

Remix and a native app?

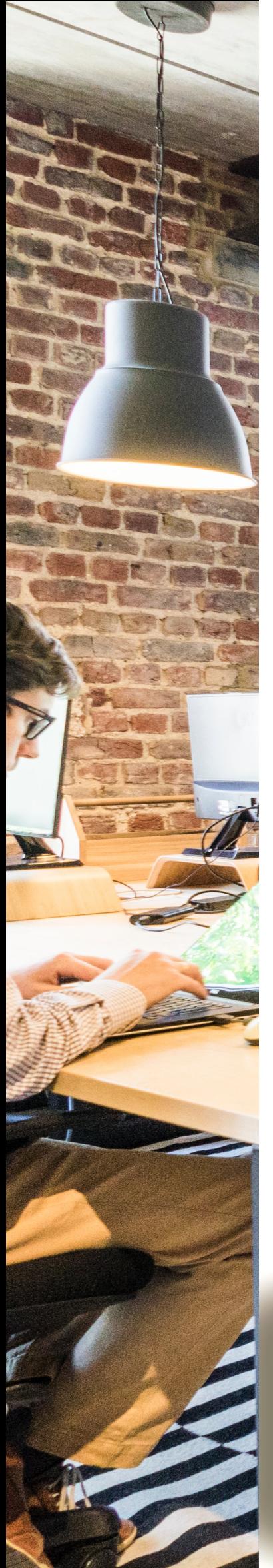
Why not both just one.

Oscar Newman

Remix Conf 2023



GROWTH



“How hard is it to make an iPhone app”

Your boss, probably



 **React Native Upgrade Helper**

Star 3,008 ⚠️ 🛡️

What's your app name? RnDiffApp

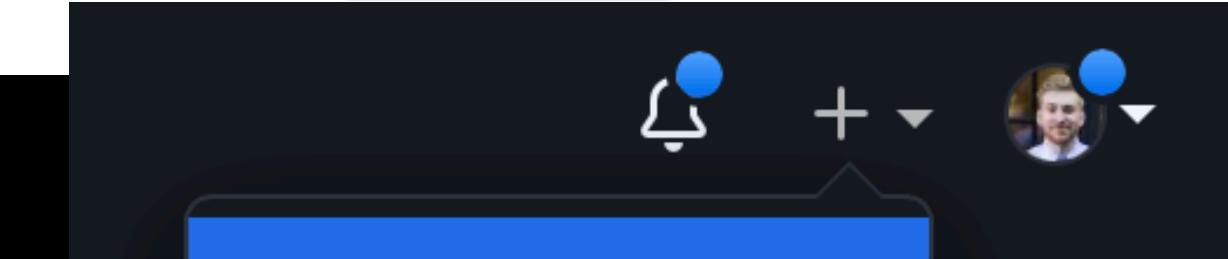
What's your current react-native version? 0.69.9

To which version would you like to upgrade? 0.71.7

Show me how to upgrade!

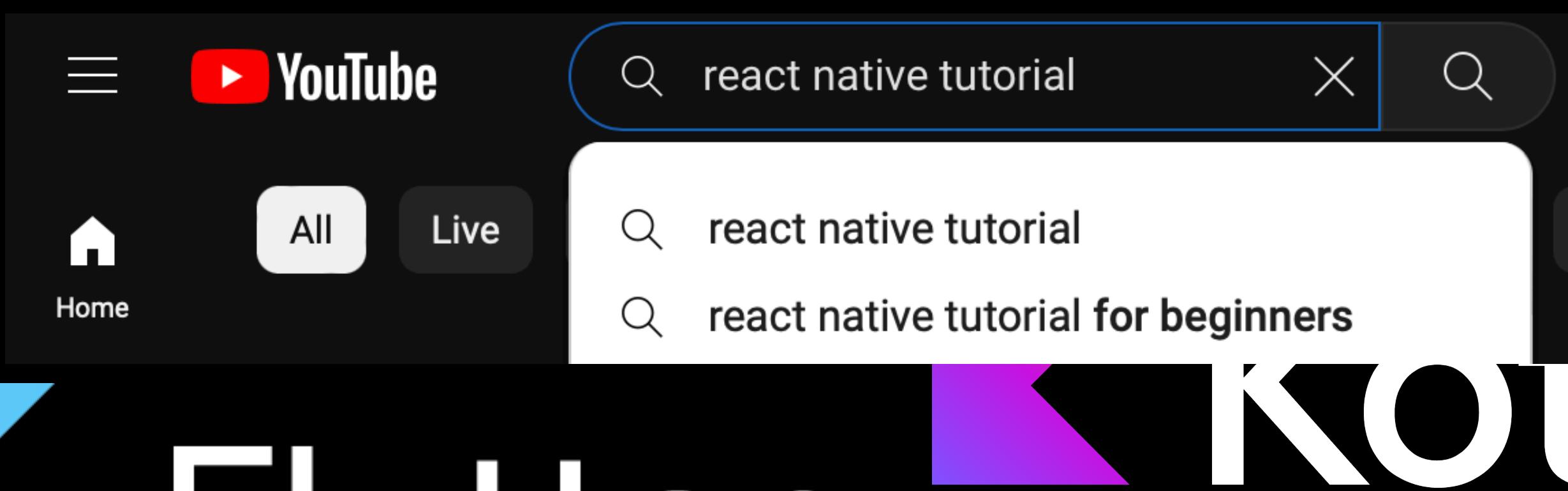
Status ! Failed to create provisioning profile.
There are no devices registered in your account on the developer website. Plug in and select a device to have Xcode register it.

Try Again



- New repository
- Import repository
- New codespace
- New gist
- New organization
- New project

Scanning

 YouTube search interface showing the query "react native tutorial". Below the search bar, there are suggestions: "react native tutorial" and "react native tutorial for beginners".



