# test(types): improve tests for type definitions #9635

**What kind of change does this PR introduce?** Code style update. Build-related changes.

**Does this PR introduce a breaking change?** No.

**The PR fulfills these requirements:** It's submitted to the dev branch for v2.x (or to a previous version branch), not the master branch. All tests are passing: <https://github.com/vuejs/vue/blob/dev/.github/CONTRIBUTING.md#development-setup>. New/updated tests are included.

**Other information:** Just test improvement.

# feat(types): improve some type definitions #9603

**What kind of change does this PR introduce?** Feature.

**Does this PR introduce a breaking change?** No.

**The PR fulfills these requirements:** It's submitted to the dev branch for v2.x (or to a previous version branch), not the master branch. All tests are passing: <https://github.com/vuejs/vue/blob/dev/.github/CONTRIBUTING.md#development-setup>.

If adding a **new feature**, the PR's description includes: A convincing reason for adding this feature (to avoid wasting your time, it's best to open a suggestion issue first and wait for approval before working on it): Improve type definition for TypeScript users to make Vue stronger typed.

# feat(global-api): warn when plugin is invalid #9601

**What kind of change does this PR introduce?** Feature.

**Does this PR introduce a breaking change?** No.

**The PR fulfills these requirements:** It's submitted to the dev branch for v2.x (or to a previous version branch), not the master branch. All tests are passing: <https://github.com/vuejs/vue/blob/dev/.github/CONTRIBUTING.md#development-setup>.

If adding a **new feature**, the PR's description includes: A convincing reason for adding this feature (to avoid wasting your time, it's best to open a suggestion issue first and wait for approval before working on it): user should be able to see issue with plugin.

# feat: add access to context from serverPrefetch (fix #9447) #9591

**What kind of change does this PR introduce?** Feature.

**Does this PR introduce a breaking change?** Yes.

If yes, please describe the impact and migration path for existing applications: First argument of hook will be context, not vm.

**The PR fulfills these requirements:** It's submitted to the dev branch for v2.x (or to a previous version branch), not the master branch. When resolving a specific issue, it's referenced in the PR's title (e.g. fix #xxx[,#xxx], where "xxx" is the issue number). All tests are passing: <https://github.com/vuejs/vue/blob/dev/.github/CONTRIBUTING.md#development-setup>. New/updated tests are included.

If adding a **new feature**, the PR's description includes: A convincing reason for adding this feature (to avoid wasting your time, it's best to open a suggestion issue first and wait for approval before working on it): [#9447](https://github.com/vuejs/vue/issues/9447).

# feat(build script): Return exit code #9567

**What kind of change does this PR introduce?** Feature.

**Does this PR introduce a breaking change?** No.

**The PR fulfills these requirements:** It's submitted to the dev branch for v2.x (or to a previous version branch), not the master branch.

**Other information:** Return error exit code from the build script in case of error.

# feat(lifecycle): add beforeDeactivate hook (implement #9454) #9564

**What kind of change does this PR introduce?** Feature.

**Does this PR introduce a breaking change?** No.

**The PR fulfills these requirements:** It's submitted to the dev branch for v2.x (or to a previous version branch), not the master branch. When resolving a specific issue, it's referenced in the PR's title (e.g. fix #xxx[,#xxx], where "xxx" is the issue number). All tests are passing: <https://github.com/vuejs/vue/blob/dev/.github/CONTRIBUTING.md#development-setup>.

If adding a **new feature**, the PR's description includes: A convincing reason for adding this feature (to avoid wasting your time, it's best to open a suggestion issue first and wait for approval before working on it).

**Other information:** Closes [#9454](https://github.com/vuejs/vue/issues/9454).

# style(util): remove unnessary variable in cached #9531

**What kind of change does this PR introduce?** Code style update.

**Does this PR introduce a breaking change?** No.

**The PR fulfills these requirements:** It's submitted to the dev branch for v2.x (or to a previous version branch), not the master branch.All tests are passing: <https://github.com/vuejs/vue/blob/dev/.github/CONTRIBUTING.md#development-setup>.

**Other information:** remove unnessary variable in cached.

# fix(util): handle more edge cases in `looseEqual()` util and add spec #9527

**What kind of change does this PR introduce?** Bugfix. Refactor.

**Does this PR introduce a breaking change?** No.

**The PR fulfills these requirements:** It's submitted to the dev branch for v2.x (or to a previous version branch), *not* the master branch. When resolving a specific issue, it's referenced in the PR's title (e.g. fix #xxx[,#xxx], where "xxx" is the issue number). All tests are passing: <https://github.com/vuejs/vue/blob/dev/.github/CONTRIBUTING.md#development-setup>. New/updated tests are included.

