@hola_soy_milk

5 lessons learned from a major React Native upgrade

—

Ramón Huidobro
https://ramonh.dev

I'm Ramón.

From Chile, living in Austria

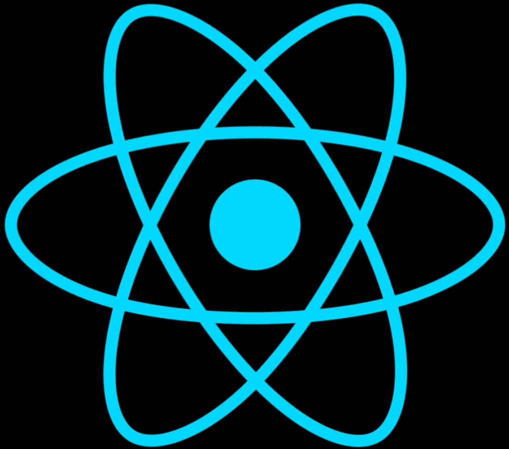
10 years: Freelance software developer

Community member

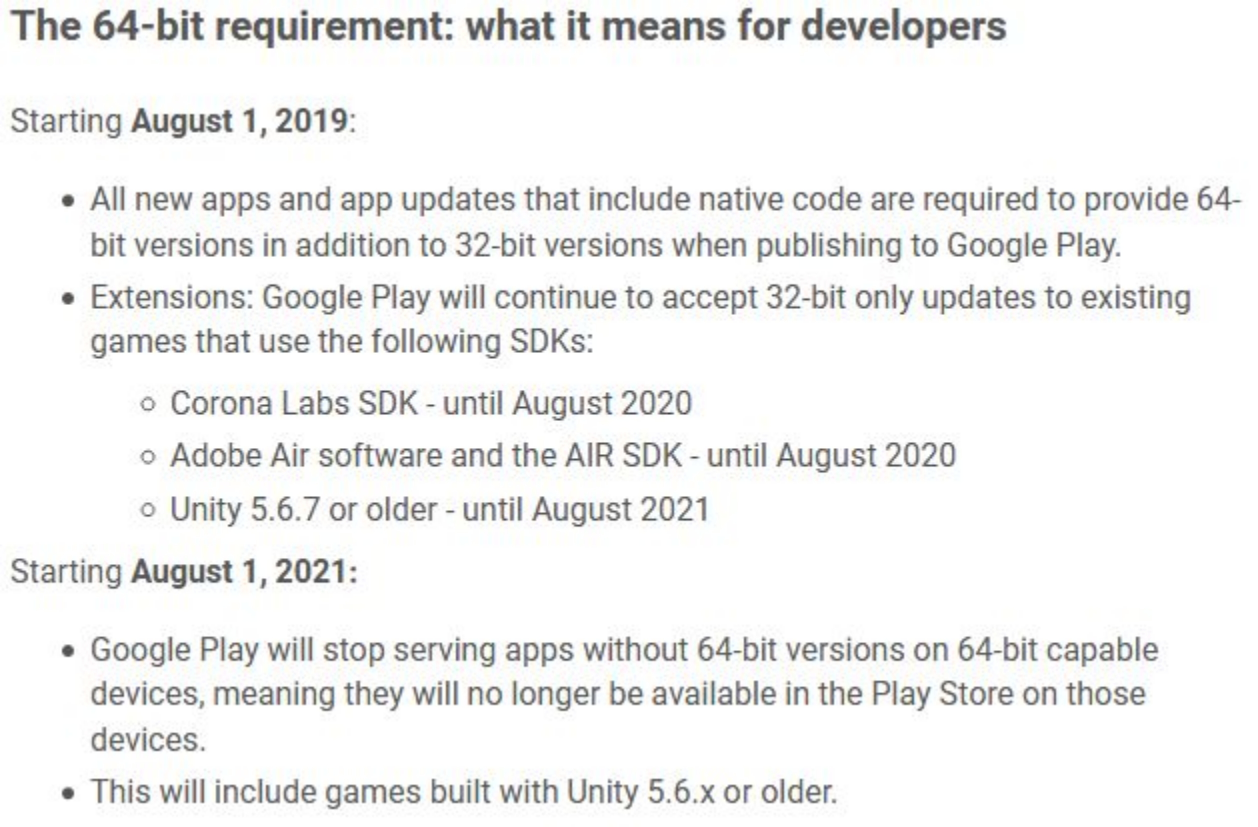
Mozilla tech speaker

“I have a client looking for a React Native dev. Remote, freelance, good rate, could end up being a retainer situation.”

Vienna Freelancer’s Slack group



React Native



The 64-bit requirement: what it means for developers

Starting **August 1, 2019**:

- All new apps and app updates that include native code are required to provide 64-bit versions in addition to 32-bit versions when publishing to Google Play.
- Extensions: Google Play will continue to accept 32-bit only updates to existing games that use the following SDKs:
  - Corona Labs SDK - until August 2020
  - Adobe Air software and the AIR SDK - until August 2020
  - Unity 5.6.7 or older - until August 2021

Starting **August 1, 2021:**

- Google Play will stop serving apps without 64-bit versions on 64-bit capable devices, meaning they will no longer be available in the Play Store on those devices.
- This will include games built with Unity 5.6.x or older.

13

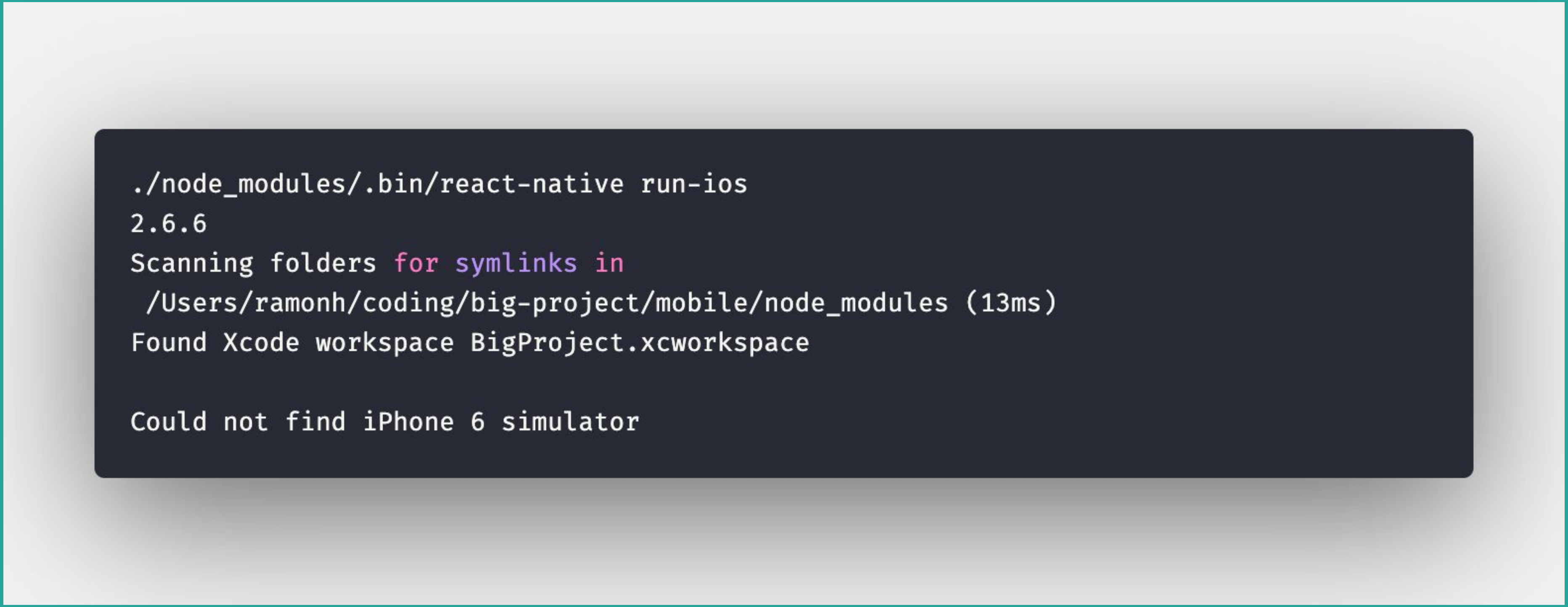
I joined the project in March 2020.