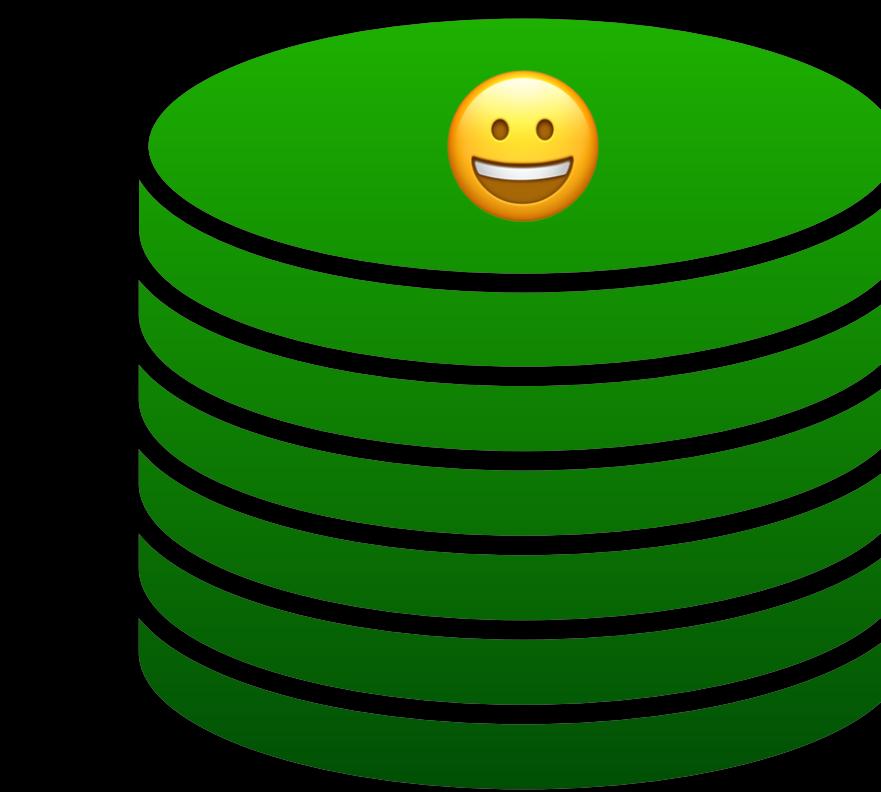
Flutter

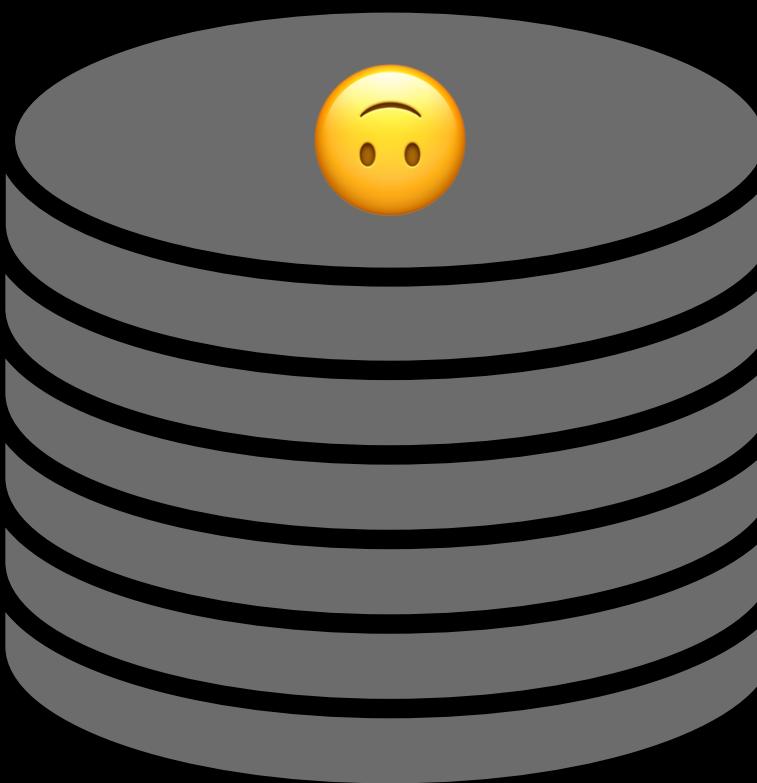
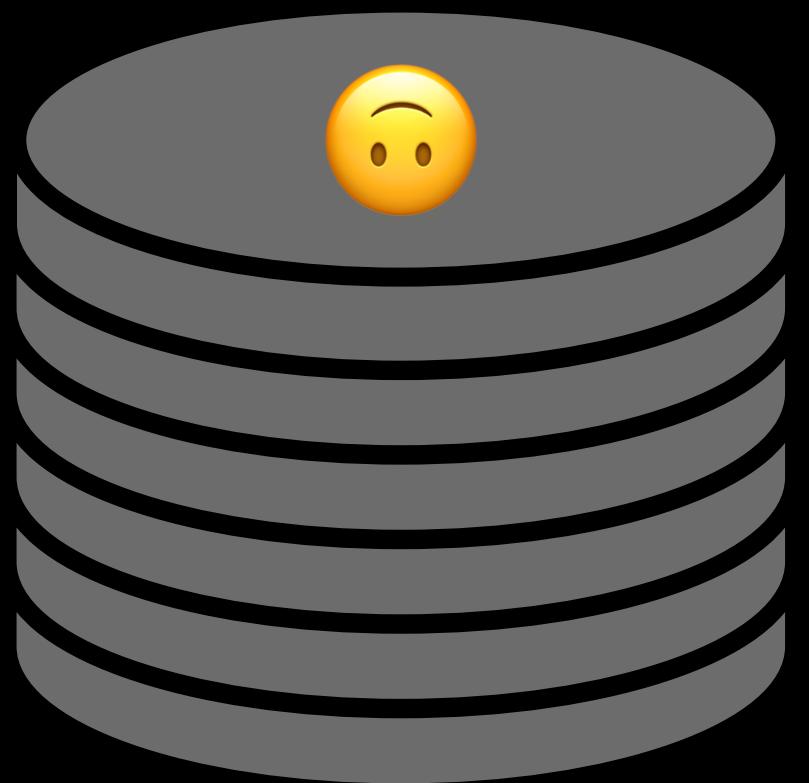


Kotlin



Expo in iPhone app”





^ Expo

```
export default function App() {  
  // secret sauce  
}
```

^ Expo

```
|-----|  
| export default function App() { |  
|   return ( |  
|     <WebView |  
|       source={{ |  
|         uri: "https://solvhealth.com" |  
|       }} |  
|     > |  
|   ); |  
| } |  
|-----|
```

PAUSE III

PAUSE III

Solv. Get care Visits Insurance Profile ON Oscar

Search Location Time Patient
Urgent Care San Francisco, CA ASAP Adult

All visits Video visits

Accepts Blue Cross Blue Shield More filters

20 instant-book locations

BEST MATCH

Dignity Health-GoHealth Urgent Care, Castro
Urgent Care • Open until 8pm
0.92 mi • 2288 Market St, San Francisco
2.0 ★★★★☆ (1 rating)
Accepts kids • Short wait time
✓ In-network with Blue Cross Blue Shield

Dignity Health-GoHealth Urgent Care, Market Street
Urgent Care • Open until 8pm
0.62 mi • 1600 Market St, San Francisco
4.8 ★★★★★ (2841 ratings)
Accepts kids • Highly rated
✓ In-network with Blue Cross Blue Shield

TODAY TODAY

10:37 am 10:52 am 11:30 am 11:45 am 12:00 pm
12:30 pm 12:45 pm 1:30 pm 1:45 pm 2:00 pm

View more availability

Sort by: Soonest Available

Map of San Francisco and surrounding areas showing 20 instant-book locations marked with pink pins. Labels on the map include Larkspur, Mill Valley, Paradise Cay, Richmond, Berkeley, Oakland, Alameda, San Francisco Bay, Daly City, South San Francisco, San Bruno, and Pacifica.

