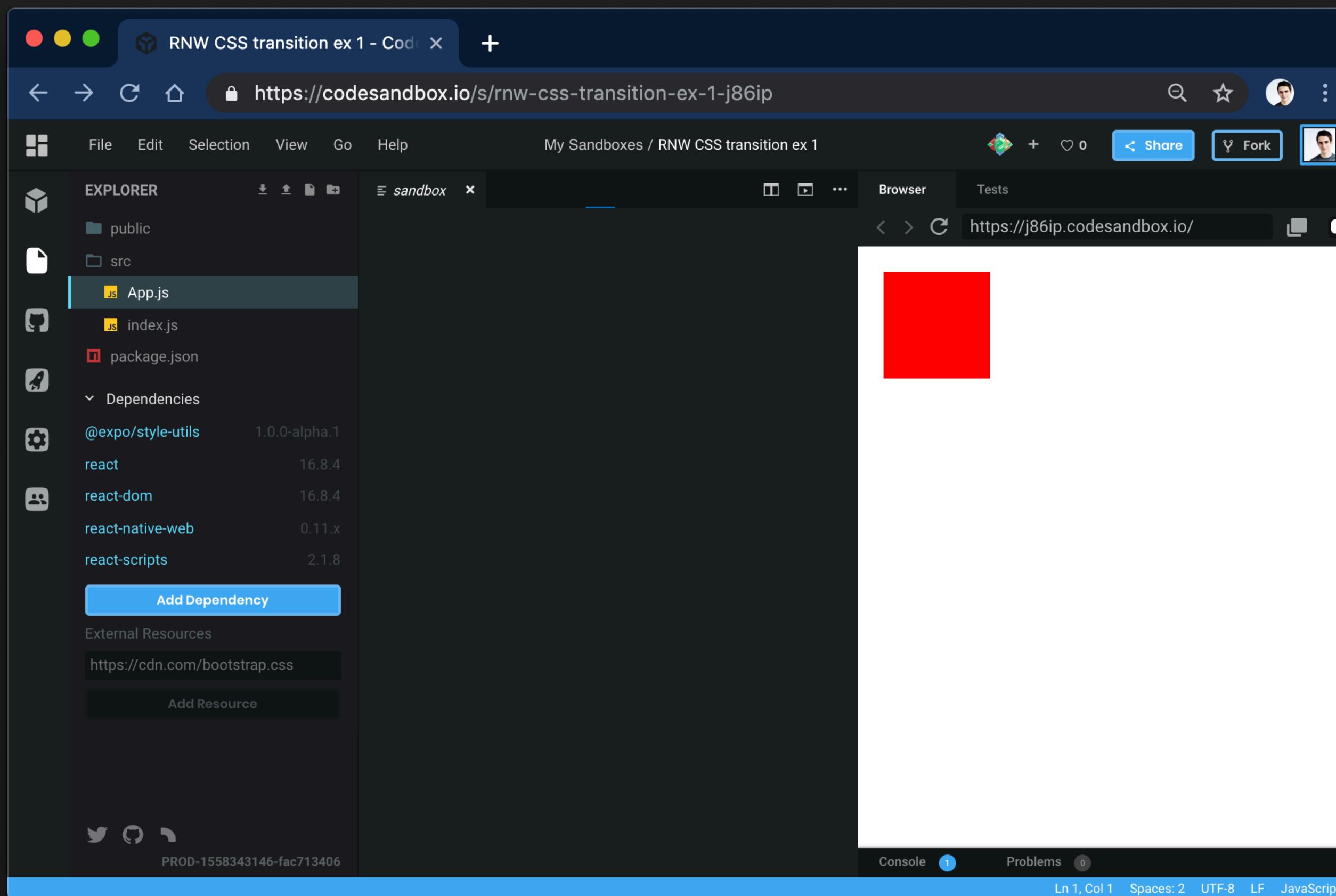
render() {
  const { isExpanded } = this.state;

  return (
    <View
      onPress={() => this.setState({ isExpanded: !isExpanded })}
      style={[
        +
        {
          transitionDuration: '1s',
          transitionProperty: 'all',
          transitionTimingFunction: 'ease-in-out',
        },
        { transform: [{ scale: isExpanded ? 10 : 1 }] }
      ]}
    />
  );
}
```



Exercise: use CSS transitions

- bit.do/rn-web-css-ex-1



- Layout Animation
- Animated API
- Animatable

- Layout Animation
- Animated API
- Animatable

```
import * as Animatable from "react-native-animatable";  
  