Last update and commit were in March 2019.

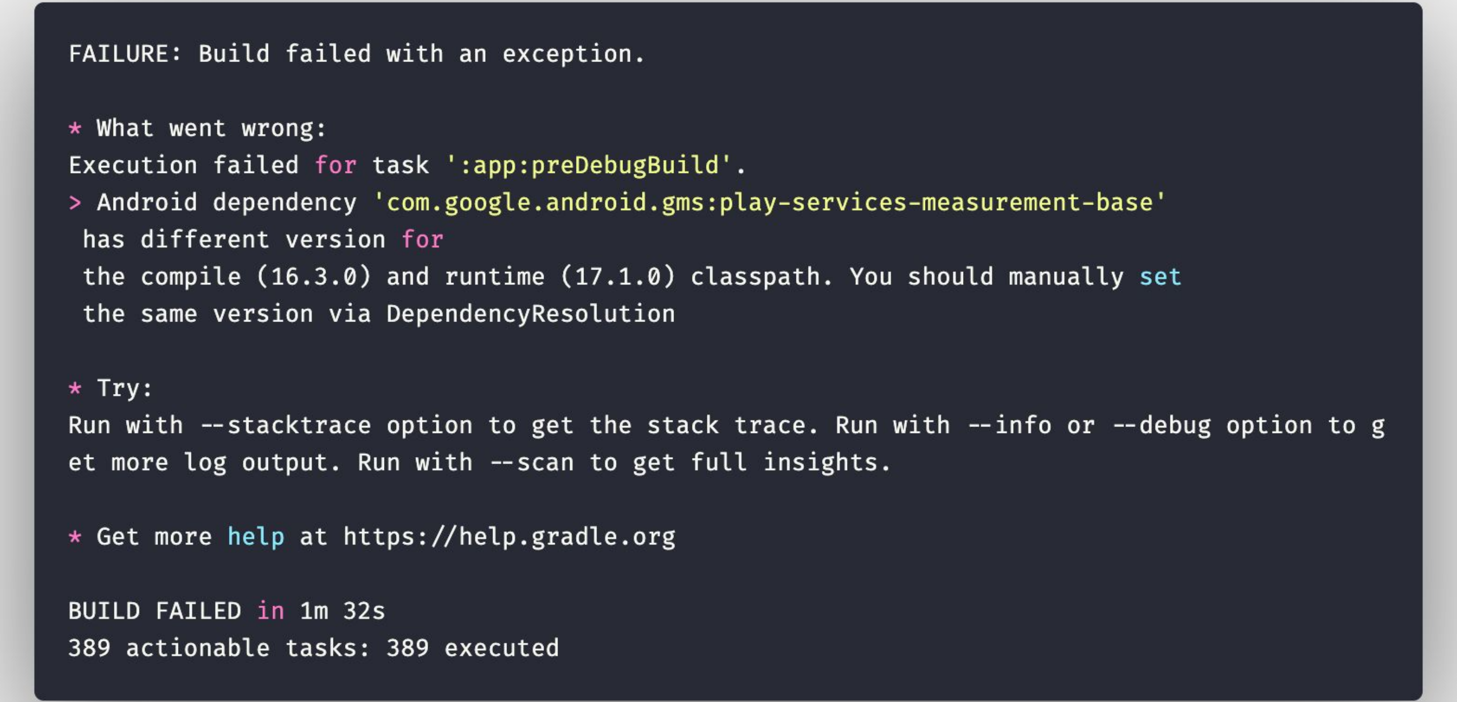
React Native

0.55.4

Current React Native version is 0.62.2

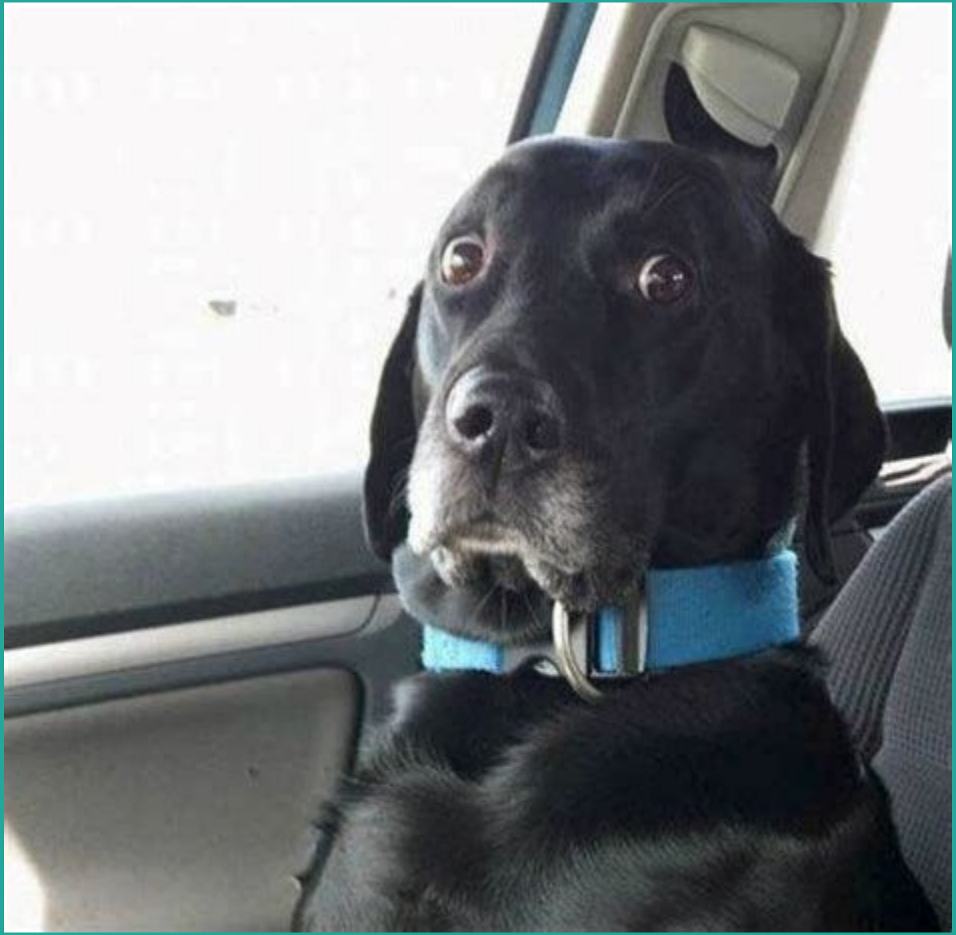


```
./node_modules/.bin/react-native run-ios
2.6.6
Scanning folders for symlinks in
 /Users/ramonh/coding/big-project/mobile/node_modules (13ms)
Found Xcode workspace BigProject.xcworkspace

Could not find iPhone 6 simulator
```



```
FAILURE: Build failed with an exception.

* What went wrong:
Execution failed for task ':app:preDebugBuild'.
> Android dependency 'com.google.android.gms:play-services-measurement-base'
 has different version for
 the compile (16.3.0) and runtime (17.1.0) classpath. You should manually set
 the same version via DependencyResolution

* Try:
Run with --stacktrace option to get the stack trace. Run with --info or --debug option to g
et more log output. Run with --scan to get full insights.

* Get more help at https://help.gradle.org

BUILD FAILED in 1m 32s
389 actionable tasks: 389 executed
```

