Understanding a new codebase in 0-60 (days)

Orville Bennett | 2023-10-19

Read

Week 1

Week 1

- Read
  - Git READMEs
  - Confluence
  - Wikis

Play around

Week 2

Interrogate the system

Week 2

Week 2

- Read
- Play Around!
  - Install/Reinstall. Pay attention to error messages.
  - Learn build systems
    - Investigate alternatives.
  - Find problems with given instructions?
    - Submit a pull request
    - Follow it through to being merged
    - Provide feedback on how painful/seamless the process was.

Do something “simple”

Week 3

Week 3

- Take on a ‘simple’ ticket
- Work with someone familiar with the codebase (if possible)
  - Ask many questions
    - About code
    - About process
    - About deficiencies (how can I help)

Summarize It

Week 4

Week 4

- Summarize all you’ve learned
  - Make a list of things you’d like to see improved
  - Start working on the list!

React Components
Develop - 385 vulnerabilities found
Severity: 12 Low | *132 Moderate* | **212 High** | **29 Critical**

55e01cd 59 vulnerabilities found - Packages audited: 2545
Severity: 6 Low | *20* Moderate | **30 High** | **3 Critical**

Old React Components
Features/update-sass - 550 vulnerabilities found - Packages audited: 1833
Severity: 14 Low | 153 Moderate | 338 High | 45 Critical

Get Adventurous

Week 5

Week 5

- Get adventurous
  - Start working towards fixing something on your list
  - Pair with other team members on their tickets
    - Or, be available to pair on things you work on
  - More learning/reading
    - Check out related git repos.
    - Do code reviews on the sly.