Remix

ST

Solv ▾



Track deductible spending and unlock hidden benefits

Snap a photo of your card to view your benefits and track your deductible.

[Add Insurance](#)

Get care



Visits



Insurance



Profile

ST

Solv ▾



Track deductible spending and unlock hidden benefits

Snap a photo of your card to view your benefits and track your deductible.

[Add Insurance](#)

Get care



Visits



Insurance



Profile

Single Page App

User taps link

New page renders immediately
(with spinners)

Data fetching begins

Data loads

1:27

61

ST

Solv



Track deductible spending and
unlock hidden benefits

Snap a photo of your card to view your benefits
and track your deductible.

Add Insurance



Get care



Visits

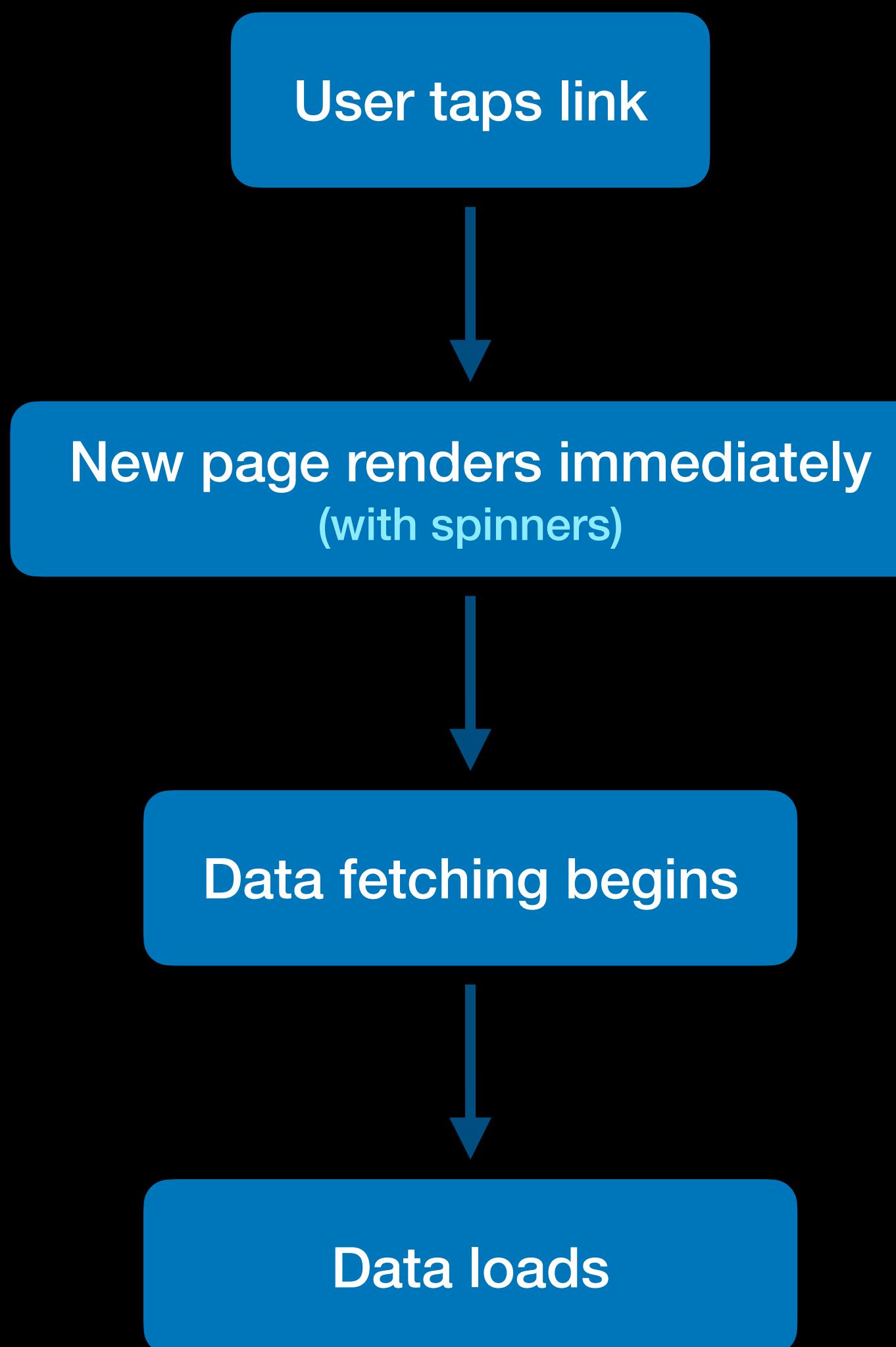