@hola_soy_milk



> “I’m gonna have to upgrade the HECK outta this.”

Me, at some point

What’s the best practice for managing dependency upgrades?

What's the ~~best~~ my favorite practice for managing dependency upgrades?

“...**if it hurts, do it more often.**
It has the happy property of seeming nonsensical on the surface, but yielding some valuable meaning when you dig deeper”

Martin Fowler

https://martinfowler.com/bliki/FrequencyReducesDifficulty.html

“Art is never finished, only abandoned.”

Leonardo da Vinci

Why is upgrading dependencies so daunting?

Photo by Jeremy Thomas on Unsplash

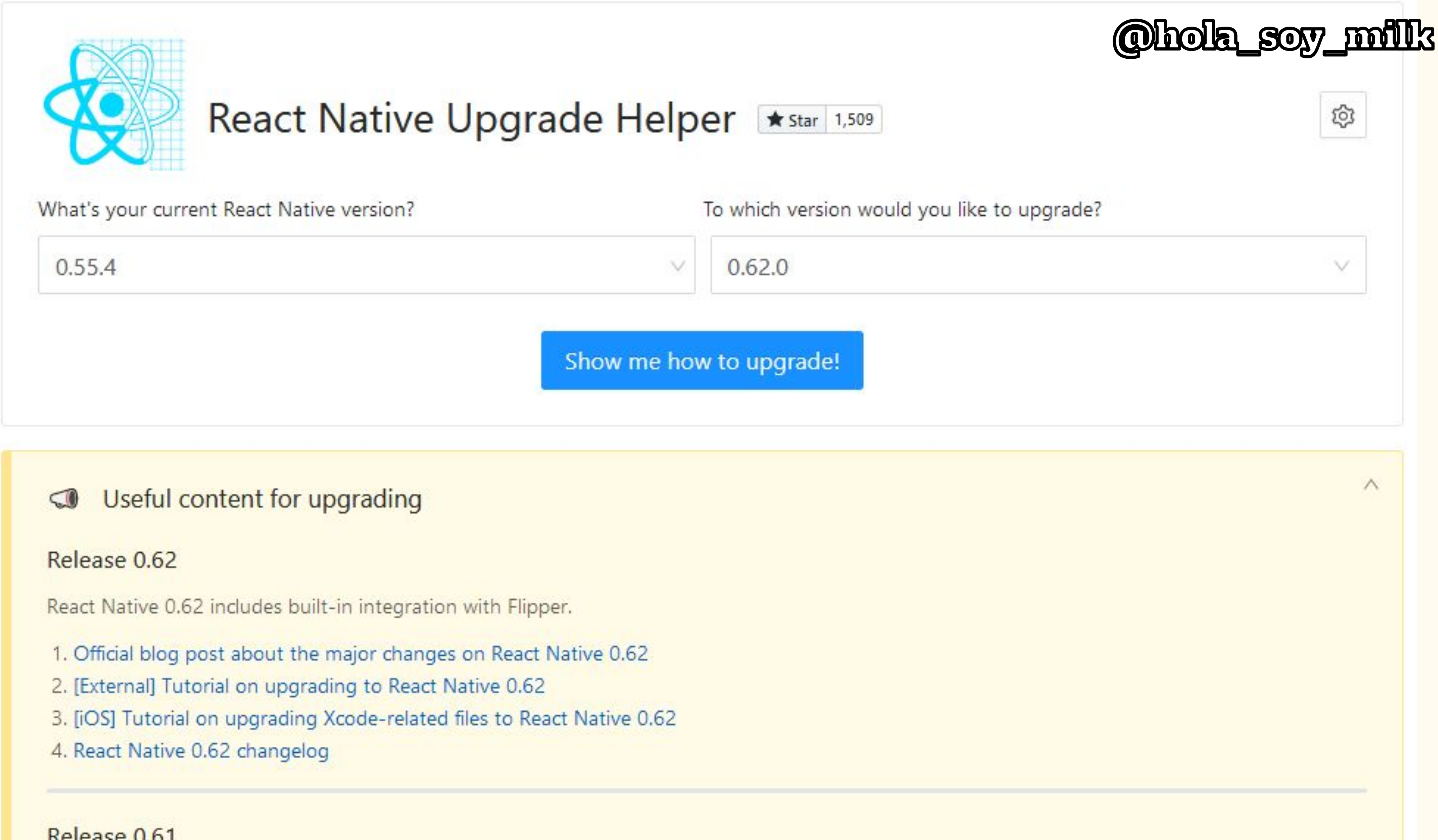
@hola_soy_milk

Photo by John Moeses Bauan on Unsplash

@hola_soy_milk

1. You’re not alone

@hola_soy_milk



React Native Upgrade Helper

What's your current React Native version?

0.55.4

To which version would you like to upgrade?