<Animatable.Text animation="fadeIn">Hi world</Animatable.Text>
```

Exercise: Play with animations

- Sandbox: bit.do/rn-web-animateable

Workshop

Navigation

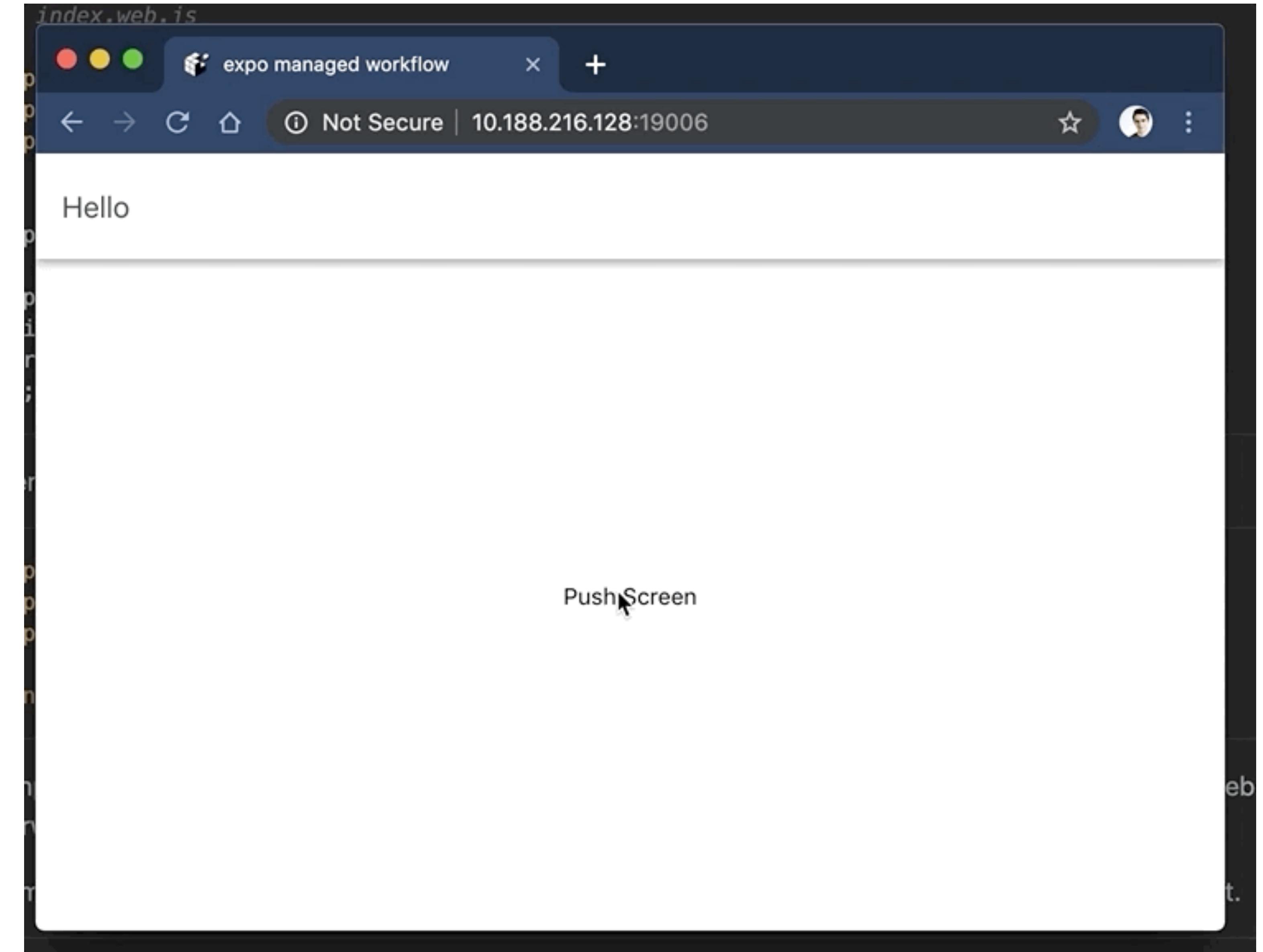
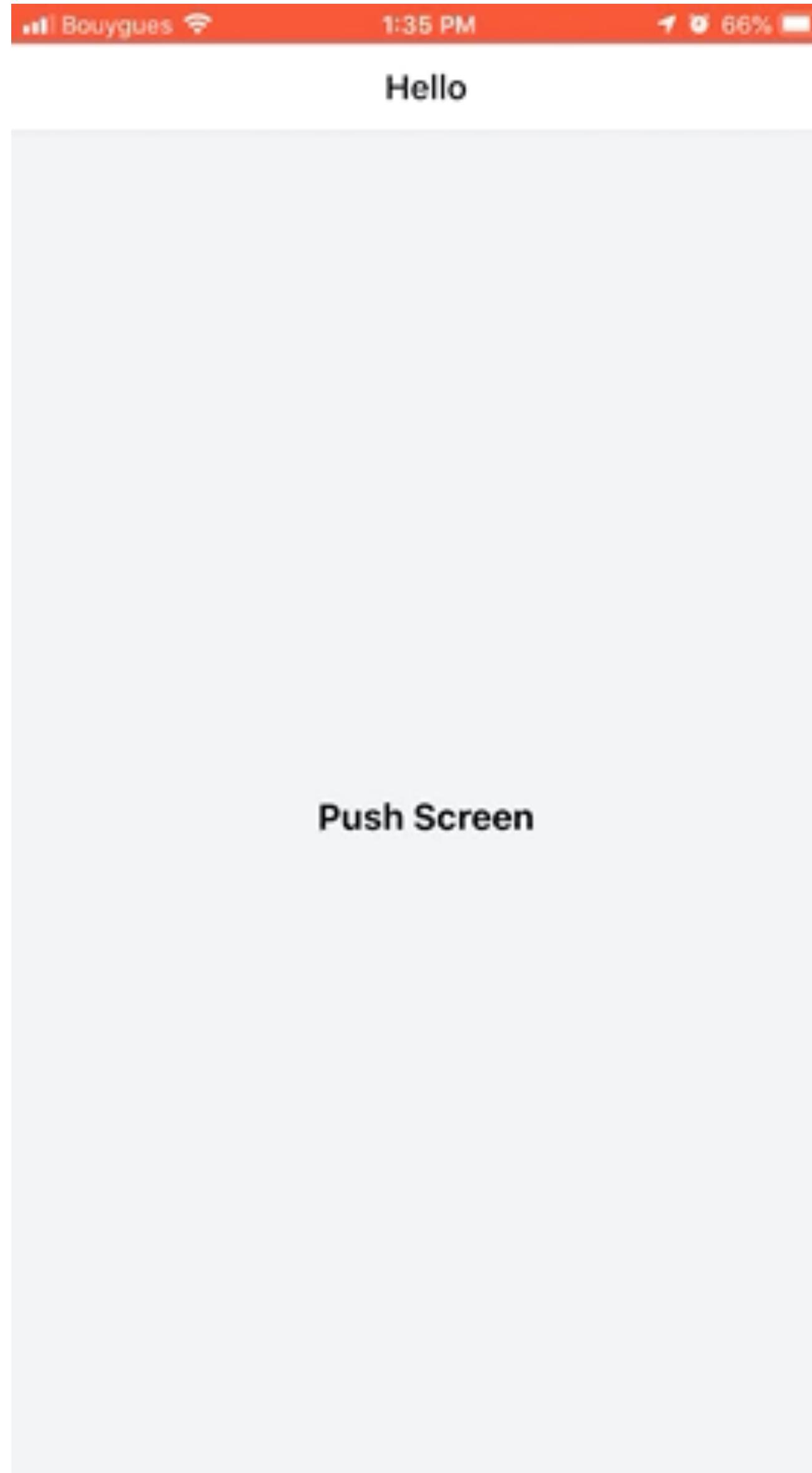
```
import React from 'react';
import { App } from './App';
import {
  createStackNavigator,
  createAppContainer
} from 'react-navigation';

const Navigator = createStackNavigator({
  App,
});

const app = createAppContainer(Navigator);

