Looking for current deprecations?

See React Native Deprecations Schedule

The intent of this schedule is to enable maintainers of third-party libraries in the community to upgrade and publish new versions of their libraries before a subsequent React Native Public Release would break their library. Library maintainers should have at least one public release to migrate away from deprecated APIs.

Looking for how to deprecate something?

Definitions

Release Target	Definition
Stable	The current stable release. Tagged `latest` on npm registry
Pre-release	The current release candidate. Tagged `next` on npm registry
Mainline	An uncut version represented by the changes being merged to `main` branch
Future	An unrealized version, on-deck after a pre-release gets promoted to stable.

Deprecation Status	What it means
Planned Deprecation	A warning is issued that this API will be deprecated with relevant link
Deprecated	A warning is issued that this API is deprecated and will be removed soon with relevant link
Planned Removal	An error is issued that this API is removed with relevant link
Removed	Feature is removed from the release

How to rollout breaking change to React Native Public APIs

1. Add your deprecation to the schedule in a new row in the appropriate section.
2. Document somewhere why the deprecation was made and steps for migrating off if it. You can either create a wiki page or if there is existing documentation, submit a PR to update for the relevant release. Link to it on the schedule under the `Change` column.
3. Mark appropriate status (`"Planned Deprecation"` , `"Planned Removal"` , etc.) for the releases that you are targeting. In general, you should be marking the release target **Mainline** as `"Planned Deprecation"` and **Future** as `"Planned Removal"` .
4. Land your change to annotate the deprecated method and/or introduce a deprecation warning. Update the schedule from “Planned Deprecation” to “Deprecated”.
5. Wait for the next relevant public release.
6. Land your change to remove the deprecated method. Update the schedule from “Planned Removal” to “Removal”.

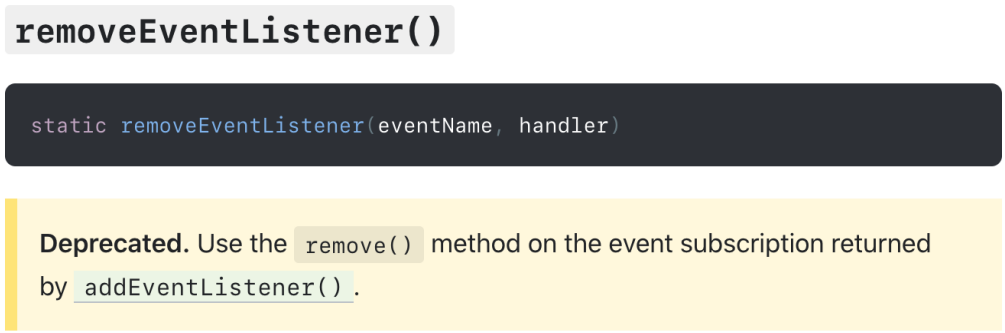
Deprecation Protocols

Different types of changes and languages warrant different deprecation mechanisms. Here is a non-comprehensive list of suggestions for how to deprecate APIs.

General

- Use "Deprecated" in the *Changelog:* section of your diff summary.
- Update the React Native Website to document when an API is deprecated (or removed).
  - Follow these 3-5 "Getting Started" steps to for the React Native Website.
  - Find the existing documentation for the API being deprecated.
  - Add a deprecation notice. For example (see example Pull Request):

```
   > **Deprecated.** Use the `remove()` method on the event subscription
returned by[addEventListener()](#addeventlistener).
```



  - Don't forget to also update any example code that references the deprecated API.

JavaScript

- Add a console.warn explaining the deprecation (TODO example)
- Add @deprecated to the docblock (TODO example).

Java

- TODO: Add @Deprecated

Objective-C

- ???

C++

- ???