0.62.0

Useful content for upgrading

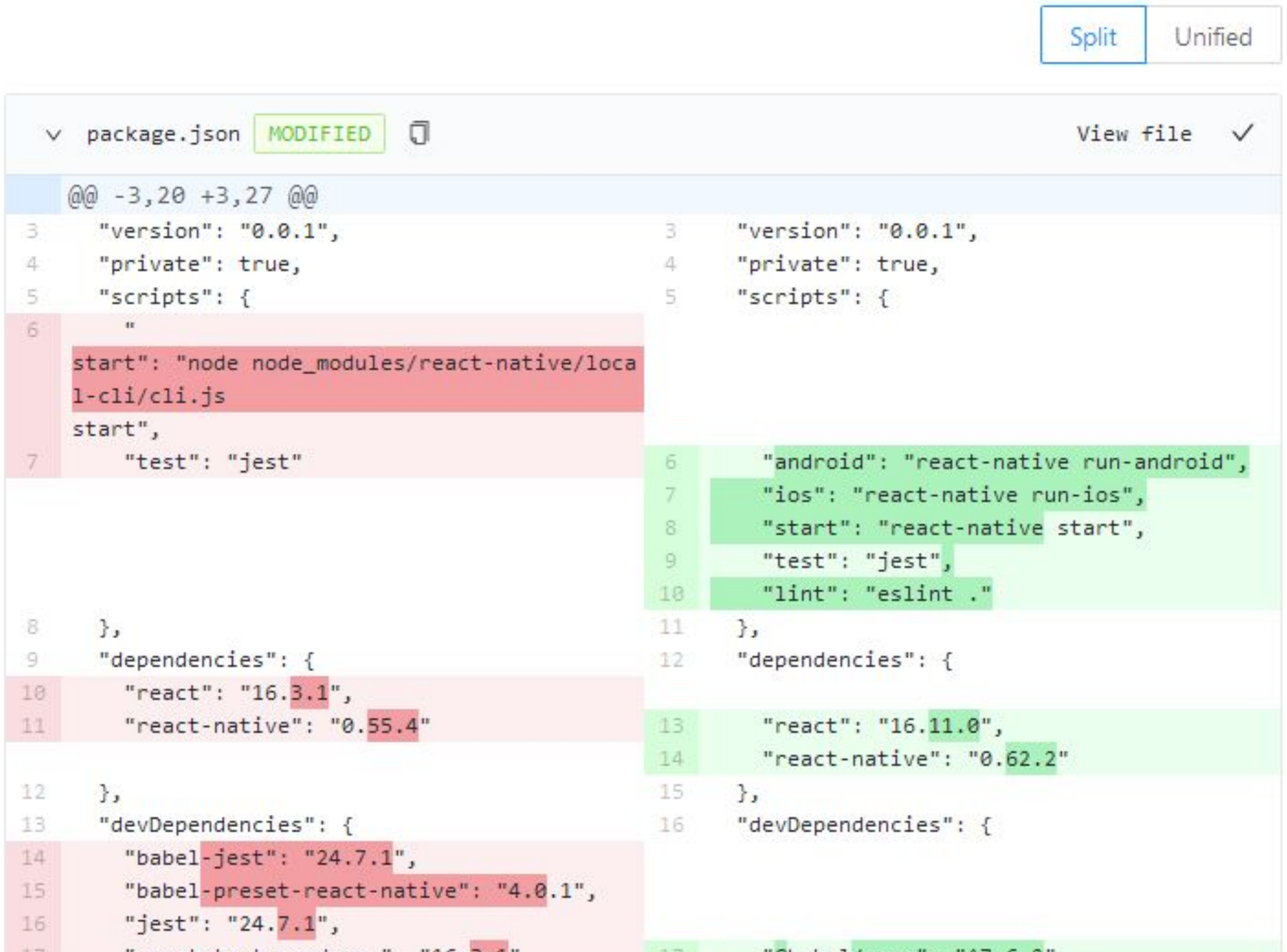
Release 0.62

React Native 0.62 includes built-in integration with Flipper.

1. Official blog post about the major changes on React Native 0.62
2. [External] Tutorial on upgrading to React Native 0.62
3. [iOS] Tutorial on upgrading Xcode-related files to React Native 0.62
4. React Native 0.62 changelog

Release 0.61

top right.

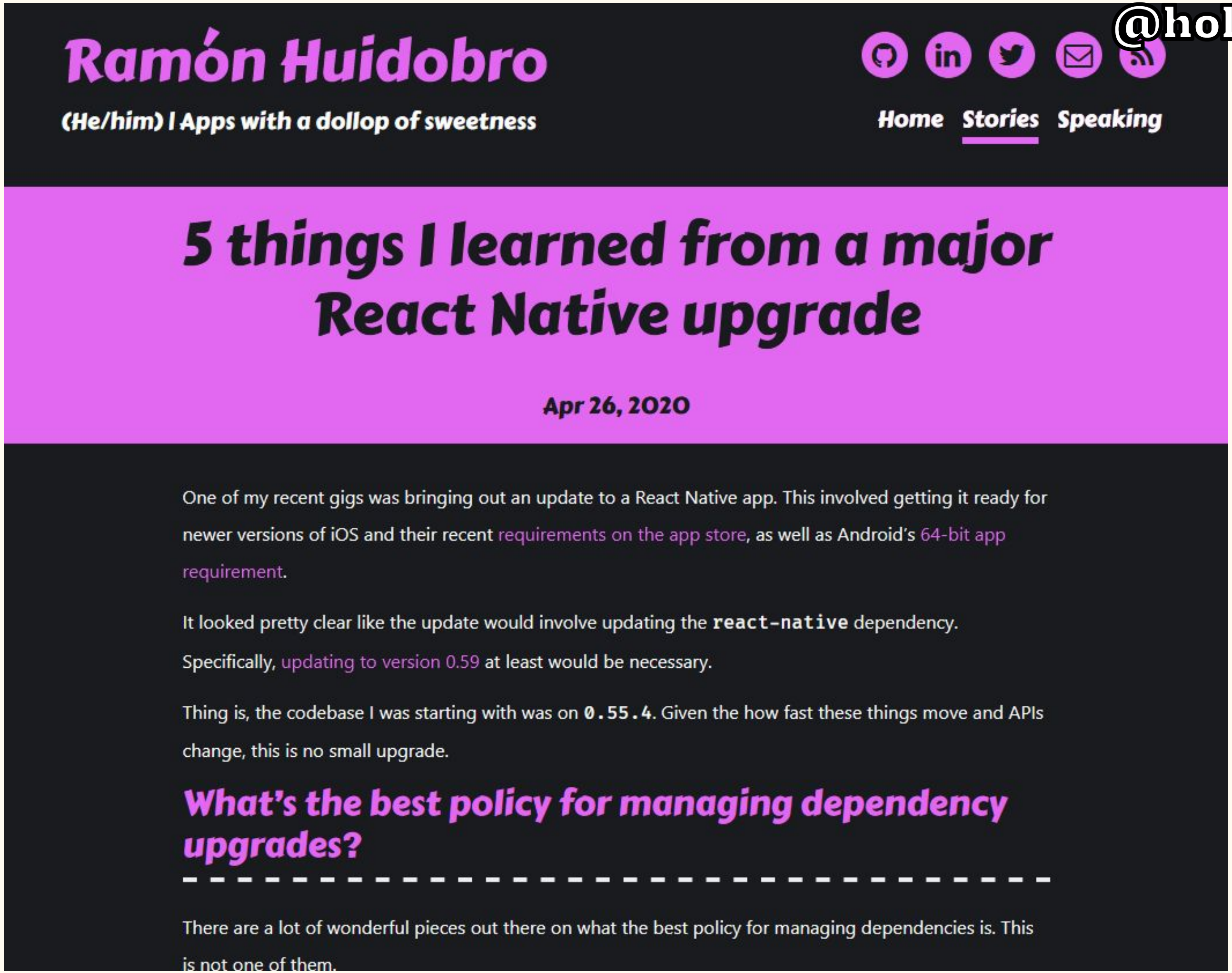


```
@@ -3,20 +3,27 @@
```

```
  "version": "0.0.1",
  "private": true,
  "scripts": {
    "start": "node node_modules/react-native/local-cli/cli.js start",
    "test": "jest"
  },
  "dependencies": {
    "react": "16.3.1",
    "react-native": "0.55.4"
  },
  "devDependencies": {
    "babel-jest": "24.7.1",
    "babel-preset-react-native": "4.0.1",
    "jest": "24.7.1",
```

```
  "version": "0.0.1",
  "private": true,
  "scripts": {
    "android": "react-native run-android",
    "ios": "react-native run-ios",
    "start": "react-native start",
    "test": "jest",
    "lint": "eslint ."
  },
  "dependencies": {
    "react": "16.11.0",
    "react-native": "0.62.2"
  },
  "devDependencies": {
```