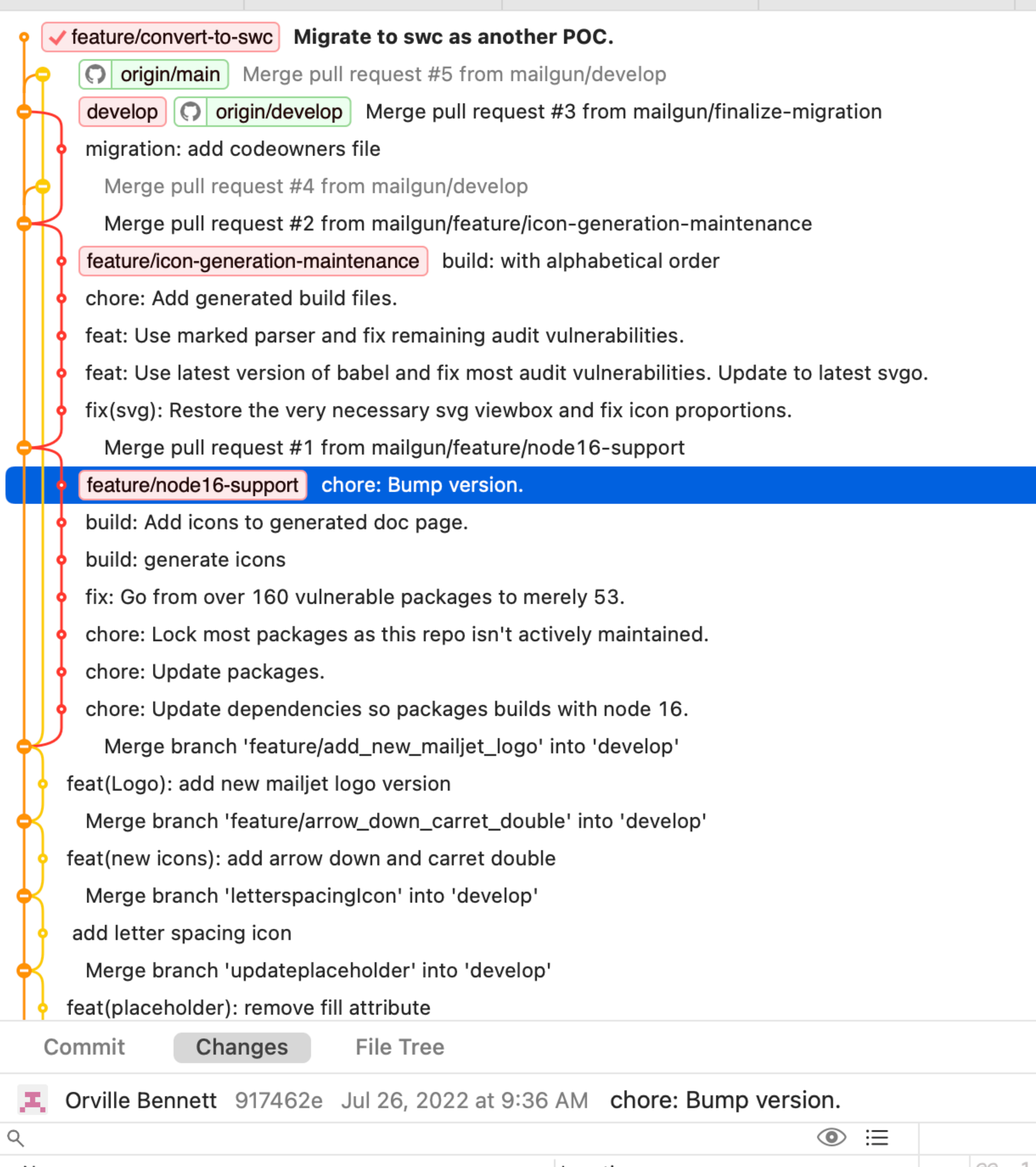
Poke at shared history

Week 6

Week 6

Getting context from our shared history

- Spelunking in git
- Speak with external and internal stakeholders
  - (Or other stakeholders, if you’ve already started this)
  - Learn what pain points they have



```
feature/convert-to-swc  Migrate to swc as another POC.
origin/main  Merge pull request #5 from mailgun/develop
develop  origin/develop  Merge pull request #3 from mailgun/finalize-migration
migration: add codeowners file
Merge pull request #4 from mailgun/develop
Merge pull request #2 from mailgun/feature/icon-generation-maintenance
feature/icon-generation-maintenance  build: with alphabetical order
chore: Add generated build files.
feat: Use marked parser and fix remaining audit vulnerabilities.
feat: Use latest version of babel and fix most audit vulnerabilities. Update to latest svgo.
fix(svg): Restore the very necessary svg viewbox and fix icon proportions.
Merge pull request #1 from mailgun/feature/node16-support
feature/node16-support  chore: Bump version.
build: Add icons to generated doc page.
build: generate icons
fix: Go from over 160 vulnerable packages to merely 53.
chore: Lock most packages as this repo isn't actively maintained.
chore: Update packages.
chore: Update dependencies so packages builds with node 16.
Merge branch 'feature/add_new_mailjet_logo' into 'develop'
feat(Logo): add new mailjet logo version
Merge branch 'feature/arrow_down_carret_double' into 'develop'
feat(new icons): add arrow down and carret double
Merge branch 'letterspacingIcon' into 'develop'
add letter spacing icon
Merge branch 'updateplaceholder' into 'develop'
feat(placeholder): remove fill attribute
```