Insurance

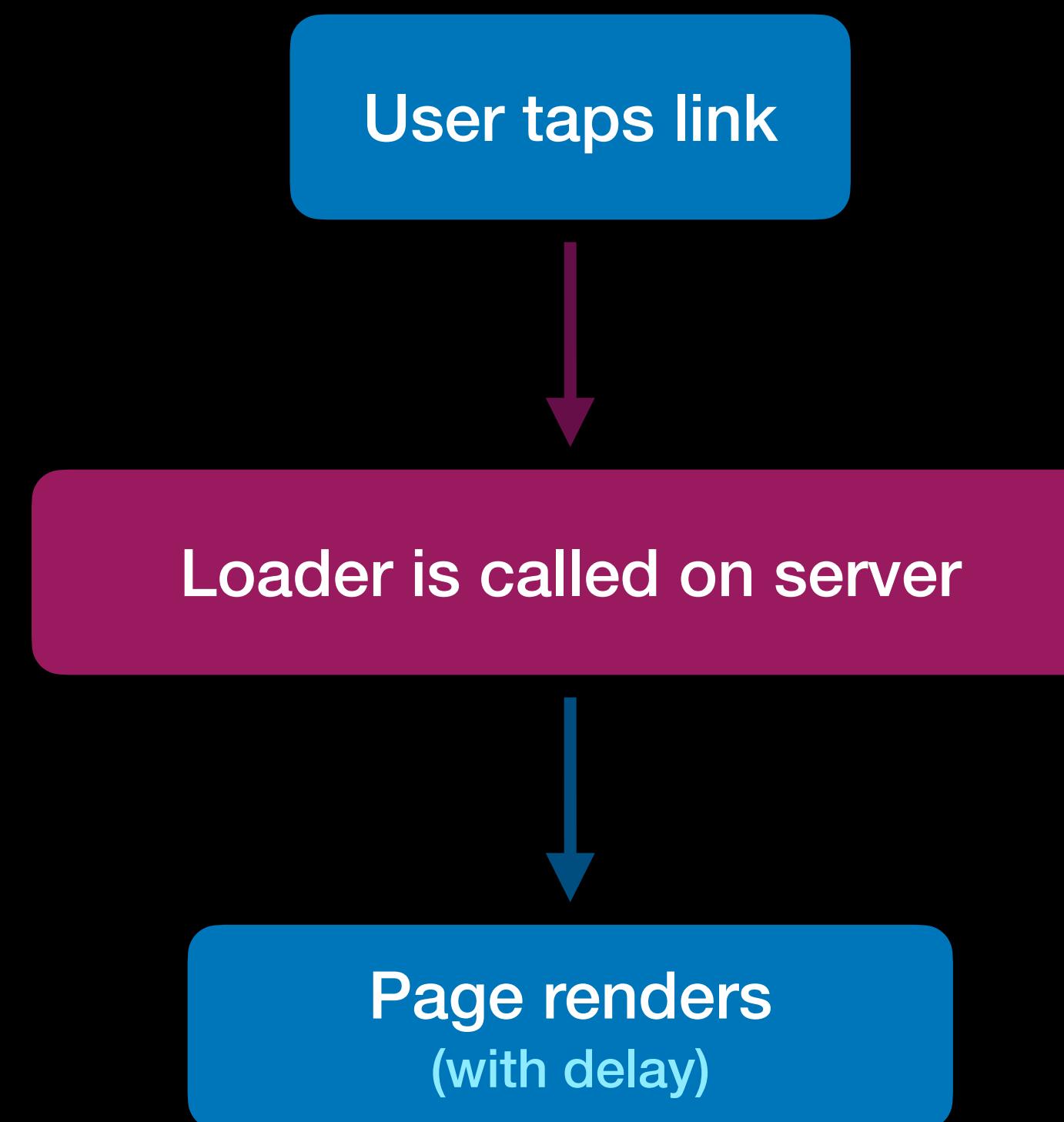


Profile

Single Page App



Remix App



Expectations are different on the
web and native platforms

Not Secure — remix.tunes/playlists

New +

Playlists

- Music >
- Movies >
- TV Shows >
- Audiobooks >
- 90's Music >
- Audiobooks >
- Movies >
- Music >
- Music Videos >
- TV Shows >
- Brazilian Music >
- Classical >
- Classical 101 - Deep Cuts >

remix.tunes

Playlists

Albums

Tom Cook

Not Secure — remix.tunes/playlists

New +

Playlists

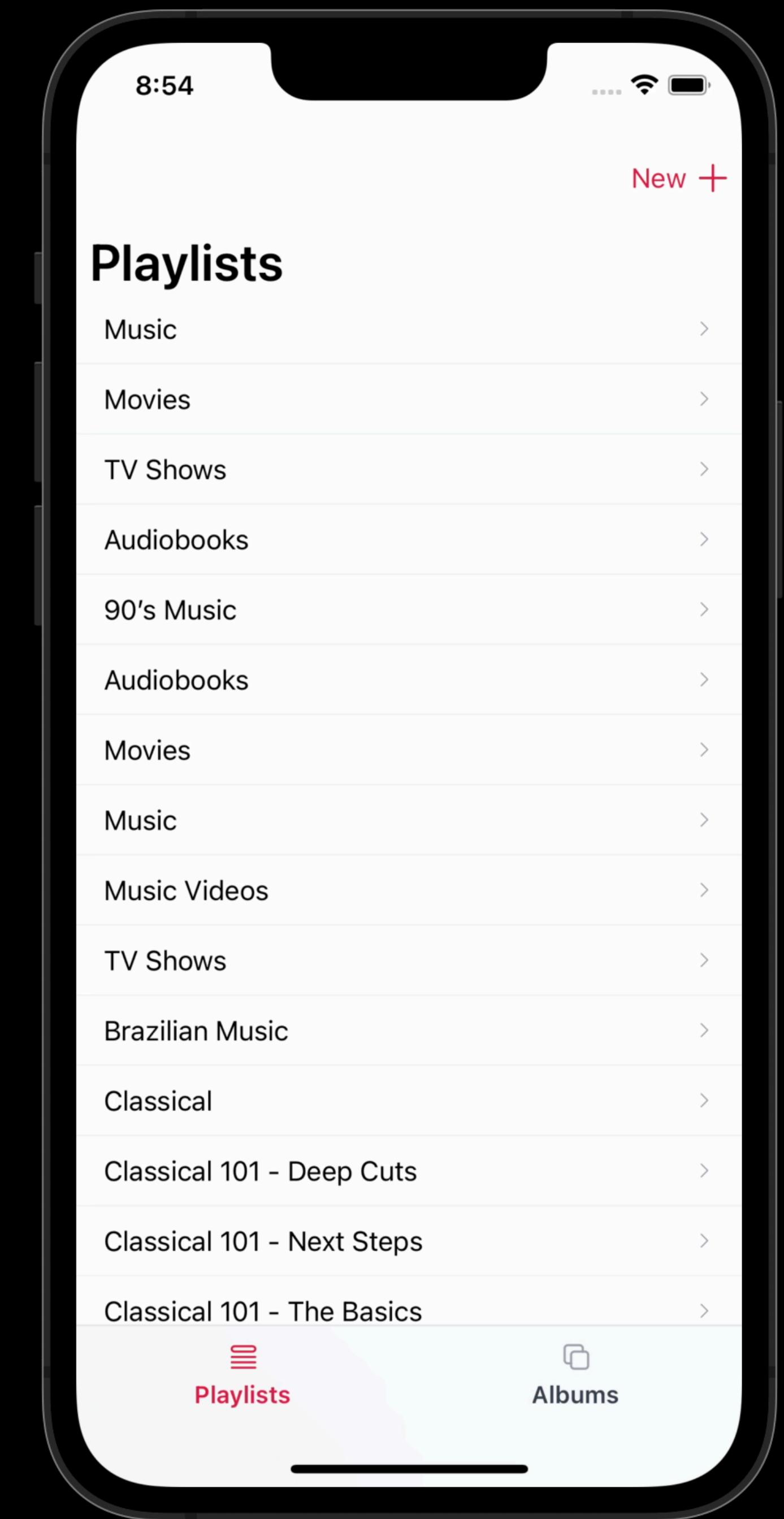
- Music >
- Movies >
- TV Shows >
- Audiobooks >
- 90's Music >
- Audiobooks >
- Movies >
- Music >
- Music Videos >
- TV Shows >
- Brazilian Music >
- Classical >
- Classical 101 - Deep Cuts >