@hola_soy_milk



Ramón Huidobro

(He/him) | Apps with a dollop of sweetness

5 things I learned from a major React Native upgrade

Apr 26, 2020

One of my recent gigs was bringing out an update to a React Native app. This involved getting it ready for newer versions of iOS and their recent requirements on the app store, as well as Android's 64-bit app requirement.

It looked pretty clear like the update would involve updating the `react-native` dependency. Specifically, updating to version 0.59 at least would be necessary.

Thing is, the codebase I was starting with was on `0.55.4`. Given the how fast these things move and APIs change, this is no small upgrade.

What's the best policy for managing dependency upgrades?

There are a lot of wonderful pieces out there on what the best policy for managing dependencies is. This is not one of them.

2. Break it down

2. Break it down

a.k.a Divide and Conquer

2. Break it down

a.k.a Divide and Conquer
a.k.a Small victories

- Get the iOS build running
- Get the Android build running

Get the iOS build running

Get the iOS build running

- Adapt to autolinking
- Fix compiler errors
- Fix broken UI elements
- Fix bugs introduced by upgrading
- Fix compiler warnings

Get the iOS build running

- ~~Adapt to autolinking~~
- ~~Fix compiler errors~~
- ~~Fix broken UI elements~~
- ~~Fix bugs introduced by upgrading~~
- ~~Fix compiler warnings~~

Get the Android build running

Get the Android build running

- Adapt to autolinking
- Fix compiler errors
- Fix broken UI elements
- Fix bugs introduced by upgrading
- Fix compiler warnings

Get the Android build running

- ~~Adapt to autolinking~~
- ~~Fix compiler errors~~
- Fix broken UI elements
- Fix bugs introduced by upgrading
- Fix compiler warnings

Get the Android build running

- ~~Adapt to autolinking~~
- ~~Fix compiler errors~~
- Upgrade React Navigation
- Fix bugs introduced by upgrading
- Fix compiler warnings

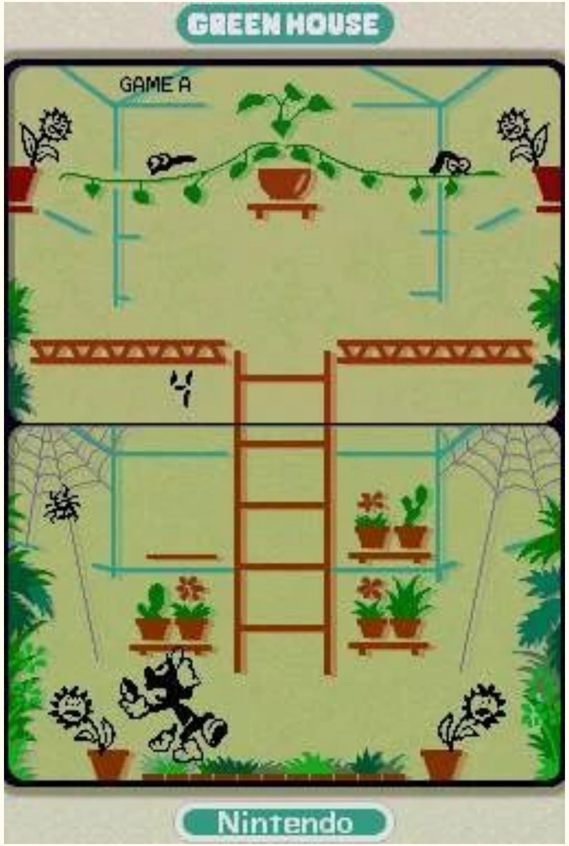
Get the Android build running

- ~~Adapt to autolinking~~
- ~~Fix compiler errors~~
- ~~Upgrade React Navigation~~
- Fix bugs introduced by upgrading
- Fix compiler warnings

Get the Android build running

- ~~Adapt to autolinking~~
- ~~Fix compiler errors~~
- ~~Upgrade React Navigation~~
- Fix crash caused by React Navigation upgrade
- Fix bugs introduced by upgrading
- Fix compiler warnings