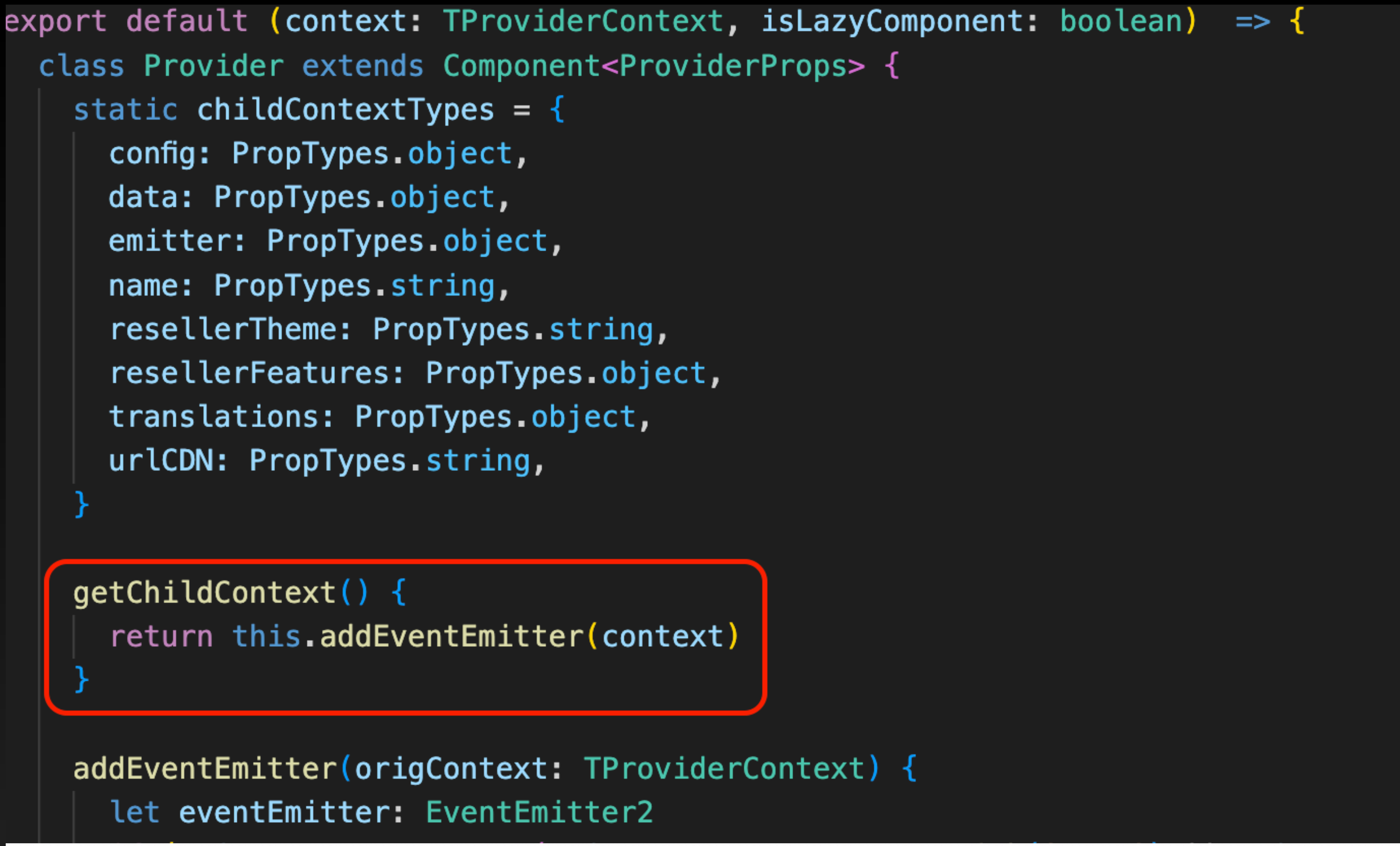
Orville Bennett 917462e Jul 26, 2022 at 9:36 AM chore: Bump version.