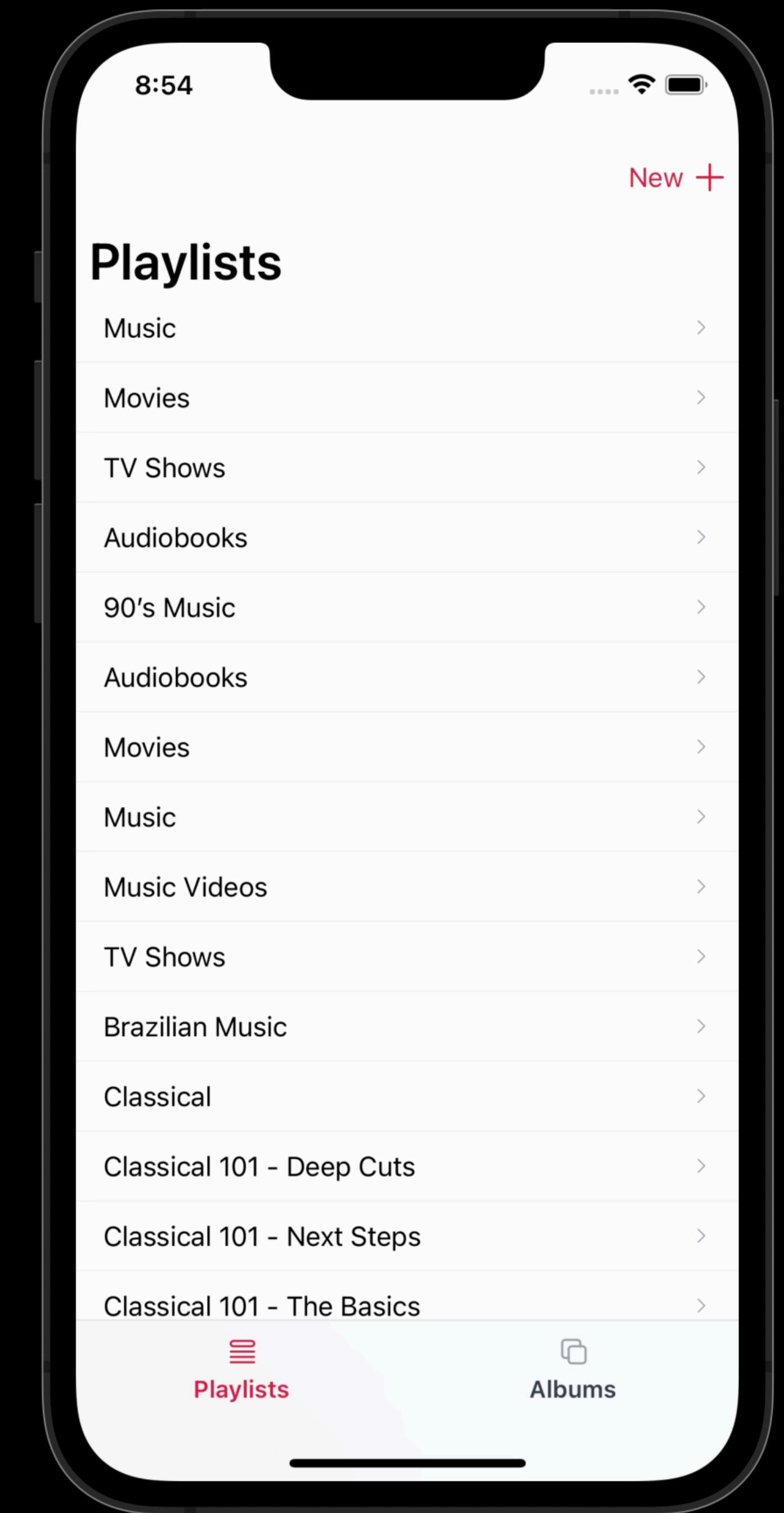
remix.tunes

Playlists

Albums

Tom Cook





```
export async function loader({ params }) {
  const playlist = await getPlaylist(params.id);
  return { playlist };
}

export default function PlaylistPage() {
  const { playlist } = useLoaderData();

  return <PlaylistDetails playlist={playlist} />;
}
```

```
export async function loader({ params }) {
  const playlist = getPlaylist(params.id);
  return defer({ playlist });
}

export default function PlaylistPage() {
  const { playlist } = useLoaderData();

  return <PlaylistDetails playlist={playlist} />;
}
```

```
export async function loader({ params }) {
  const playlist = getPlaylist(params.id);
  return defer({ playlist });
}

export default function PlaylistPage() {
  const { playlist } = useLoaderData();

  return (
    <Suspense fallback=<Loading />>
      <Await resolve={playlist}>
        {(playlist) => <PlaylistDetails playlist={playlist} />}
      </Await>
    </Suspense>
  );
}
```

How do we wait for data on the web, but defer in the app?

< Playlists

Music

1. For Those About To Rock (We Salute You)
AC/DC
2. Balls to the Wall
Accept
3. Fast As a Shark
Accept
4. Restless and Wild
Accept
5. Princess of the Dawn
Accept
6. Put The Finger On You
AC/DC
7. Let's Get It Up
AC/DC
8. Inject The Venom
AC/DC
9. Snowballed
AC/DC
10. Evil Walks
AC/DC
11. C.O.D.