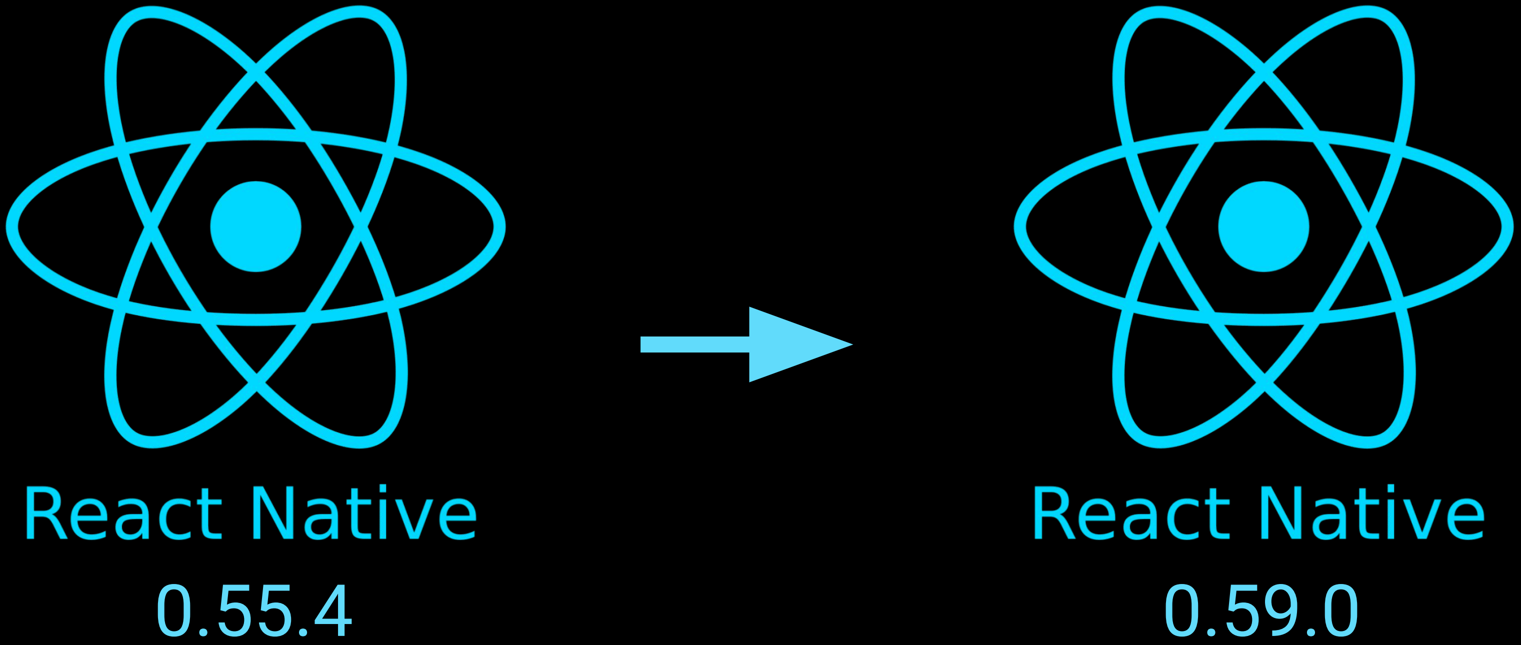
@hola_soy_milk



Game & Watch Greenhouse, Nintendo

Staying focussed on one issue at a time

3. Making large version upgrade leaps



React Native

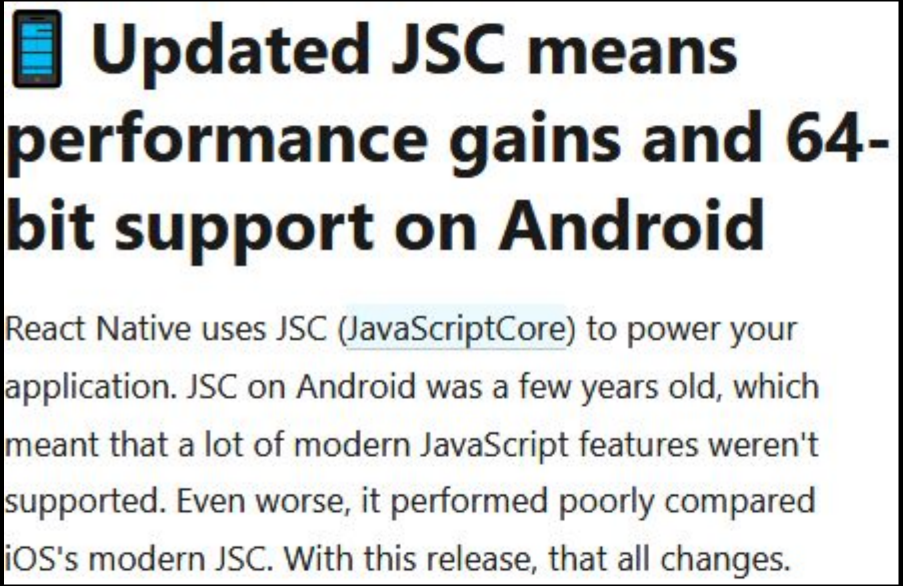
0.55.4

React Native

0.59.0

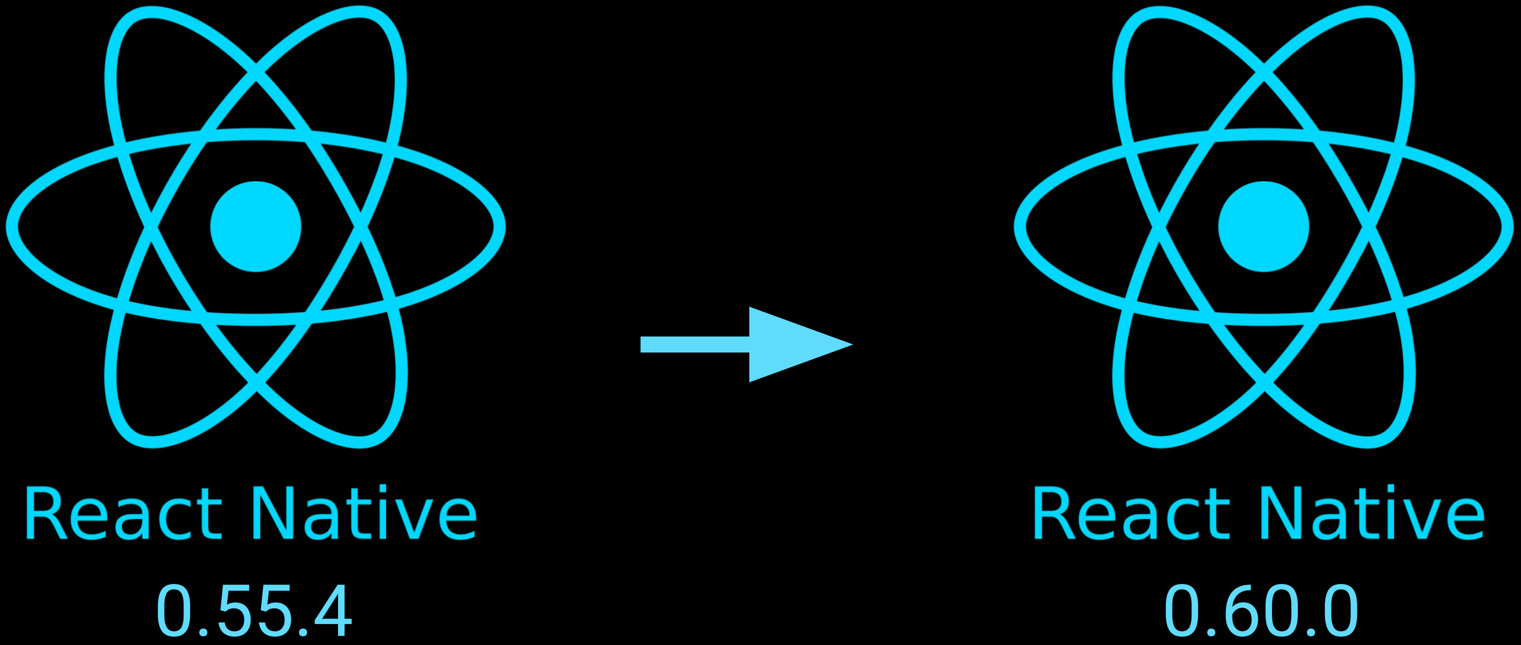
React Native

0.59.0



📱 Updated JSC means performance gains and 64-bit support on Android

React Native uses JSC (JavaScriptCore) to power your application. JSC on Android was a few years old, which meant that a lot of modern JavaScript features weren't supported. Even worse, it performed poorly compared iOS's modern JSC. With this release, that all changes.