export default app;
```





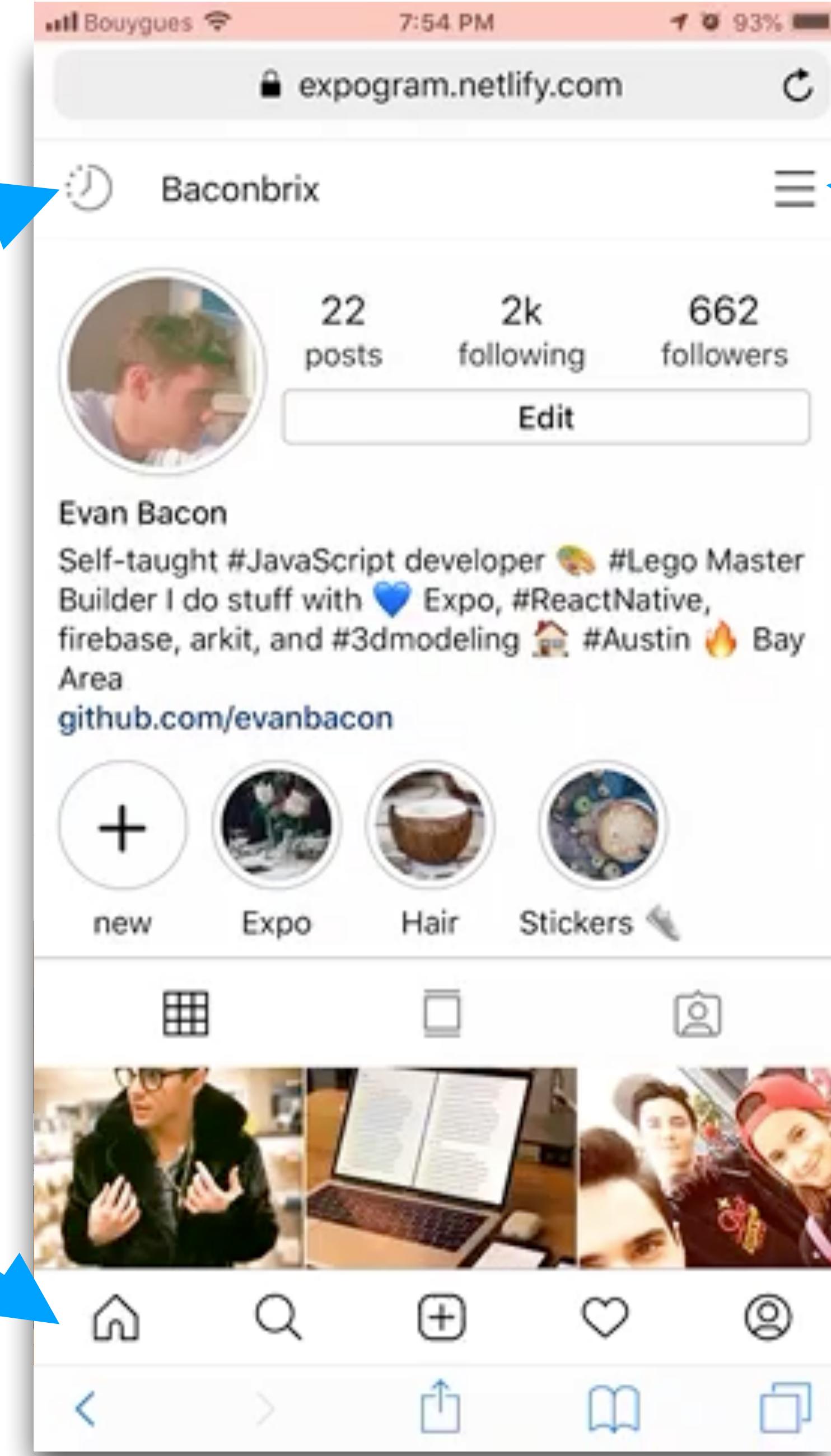
snack.expo.io/@bacon/react-euro-nav-example

Navigation

Stack Nav

Drawer Nav

Tab Nav



Web Nav

- No history API
- First meaningful paint takes a while
- Mobile gestures and controls don't work as expected

Web Nav

```
yarn add react-navigation @react-navigation/web
```

```
import React from 'react';
import { App } from './App';
import {
  createStackNavigator,
-  createAppContainer
} from 'react-navigation';
+ import { createBrowserApp } from '@react-navigation/web';
```

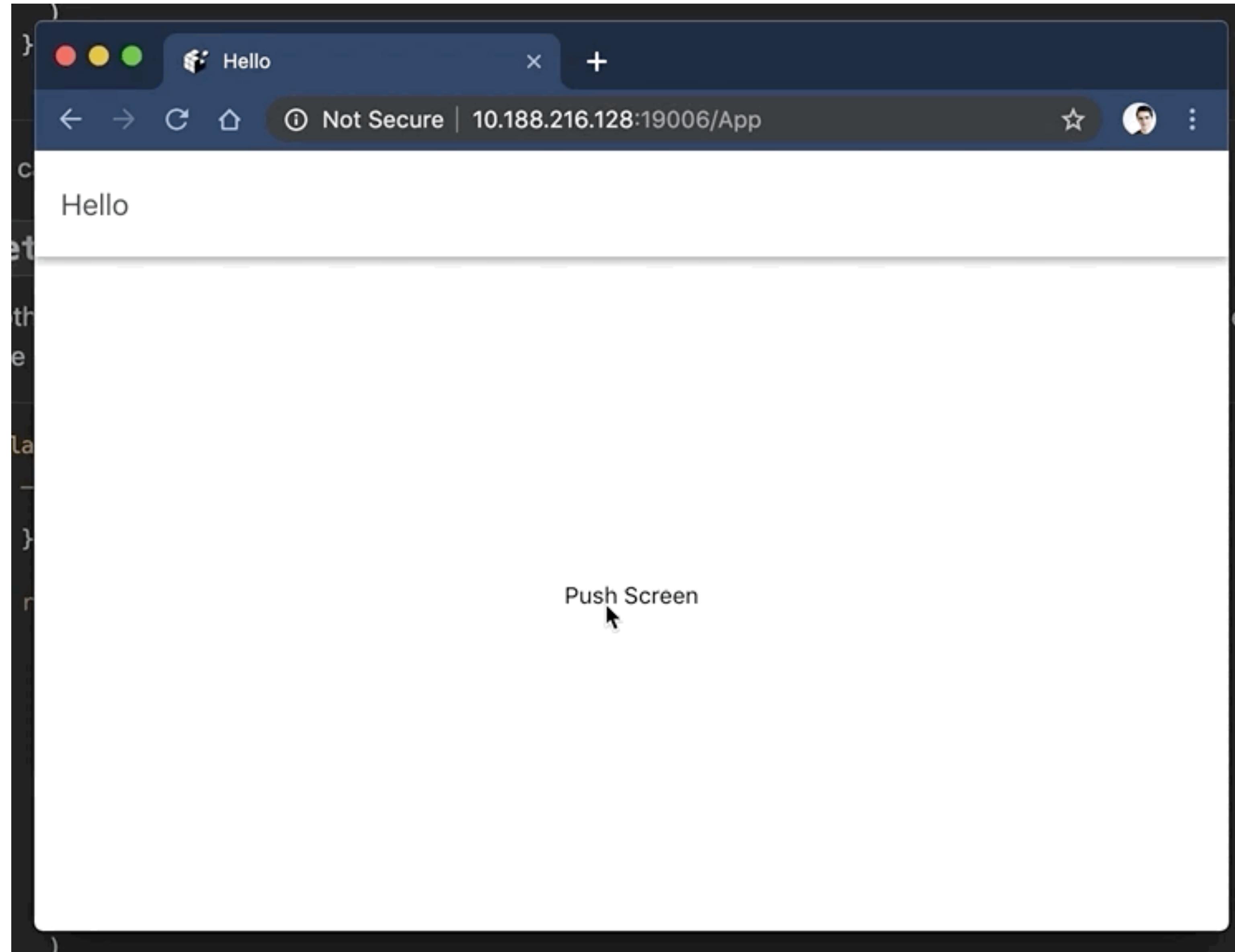
```
const Navigator = createStackNavigator({
  App,
  App2: App,
});
```