Playlists Albums

```
export async function loader({ params }) {  
  const playlist = getPlaylist(params.id);  
  return defer({ playlist });  
}  
  
export default function PlaylistPage() {  
  const { playlist } = useLoaderData();  
  
  return (  
    <Suspense fallback={<Loading />}>  
      <Await resolve={playlist}>  
        {(playlist) => <PlaylistDetails playlist={playlist} />}  
      </Await>  
    </Suspense>  
  );  
}
```

< Playlists

Music

1. For Those About To Rock (We Salute You)
AC/DC
2. Balls to the Wall
Accept
3. Fast As a Shark
Accept
4. Restless and Wild
Accept
5. Princess of the Dawn
Accept
6. Put The Finger On You
AC/DC
7. Let's Get It Up
AC/DC
8. Inject The Venom
AC/DC
9. Snowballed
AC/DC
10. Evil Walks
AC/DC
11. C.O.D.

Playlists Albums

```
export async function loader({ params }) {  
  const playlist = getPlaylist(params.id);  
  return defer({ playlist });  
}  
  
export default function PlaylistPage() {  
  const { playlist } = useLoaderData();  
  
  return (  
    <Suspense fallback={<Loading />}>  
      <Await resolve={playlist}>  
        {(playlist) => <PlaylistDetails playlist={playlist} />}  
      </Await>  
    </Suspense>  
  );  
}
```

Not Secure — remix.tunes/play

< Playlists

Music

1. For Those About To Rock (We Salute You)
AC/DC
2. Balls to the Wall
Accept
3. Fast As a Shark
Accept
4. Restless and Wild
Accept
5. Princess of the Dawn
Accept
6. Put The Finger On You
AC/DC
7. Let's Get It Up
AC/DC
8. Inject The Venom
AC/DC
9. Snowballed
AC/DC
10. Evil Walks
AC/DC
11. C.O.D.

Playlists Albums

```
export async function loader({ params }) {  
  const playlist = await getPlaylist(params.id);  
  return defer({ playlist });  
}  
  
export default function PlaylistPage() {  
  const { playlist } = useLoaderData();  
  
  return (  
    <Suspense fallback={<Loading />}>  
      <Await resolve={playlist}>  
        {(playlist) => <PlaylistDetails playlist={playlist} />}  
      </Await>  
    </Suspense>  
  );  
}
```

< Playlists

Music

1. For Those About To Rock (We Salute You)
AC/DC
2. Balls to the Wall
Accept
3. Fast As a Shark
Accept
4. Restless and Wild
Accept
5. Princess of the Dawn
Accept
6. Put The Finger On You
AC/DC
7. Let's Get It Up
AC/DC
8. Inject The Venom
AC/DC
9. Snowballed
AC/DC
10. Evil Walks
AC/DC
11. C.O.D.

Playlists Albums

```
export async function loader({ params }) {  
  const playlist = await getPlaylist(params.id);  
  return defer({ playlist });  
}  
  
export default function PlaylistPage() {  
  const { playlist } = useLoaderData();  
  
  return (  
    <Suspense fallback={<Loading />}>  
      <Await resolve={playlist}>  
        {(playlist) => <PlaylistDetails playlist={playlist} />}  
      </Await>  
    </Suspense>  
  );  
}
```

```
export function isNative(request) {  
    const userAgent = request.headers.get("User-Agent");  
    return !!userAgent?.match(/my-native-app/i);  
}
```

```
export function isNative(request) {
  const userAgent = request.headers.get("User-Agent");
  return !!userAgent?.match(/my-native-app/i);
}

export async function loader({ params }) {
  const playlist = getPlaylist(params.id);
  return defer({ playlist });
}
```

```
export function isNative(request) {
  const userAgent = request.headers.get("User-Agent");
  return !!userAgent?.match(/my-native-app/i);
}

export async function loader({ params }) {
  const playlist = getPlaylist(params.id);

  if (!isNative(request)) {
    await playlist;
  }

  return defer({ playlist });
}
```

A screenshot of a web browser window titled "Not Secure — remix.tunes/playlists". The page displays a list of playlists categorized under "Playlists". The categories include Music, Movies, TV Shows, Audiobooks, 90's Music, Audiobooks, Movies, Music, Music Videos, TV Shows, Brazilian Music, Classical, Classical 101 - Deep Cuts, Classical 101 - Next Steps, and Classical 101 - The Basics. A "New +" button is located at the top right of the list. On the left side of the browser window, there is a sidebar with icons for "remix.tunes" (music note), "Playlists" (list icon), and "Albums" (square icon). At the bottom left, there is a user profile picture of a man with glasses and the name "Tom Cook".

- Music >
- Movies >
- TV Shows >
- Audiobooks >
- 90's Music >
- Audiobooks >
- Movies >
- Music >
- Music Videos >
- TV Shows >
- Brazilian Music >
- Classical >
- Classical 101 - Deep Cuts >
- Classical 101 - Next Steps >
- Classical 101 - The Basics >

A screenshot of an iPhone displaying a list of playlists. The screen shows a "Playlists" section with various categories: Music, Movies, TV Shows, Audiobooks, 90's Music, Audiobooks, Movies, Music, Music Videos, TV Shows, Brazilian Music, Classical, Classical 101 - Deep Cuts, Classical 101 - Next Steps, and Classical 101 - The Basics. At the top right, there is a "New +" button. At the bottom, there is a navigation bar with "Playlists" selected (indicated by a red background) and "Albums" as an option. The phone's status bar shows the time as 8:34 and signal strength.

A screenshot of a web browser window titled "Not Secure — remix.tunes/playlists". The browser interface includes standard controls like back, forward, and search. On the left, there's a sidebar with a user profile picture of "Tom Cook" and a "Playlists" tab which is currently selected, indicated by a red background. Below the sidebar is a main content area with a heading "Playlists" and a "New +" button. The main content lists various categories: Music, Movies, TV Shows, Audiobooks, 90's Music, Audiobooks, Movies, Music, Music Videos, TV Shows, Brazilian Music, Classical, Classical 101 - Deep Cuts, Classical 101 - Next Steps, and Classical 101 - The Basics.

Category
Music
Movies
TV Shows
Audiobooks
90's Music
Audiobooks
Movies
Music
Music Videos
TV Shows
Brazilian Music
Classical
Classical 101 - Deep Cuts
Classical 101 - Next Steps
Classical 101 - The Basics

A screenshot of a mobile device screen showing a "Playlists" list. The top status bar indicates the time is 8:34 and shows signal, battery, and Wi-Fi icons. The main content area has a heading "Playlists" and a "New +" button. The list contains items: Music, Movies, TV Shows, Audiobooks, 90's Music, Audiobooks, Movies, Music, Music Videos, TV Shows, Brazilian Music, Classical, Classical 101 - Deep Cuts, Classical 101 - Next Steps, and Classical 101 - The Basics. At the bottom of the screen, there are navigation tabs for "Playlists" (which is highlighted in red) and "Albums".