React Native

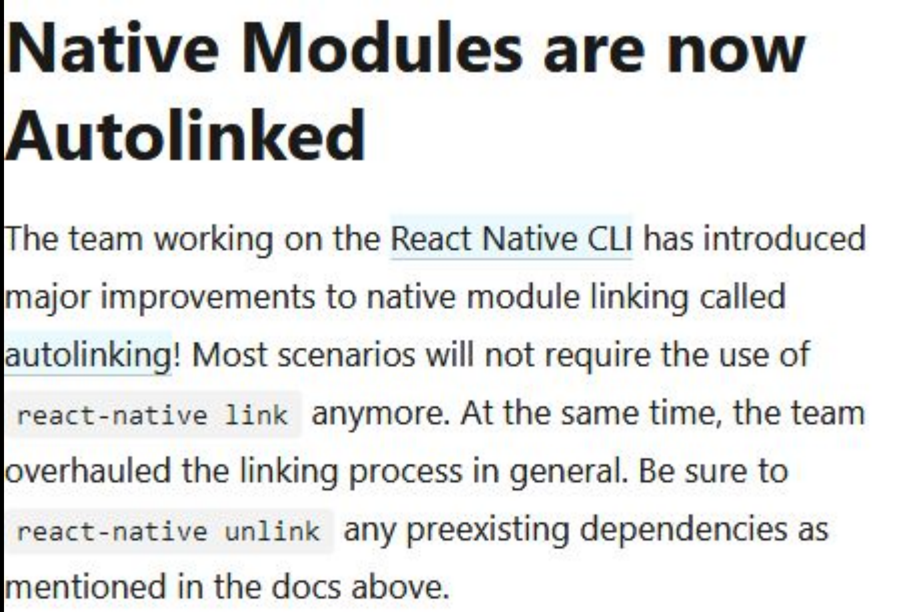
0.55.4

React Native

0.60.0

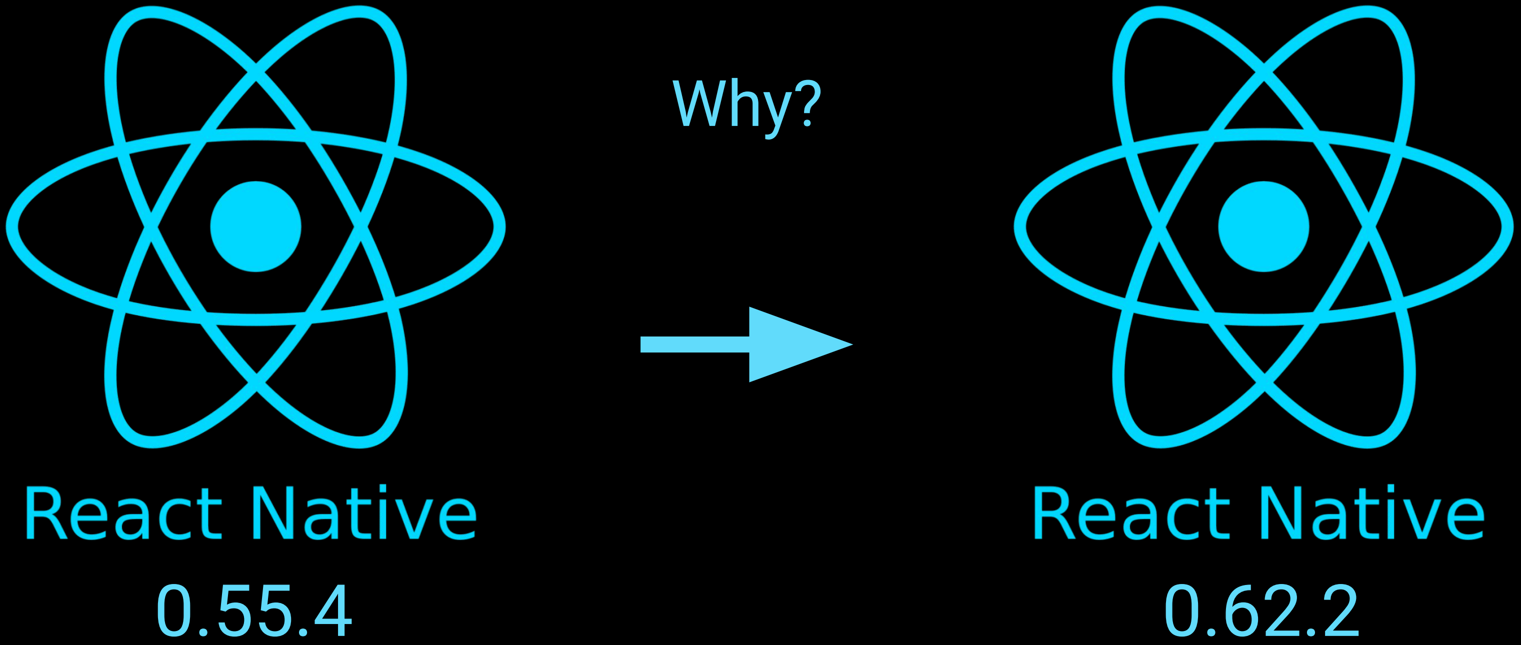
React Native

0.60.0



Native Modules are now Autolinked

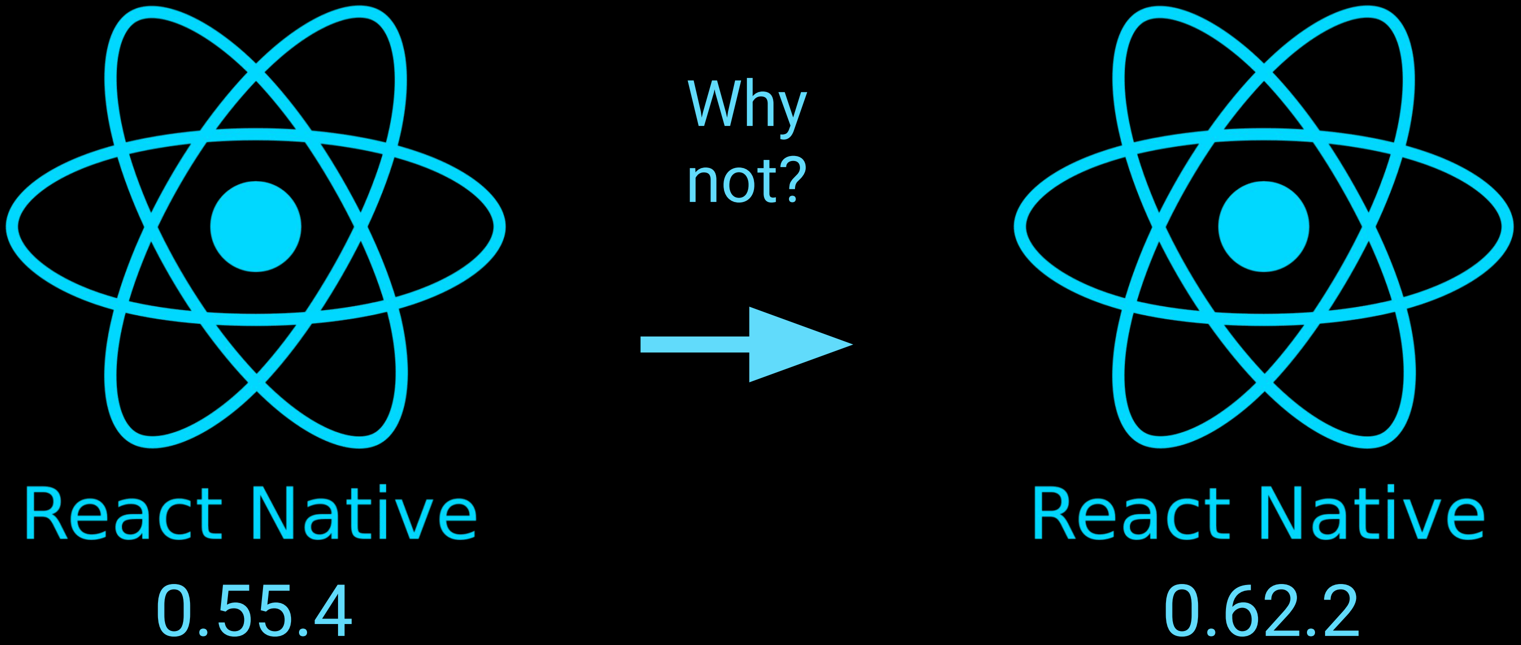
The team working on the React Native CLI has introduced major improvements to native module linking called autolinking! Most scenarios will not require the use of `react-native link` anymore. At the same time, the team overhauled the linking process in general. Be sure to `react-native unlink` any preexisting dependencies as mentioned in the docs above.



React Native
0.55.4

Why?

React Native
0.62.2



React Native

0.55.4

Why not?

React Native

0.62.2

- Stability
- Future-proofing

4. Keep a work-in-progress version control branch

bleeding-edge branches

https://git-scm.com/book/en/v2/Git-Branching-Branching-Workflows

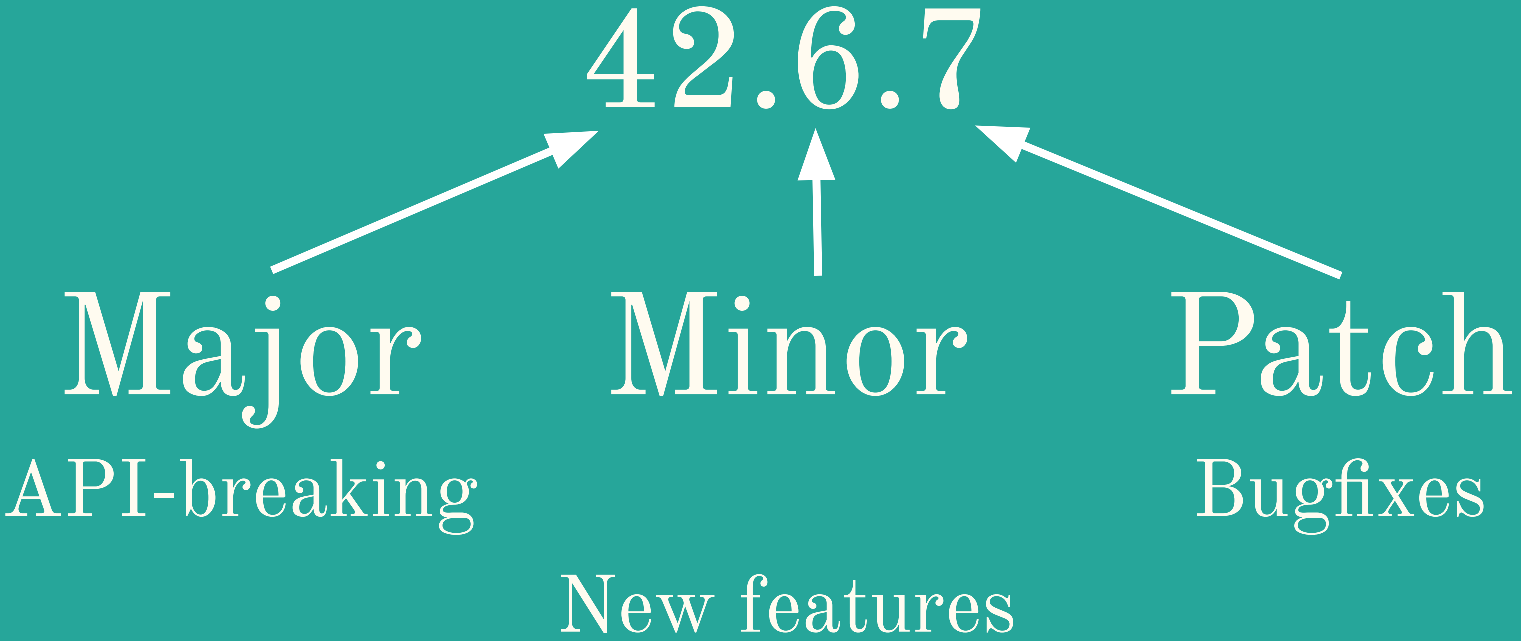
@hola_soy_milk

- Tiny, likely broken commits
- Clear, broken down history
- Rebase-friendly!

5. Know when to stop

72 dependencies listed in package.json

I did not do a major upgrade on all of them



42.6.7

Major

API-breaking

Minor

New features

Patch

Bugfixes

Not now != Never

5 lessons learned

Upgrading React Native
0.55.4 -> 0.62.2

@hola_soy_milk

- You're not alone
- Break it down
- Making large version upgrade leaps
- Keep a work-in-progress version control branch
- Know when to stop

Gratitude

react-native-community upgrade-helper

Contributors 21

+ 10 contributors

Patience

@hola_soy_milk

We’ve all been there before, and likely will again

So am I done?

So am I done?

Absolutely not!

“Software is never finished, only abandoned.”

Leonardo da Vinci, probably

Links

- https://ramonh.dev/js/dependency-management/react-native/2020/04/26/upgrading-dependencies/
- https://martinfowler.com/bliki/FrequencyReducesDifficulty.html
- https://react-native-community.github.io/upgrade-helper/

@hola_soy_milk

❤️🤗

Ramón Huidobro