```
// Add routing
+ const app = createBrowserApp(Navigator);
- const app = createAppContainer(Navigator);
```



```
export default app;
```

Web Nav



Web Nav

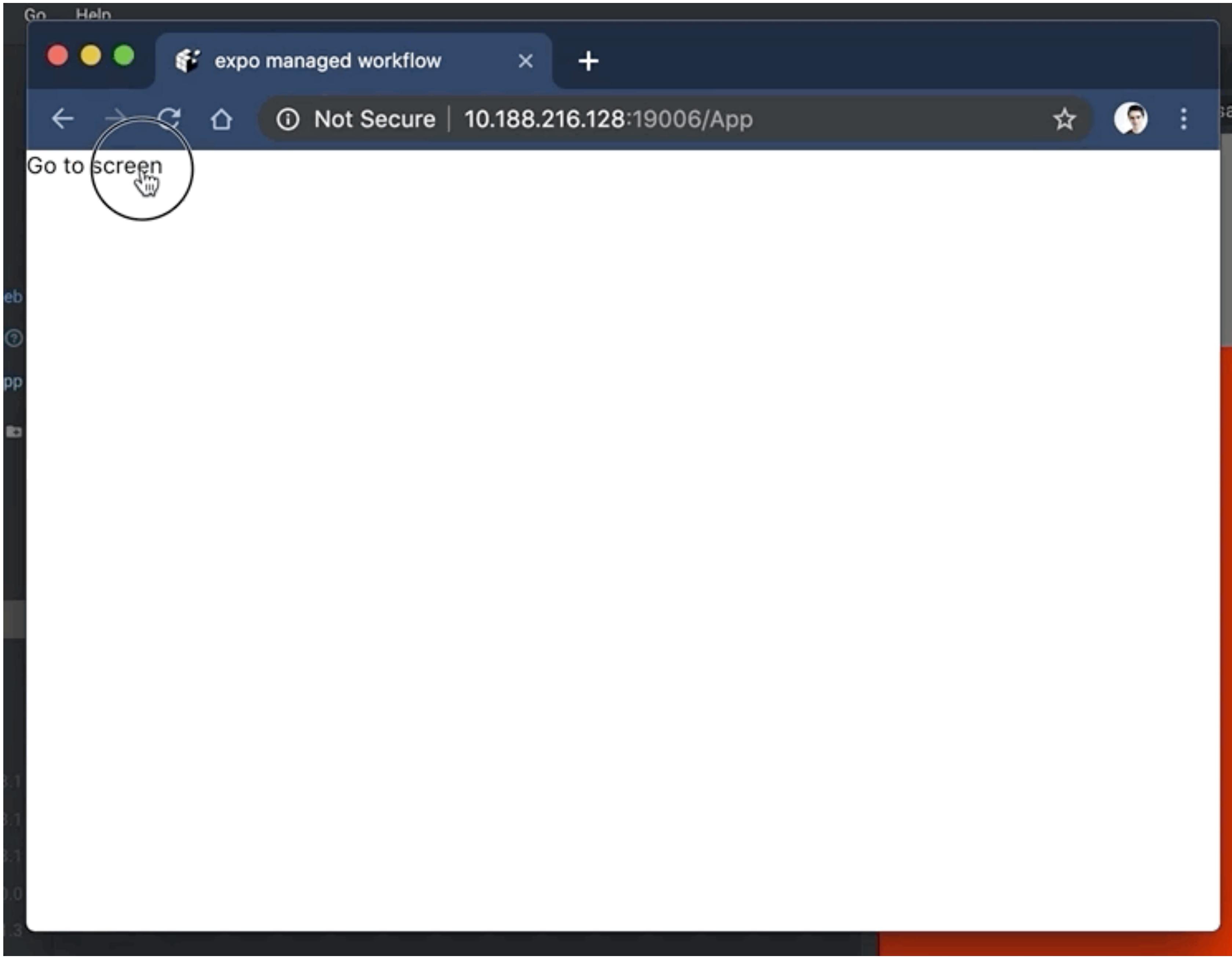
- Storing the entire stack in memory
- Generally no animations in page navigation
- Cannot Rebuild stack on URL

```
import { createSwitchNavigator } from 'react-navigation';
import { createBrowserApp } from '@react-navigation/web';

export default createBrowserApp(
  createSwitchNavigator({
    App({ navigation }) {
      return (
        <Text onPress={() => navigation.navigate('App2')}>Go to screen</Text>
      );
    },
    App2({ navigation }) {
      return (
        <Text onPress={() => navigation.navigate('App')}>
          Go to other screen
        </Text>
      );
    },
  }),
);
```



Web Nav



Navigation



```
class App extends React.Component {  
  static navigationOptions = { title: 'Hello' };  
+  static path = 'Batman';  
  
  render() {  
    return (  
      <View style={styles.container}>  
        <Text onPress={() => this.props.navigation.push('App2')}>  
          Push Screen  
        </Text>  
      </View>  
    );  
  }  
}
```



Exercise: Use navigation in the web

- Run locally - code sandbox doesn't support complex libs (well) like React Navigation yet.
- Bonus: combine navigators - make Brentikins proud 😊

Workshop

Bundling

Expo for Web

```
expo build:web
```

```
expo build:web --no-polyfill
```

```
expo build:web --no-pwa
```

Expo for Web

```
npm i -g serve  
serve ./web-build
```

Expo Web

```
▲ web
  ★ favicon.ico
  ▶ web-build
  ♦ .gitignore
  {} .watchmanconfig
  JS App.js
  {} app.json
```

Webpack Config

```
// app.json
{
  ...
  "web": {
    "build": {
      "brotli": true,
    },
    "serviceWorker": {
      ...
    }
  }
}
```