Category
Music
Movies
TV Shows
Audiobooks
90's Music
Audiobooks
Movies
Music
Music Videos
TV Shows
Brazilian Music
Classical
Classical 101 - Deep Cuts
Classical 101 - Next Steps
Classical 101 - The Basics

A screenshot of a web browser window titled "Not Secure — remix.tunes/playlists". The left sidebar shows navigation icons and a user profile for "Tom Cook". The main content area displays a list of playlists under the heading "Playlists". The list includes categories like Music, Movies, TV Shows, Audiobooks, and various genre-specific playlists such as "90's Music", "Brazilian Music", and "Classical". A "New +" button is located at the top right of the list.

- Music >
- Movies >
- TV Shows >
- Audiobooks >
- 90's Music >
- Audiobooks >
- Movies >
- Music >
- Music Videos >
- TV Shows >
- Brazilian Music >
- Classical >
- Classical 101 - Deep Cuts >
- Classical 101 - Next Steps >
- Classical 101 - The Basics >

A screenshot of an iPhone displaying a list of playlists. The screen shows a "Playlists" section with a "New +" button at the top right. Below it is a list of items, many of which are repeated from the web browser's list. The bottom of the screen features a navigation bar with "Playlists" and "Albums" tabs, and a small cursor icon is visible near the bottom center.

- Music
- Movies
- TV Shows
- Audiobooks
- 90's Music
- Audiobooks
- Movies
- Music
- Music Videos
- TV Shows
- Brazilian Music
- Classical
- Classical 101 - Deep Cuts
- Classical 101 - Next Steps
- Classical 101 - The Basics

defer-if

`defer-if` is a utility package that extends the functionality of Remix's `defer` method, allowing you to easily conditionally defer or not defer data based on some programmatic criteria.

Installation

```
npm install defer-if
```

Usage

This is a quick demo showing how to use `defer-if` to only defer data when the user agent is a mobile device.

```
import { deferIf } from "defer-if";
import { isMobileUserAgent } from "../your-utils";

export function loader({ request }) {
  const data = {
    value1: await fetchSomething(), // this will always block (never
    value2: "This is a static value",
    value3: fetchSomethingElse(), // this will either block or defer
  };

  // Using deferIf
  return await deferIf(
    data,
    () => isMobileUserAgent(request), // Predicate function or boolean
  );
}
```

Install

```
> npm i defer-if
```

Repository

 github.com/oscarnewman/defer-if

Homepage

 [github.com/oscarnewman/defer-if#read...](https://github.com/oscarnewman/defer-if#readme)

Weekly Downloads

175

Version

1.0.4

License

ISC

Unpacked Size

185 kB

Total Files

14

Issues

0

Pull Requests

1

Last publish

14 hours ago

Collaborators



>-Try on RunKit

```
export function isNative(request) {  
    const userAgent = request.headers.get("User-Agent");  
    return !!userAgent?.match(/my-native-app/i);  
}
```

```
export async function loader({ params }) {  
    const playlist = getPlaylist(params.id);  
  
    if (!isNative(request)) {  
        await playlist;  
    }  
  
    return defer({ playlist });  
}
```

```
import { deferIf } from "defer-if";

export function isNative(request) {
    const userAgent = request.headers.get("User-Agent");
    return !!userAgent?.match(/my-native-app/i);
}

export async function loader({ params }) {
    const playlist = getPlaylist(params.id);

    return deferIf({ playlist }, isNative(request));
}
```

What else can we do?

Caching

```
// routes/playlists.$id.js
export async function loader({ params }) {
  const { id } = params;
  const playlist = cachified(`playlist-${id}`, getPlaylist(id), {
    ttl: 30,
  });

  return deferIf({ playlist }, isNative(request));
}
```

Caching + Prefetching

```
// routes/playlists.js
export async function loader() {
  const playlists = await getPlaylists();

  return defer({ playlists });
}
```

Caching + Prefetching

```
// routes/playlists.js
export async function loader() {
  const playlists = await getPlaylists();

  // async function we don't await
  fetchAndHydratePlaylistCache(playlists);

  return defer({ playlists });
}
```

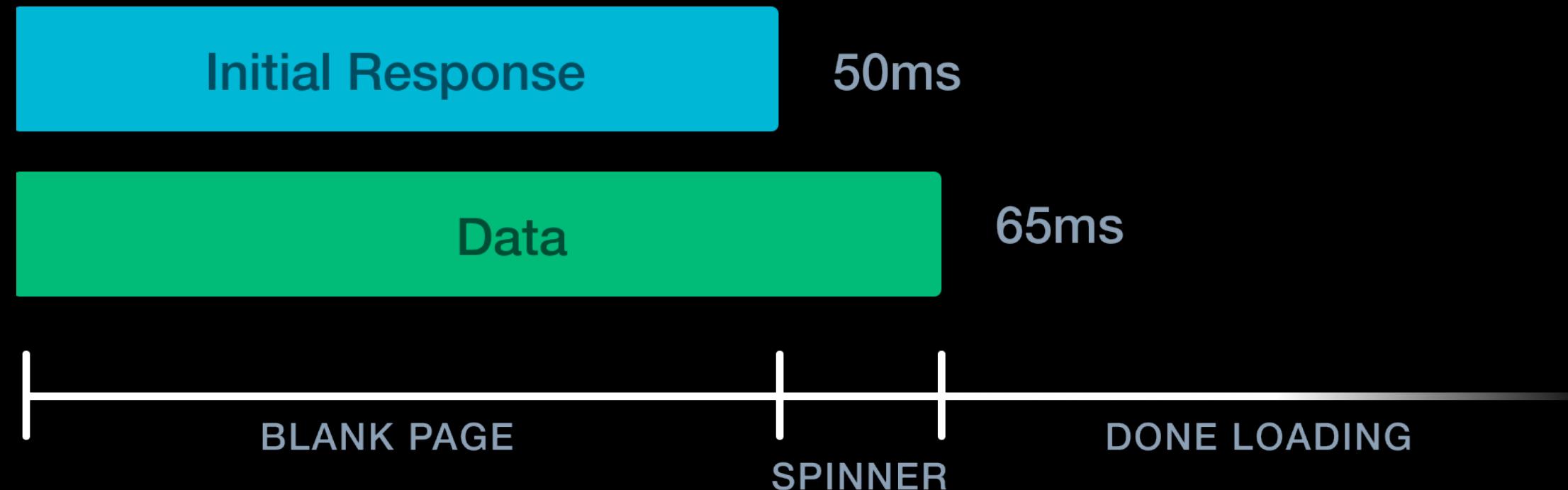
Cache miss



Cache miss



Cache Hit



Promise.race()

“The `Promise.race()` static method takes an iterable of promises as input and returns a single `Promise`. This returned promise settles with the eventual state of the first promise that settles.”