Get Ambitious

Week 7

Week 7

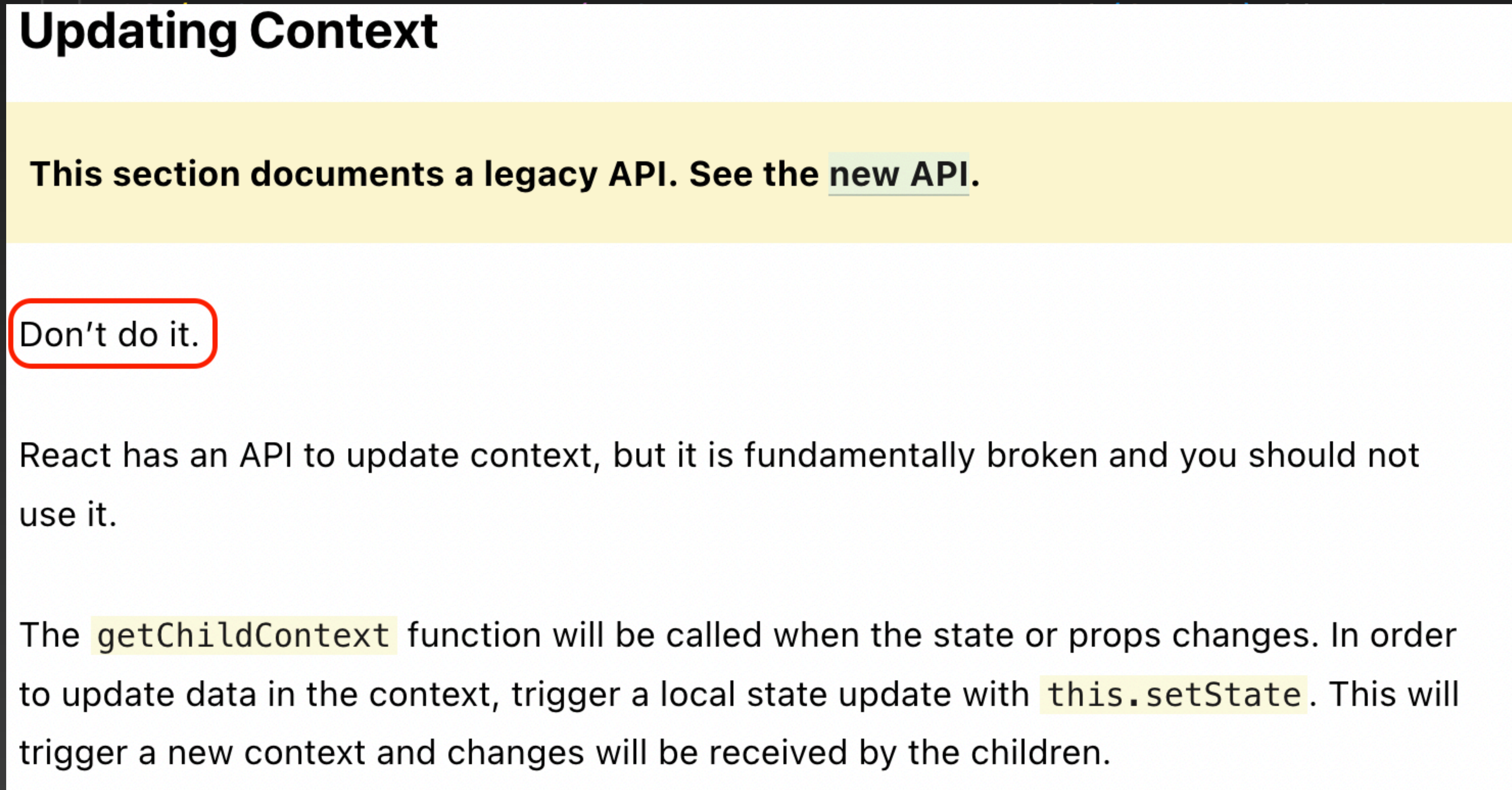
- Attempt more ambitious feature work
- Learn how to deploy your changes



```
export default (context: TProviderContext, isLazyComponent: boolean)  => {
  class Provider extends Component<ProviderProps> {
    static childContextTypes = {
      config: PropTypes.object,
      data: PropTypes.object,
      emitter: PropTypes.object,
      name: PropTypes.string,
      resellerTheme: PropTypes.string,
      resellerFeatures: PropTypes.object,
      translations: PropTypes.object,
      urlCDN: PropTypes.string,
    }

    getChildContext() {
      return this.addEventEmitter(context)
    }

    addEventEmitter(origContext: TProviderContext) {
      let eventEmitter: EventEmitter2
```



Updating Context

This section documents a legacy API. See the new API.

Don’t do it.

React has an API to update context, but it is fundamentally broken and you should not use it.

The `getChildContext` function will be called when the state or props changes. In order to update data in the context, trigger a local state update with `this.setState`. This will trigger a new context and changes will be received by the children.

Keep building. Share.

Week 8

Week 8

- Share your PR for ‘adventurous’ project
- Enjoy the fallout