```
// webpack.config.js

const config = require('@expo/webpack-config');
const merge = require('webpack-merge');
module.exports = function(env) {
  return merge(config(env), {});
};
```

Exercise: Build an expo web project

- This exercise will help me know if building works on different computers, thanks for beta testing ;-)
- Run expo build:web
- Bonus: Add the value `expo.web.build.report: true` to the app.json and view the build report at `web-report/report.html`

Workshop

Deploying

Expo Web

- Amplify
- Surge
- Netlify
- GH Pages



Expo for Web

AWS Amplify Console

https://us-east-2.console.aws.amazon.com/amplify/home?region=us-east-2&code=4ea7a1bea19e8109bbd9#/create

Services Resource Groups Bacon Ohio Support

Step 1 Add repository service

Step 2 Add repository branch

Step 3 Configure build settings

Step 4 Review

Add repository service

Repository service providers

- GitHub 
- BitBucket 
- GitLab 
- AWS CodeCommit 

Amplify Console requires read-only access to your repository.

Feedback English (US) © 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Expo for Web

The screenshot shows the AWS Amplify Console interface for the application "amplify-expo-web-demo". A blue banner at the top indicates "A build is in progress for branch master." Below this, the navigation path is "All apps > amplify-expo-demo > master". The main section displays "Build 5" with four green checkmark icons representing the steps: Provision, Build, Deploy, and Verify. Below the steps, detailed information is provided:

Domain	Source repository	Last commit message
https://master.d33alyqvzb76gi.amplifyapp.com	https://github.com/EvanBacon/amplify-expo-demo/tree/master	Delete amplify.yml
Started at	Build duration	
4/15/2019, 7:08:29 PM	2 minutes 58 seconds	

At the bottom, there is a "Build activity" section with a "Download" button. The footer includes links for Feedback, English (US), Privacy Policy, and Terms of Use.

Exercise: Deploy the project somewhere

- Hint: I use netlify a lot. Install the netlify CLI, run `netlify deploy`
- Bonus: Use the most obscure provider

Workshop

PWA

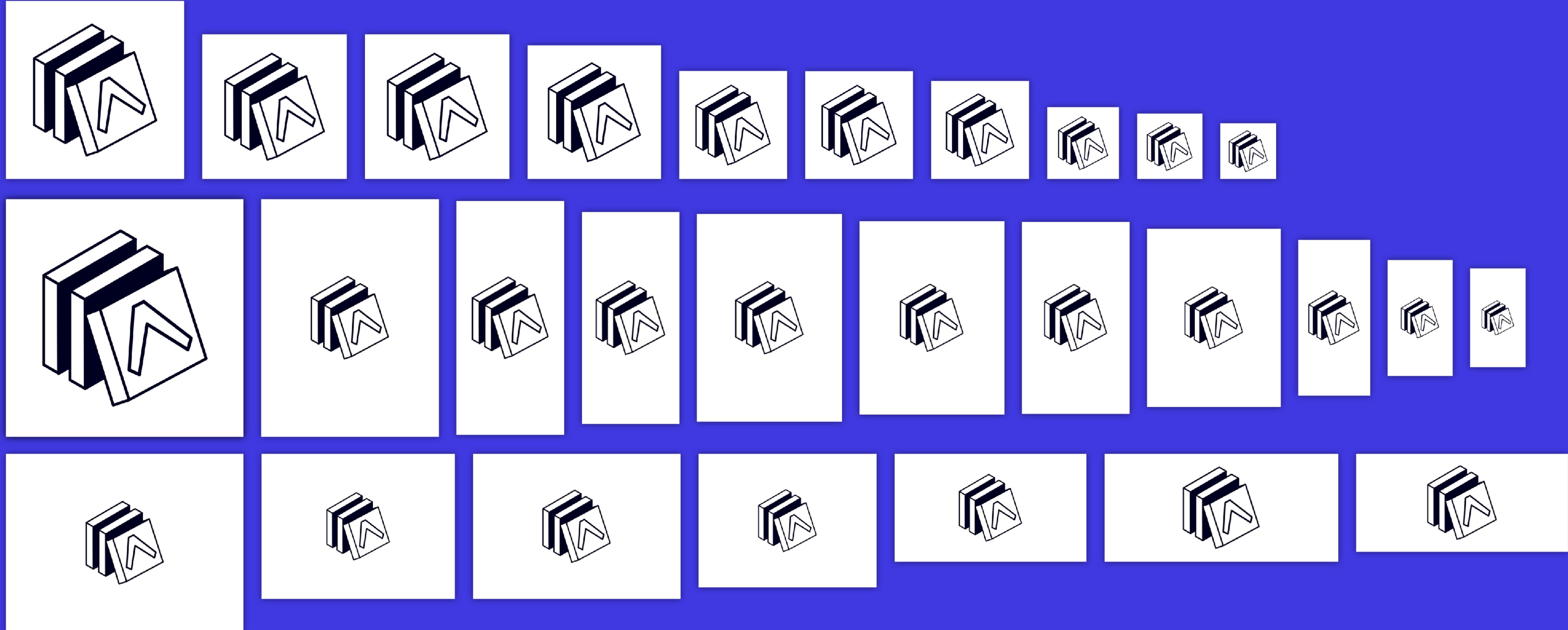
PWA

app.json -> manifest.json

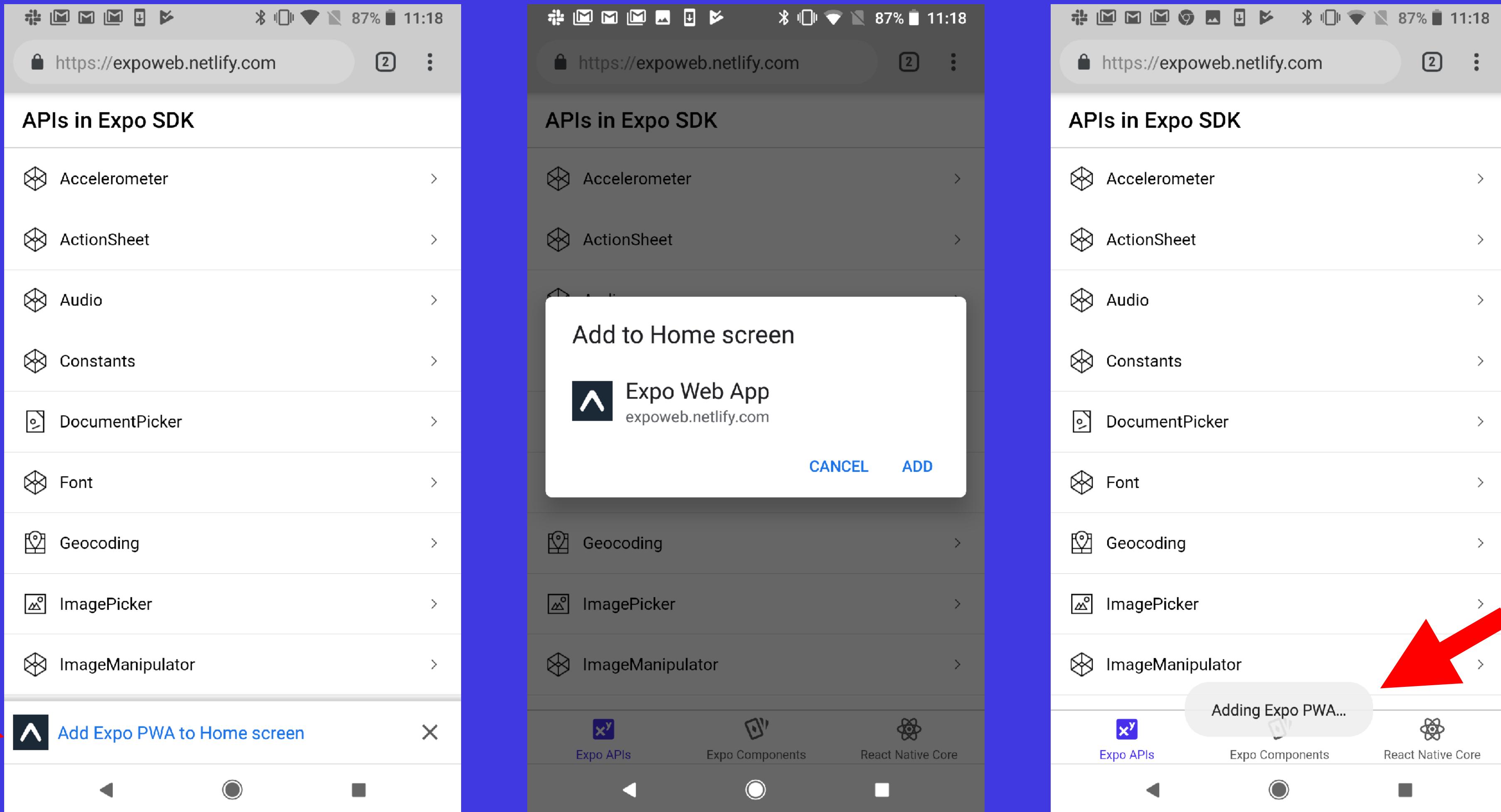
PWA