– MDN

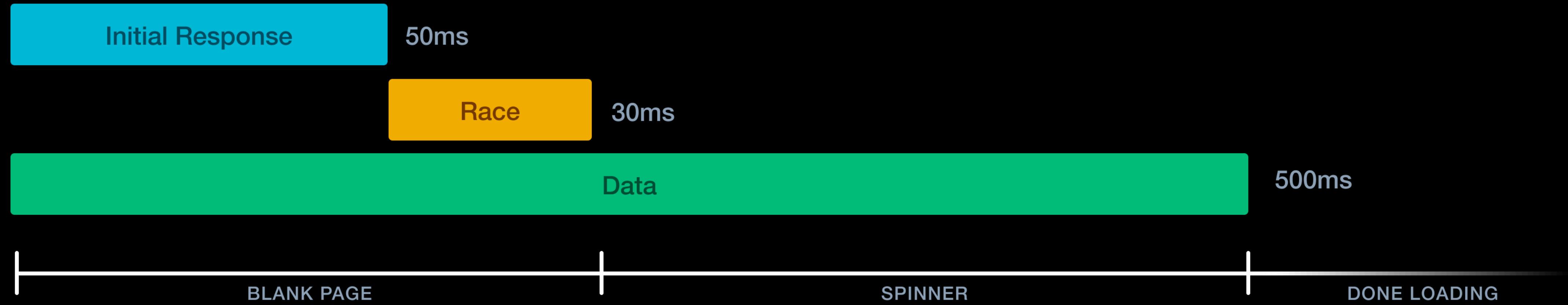
```
export async function loader() {  
  const playlists = getPlaylists();  
  
  return defer({ playlists });  
}
```

```
export async function loader() {
  const playlists = getPlaylists();

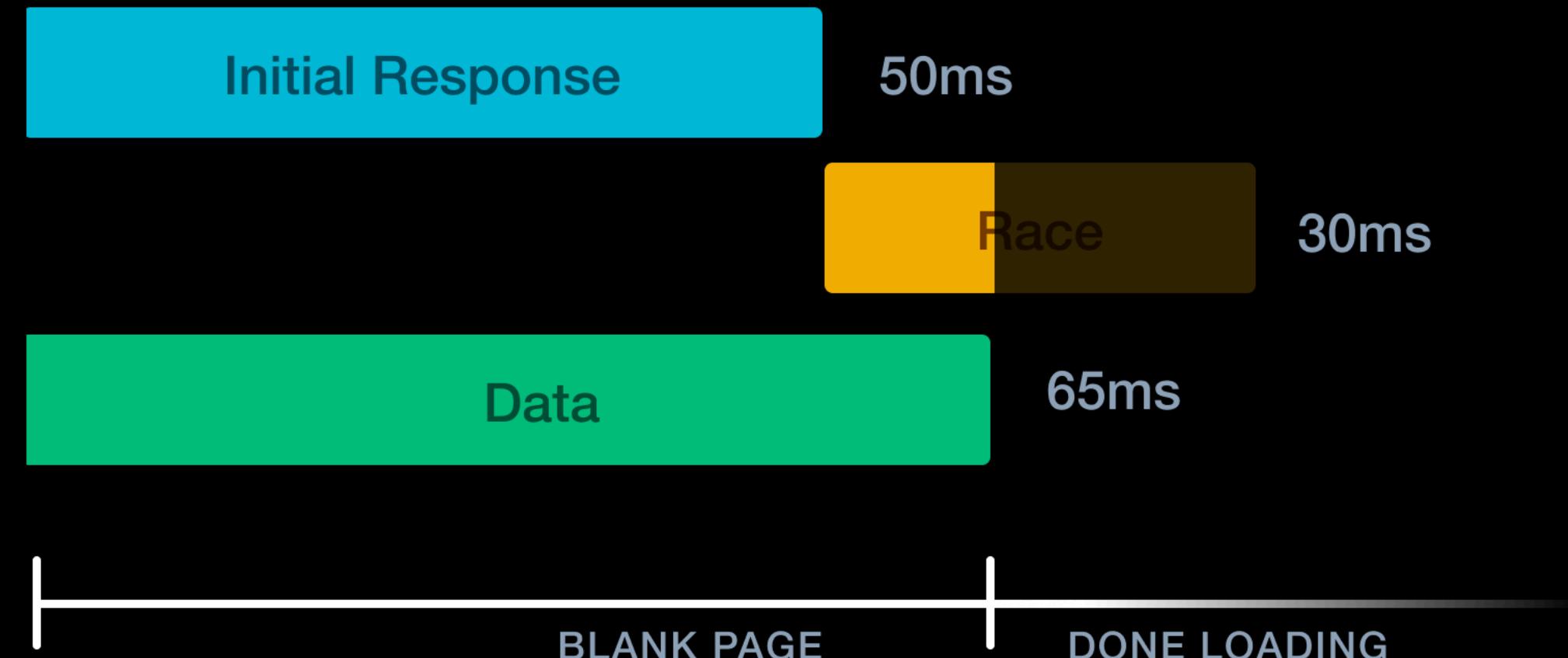
  await Promise.race([
    new Promise((resolve) => setTimeout(resolve, 30)),
    playlists,
  ]);

  return defer({ playlists });
}
```

Cache miss



Cache Hit



Other tips

Other tips

Deploy to the edge for low latency

Other tips

Deploy to the edge for low latency

Use CSS to mimic native interaction

```
touch-action: manipulation; // buttons are instant, no double tap to zoom  
-webkit-tap-highlight-color: transparent; // no black “overlay” on tap target  
padding: env(safe-area-inset-[side]); // extend UI into notch + home bar
```

Other tips

Deploy to the edge for low latency

Use CSS to mimic native interaction

```
touch-action: manipulation; // buttons are instant, no double tap to zoom  
-webkit-tap-highlight-color: transparent; // no black “overlay” on tap target  
padding: env(safe-area-inset-[side]); // extend UI into notch + home bar
```

Read the human interface guidelines and UI documentation

Slides and demo code

github.com/oscarnewman/remix-conf-2023