```
{  
  "background_color": "#ffffff",  
  "description": "A Neat Expo App",  
  "dir": "auto",  
  "display": "standalone",  
  "lang": "en",  
  "name": "expo managed workflow",  
  "orientation": "portrait",  
  "short_name": "expo managed workflow",  
  "start_url": ".",  
  "theme_color": "#4630EB",  
  "icons": [...]  
}
```

Icon Generation

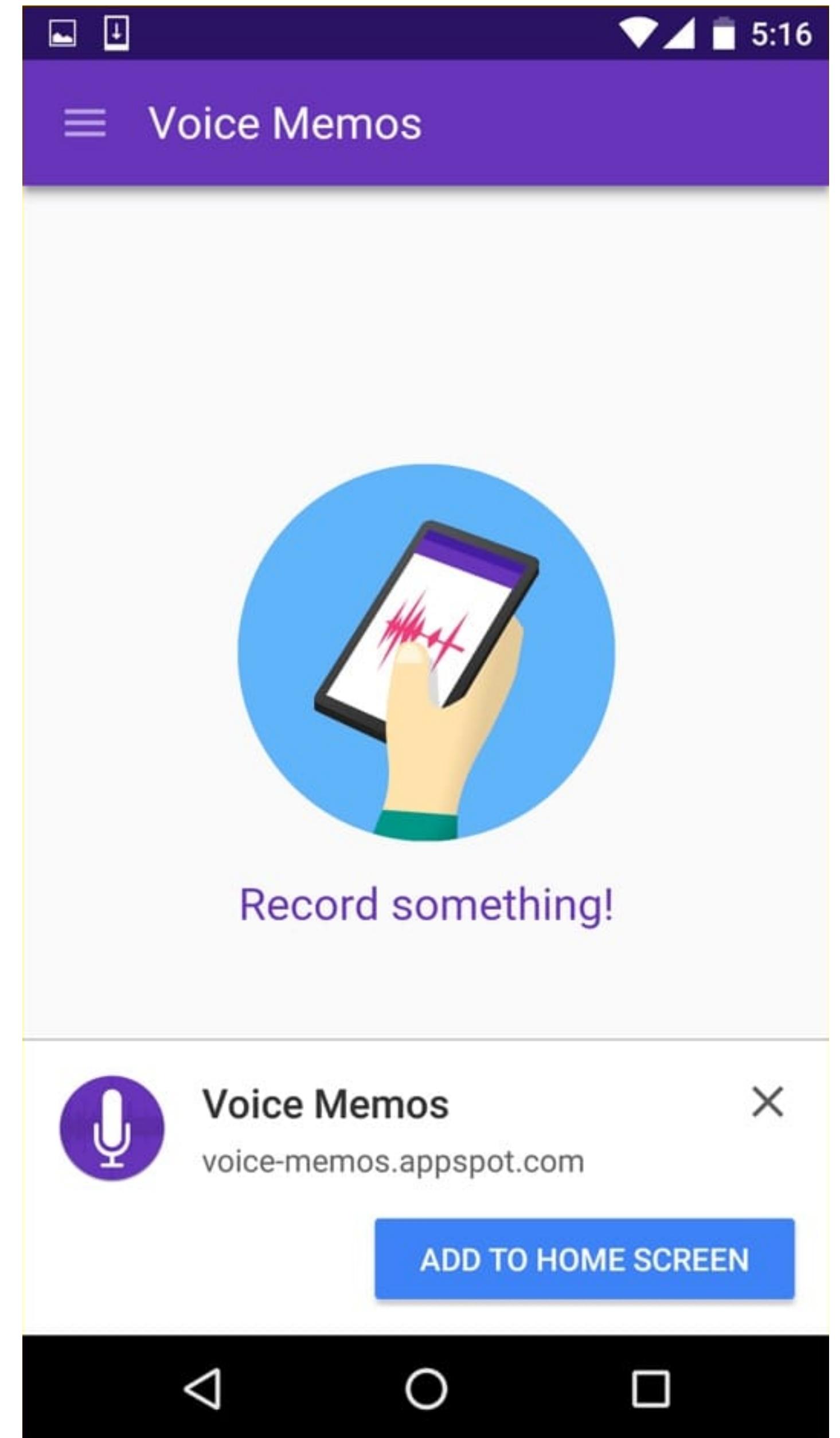


Adding PWAs

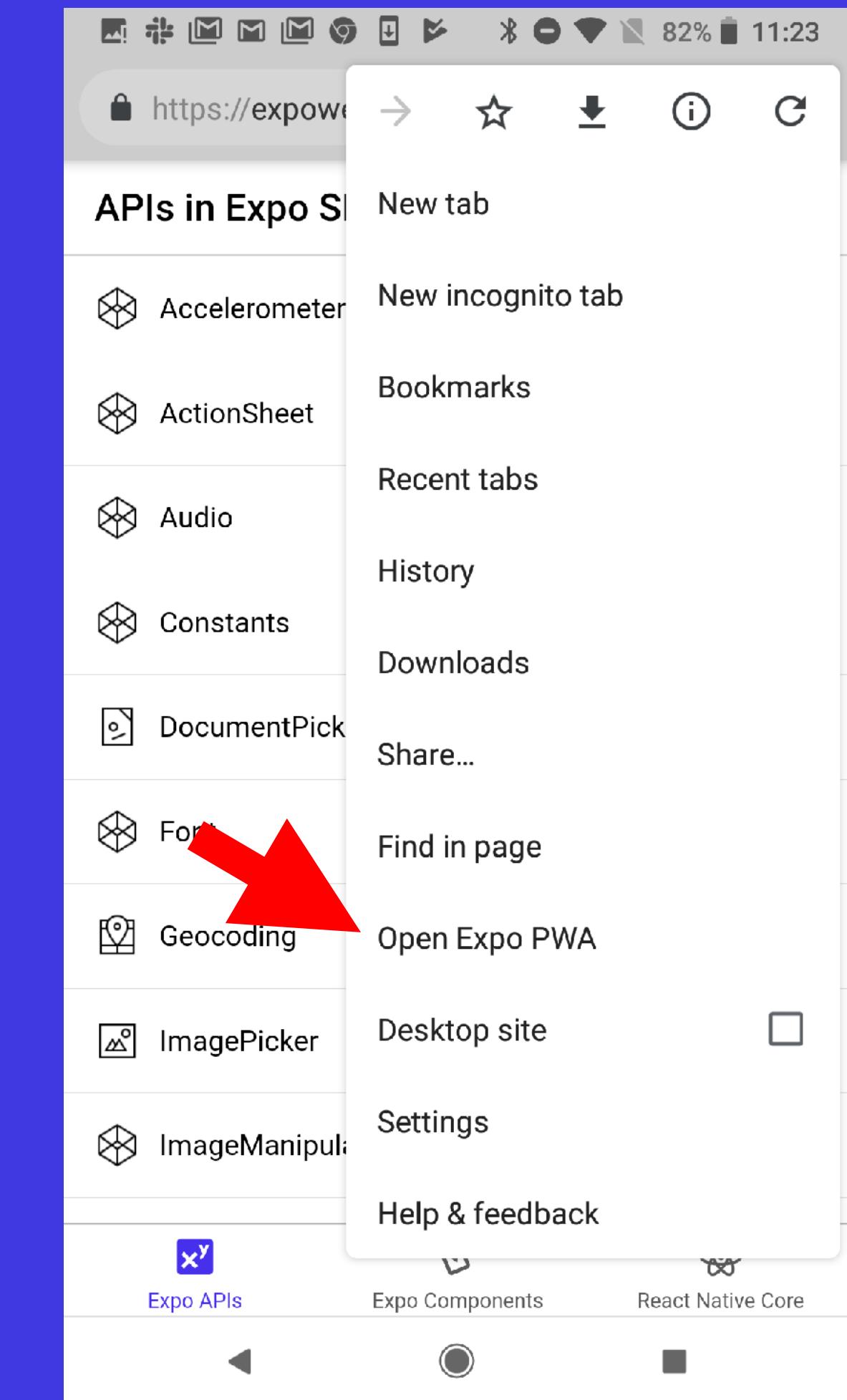
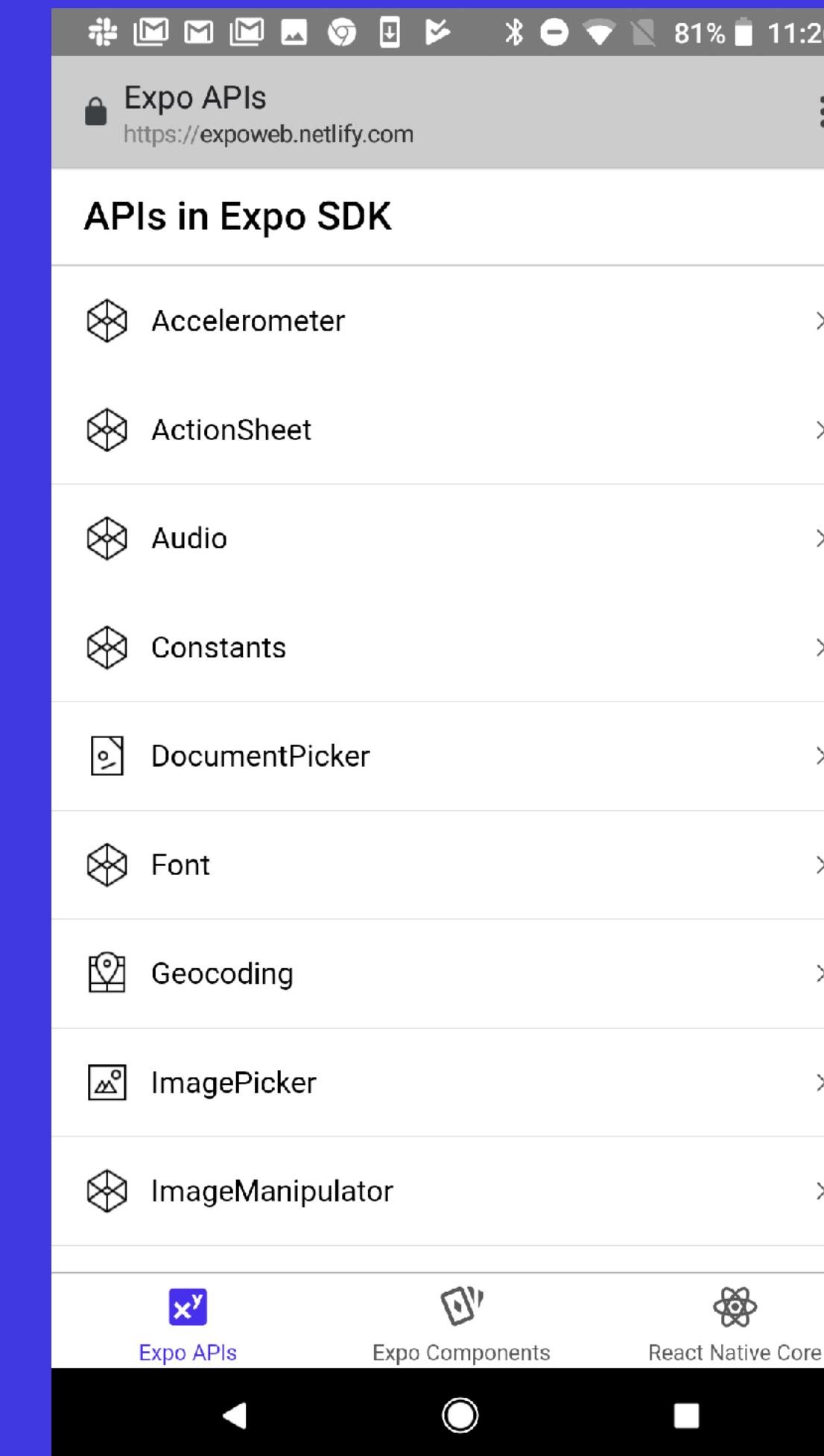
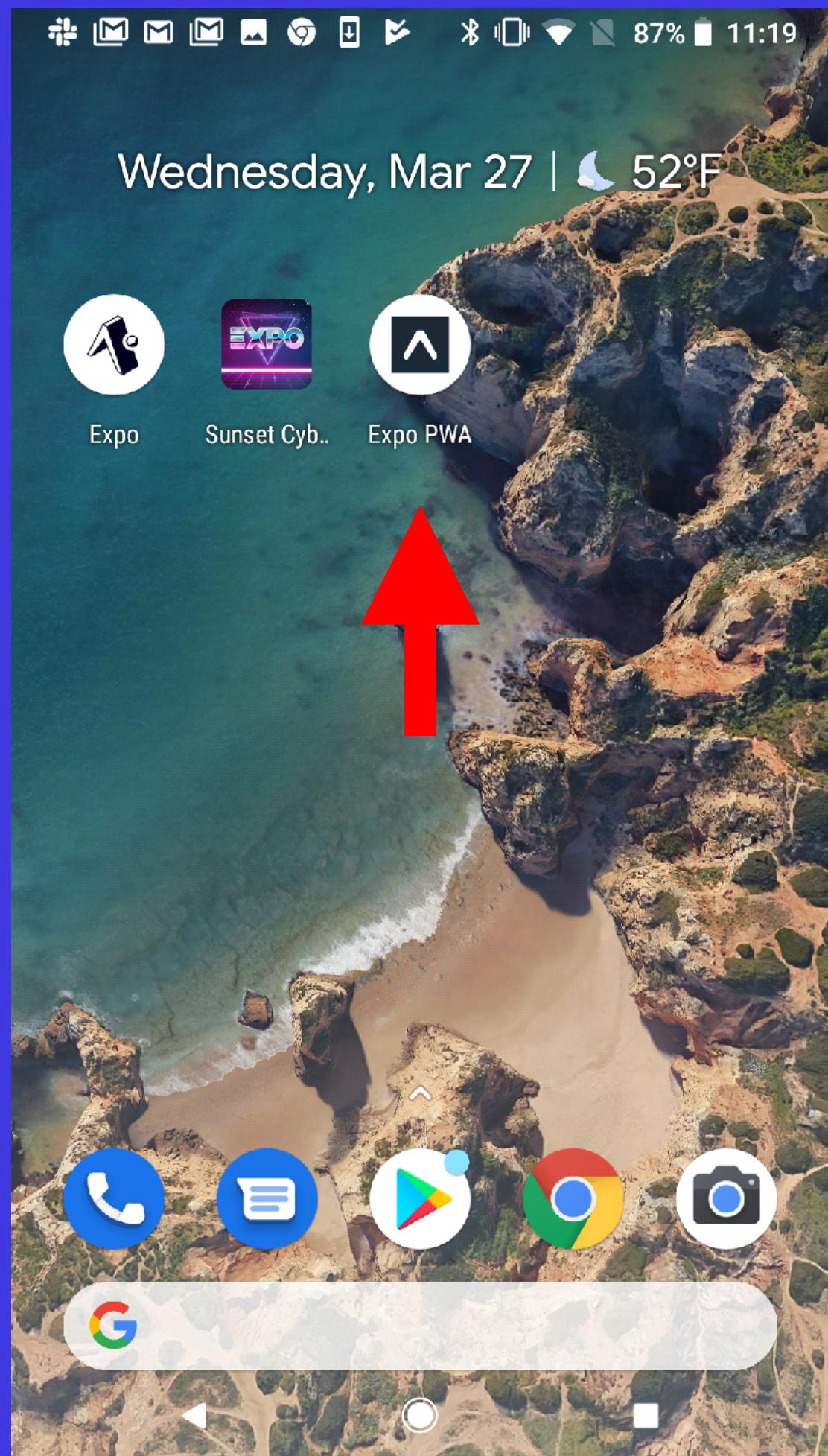


PWA

```
// app.json
{
  ...
  "android": {
    "package": "com.bacon.memos"
  }
}
```



Using PWAs



Expo Web

- Mask Icon isn't implemented yet
- Open Graph is still experimental
- Notifications API isn't implemented yet
- Safe Area View for standalone iOS PWAs
isn't finished

Exercise: Save the PWA

- Run the project locally
- visit the project on your device
- install the PWA

Workshop

TypeScript

TypeScript

TypeScript - “JavaScript that scales.”

TypeScript is a superset of JavaScript which gives you static types and powerful tooling in [Visual Studio Code](#).

TypeScript

- Expo CLI
- Webpack Config
- Babel-Preset-Expo
- Expo SDK

Expo for Web

```
yarn add -D typescript@^3.3.0 @types/expo@^32.0.0 @types/react-native@^0.57.0
```

Expo for Web

```
// tsconfig.json
{
  "compilerOptions": {
    "allowJs": false,
    "allowSyntheticDefaultImports": true,
    "noFallthroughCasesInSwitch": true,
    "experimentalDecorators": true,
    "esModuleInterop": true,
    "isolatedModules": true,
    "jsx": "react-native",
    "lib": ["es6"],
    "moduleResolution": "node",
    "noEmit": true,
    "strict": true,
    "target": "esnext"
  },
  "exclude": ["node_modules"]
}
```

Expo for Web

- App.js
- + App.tsx

You are now using
TypeScript!

Workshop

Building Web Modules

```
// Camera.js

import { NativeModulesProxy } from '@unimodules/core';

const { NativeCamera } = NativeModulesProxy;

// ✗ undefined is not an object
console.log(NativeCamera.constants);

takePicture() {
  // ✗ undefined is not an object
  return NativeCamera.takePicture();
}
```

```
// Camera.js

takePicture() {
  if (Platform.OS === 'web') {
    // Take web photo
    return {}
  }
  // Take iOS/Android photo
  return NativeCamera.takePicture();
}
```

```
// NativeCamera.js

import { NativeModulesProxy } from '@unimodules/core';

export default NativeModulesProxy.NativeCamera;
```

```
// NativeCamera.web.js

export default {
  constants: {},

  takePicture() {
    // Take picture with Media Services
  }
}
```

```
// Camera.js

import NativeCamera from './NativeCamera';

// ✅ Defined
console.log(NativeCamera.constants);

takePicture() {
  // ✅ Defined
  return NativeCamera.takePicture();
}
```

```
// 😁
await Feature.isAvailableAsync()

// 😁
const { status } await Permissions.getAsync(/*...*/);
// status === 'unavailable'

// 😊
Feature.doThing();
// throws UnavailabilityError
// Feature.doThing() is not supported on web
```

Exercise: Do my job for me

- Hint: <https://github.com/expo/expo/labels/project%3A%20web>



λ Expo