**Other information:**

This PR makes the looseEqual() util more bulletproof and adds an extra spec for it.

The previous implementation failed at following tests:

Expected: false

Received: true

65 | // Files with slightly different dates

> 66 | expect(looseEqual(file1, file3)).toBe(false)

Expected: false

Received: true

102 | // Different regex

> 103 | expect(looseEqual(rx1, rx3)).toBe(false)

Expected: false

Received: true

> 153 | expect(looseEqual(obj1, date1)).toBe(false)

Performance should also be equal or slightly better because early exits are used wherever possible: <https://jsperf.com/looseequal-comparison>.

Shoutout to [**@tmorehouse**](https://github.com/tmorehouse) for helping with the tests and handling all the edge cases.

# Expose handleError utility function. #9502

In Vue plugins sometimes it is useful to handle errors in the same way Vue does. In this way hooks can intercept them and so on. To not duplicate the code, handleError could be exposed under Vue.util.

**What kind of change does this PR introduce?** Feature.

**Does this PR introduce a breaking change?** No.

**The PR fulfills these requirements:** It's submitted to the dev branch for v2.x (or to a previous version branch), not the master branch. All tests are passing: <https://github.com/vuejs/vue/blob/dev/.github/CONTRIBUTING.md#development-setup>.

If adding a **new feature**, the PR's description includes: A convincing reason for adding this feature (to avoid wasting your time, it's best to open a suggestion issue first and wait for approval before working on it)**.**

# feat(errorHandler): async error handling for watchers (fix #9625 #10009) #9484

Added async error handling for watchers and immediate watchers

**What kind of change does this PR introduce?** Feature.

**Does this PR introduce a breaking change?** No.

**The PR fulfills these requirements:** It's submitted to the dev branch for v2.x (or to a previous version branch), not the master branch. When resolving a specific issue, it's referenced in the PR's title (e.g. fix #xxx[,#xxx], where "xxx" is the issue number). All tests are passing: <https://github.com/vuejs/vue/blob/dev/.github/CONTRIBUTING.md#development-setup>. New/updated tests are included.

**Other information:** Fixes [#9625](https://github.com/vuejs/vue/issues/9625). Fixes [#10009](https://github.com/vuejs/vue/issues/10009).

# feat: include errors in Vue.compile result #9482

**What kind of change does this PR introduce?** Feature. Code style update.

**Does this PR introduce a breaking change?** No.

**The PR fulfills these requirements:** It's submitted to the dev branch for v2.x (or to a previous version branch), not the master branch. When resolving a specific issue, it's referenced in the PR's title (e.g. fix #xxx[,#xxx], where "xxx" is the issue number). All tests are passing: <https://github.com/vuejs/vue/blob/dev/.github/CONTRIBUTING.md#development-setup>. New/updated tests are included.

**Other information:** fixed [#8467](https://github.com/vuejs/vue/issues/8467).

# feat(ssr): add ability to cleanup after request #9479

close [#9463](https://github.com/vuejs/vue/issues/9463)

**What kind of change does this PR introduce?** Bugfix. Feature.

**Does this PR introduce a breaking change?** No.

**The PR fulfills these requirements:** It's submitted to the dev branch for v2.x (or to a previous version branch), not the master branch. When resolving a specific issue, it's referenced in the PR's title (e.g. fix #xxx[,#xxx], where "xxx" is the issue number). All tests are passing: <https://github.com/vuejs/vue/blob/dev/.github/CONTRIBUTING.md#development-setup>. New/updated tests are included.

If adding a **new feature**, the PR's description includes: A convincing reason for adding this feature (to avoid wasting your time, it's best to open a suggestion issue first and wait for approval before working on it).

**Other information:** The point of this "feature" is to be able to do something to the app to mark it as "invalid" so that vue-router has a way to detect and cleanup objects it might have left hanging around in memory. This feature might be able to be used for other reasons, but it is probably truly a fix. Also, a change would need to be approved in vue-router as well for the memory leak there to even go away (see [vuejs/vue-router#2606](https://github.com/vuejs/vue-router/issues/2606)), and that might get fixed by other means, so this might not be worth doing anything with if that doesn't also get approved.

Because we're in control of when the app is created in SSR it didn't seem like there was any other sensible option but to give us control of when to clean it up, though maybe something can be done based on runInNewContext. Also, maybe a more explicit API would be better, e.g. some way to mark the app as "served", $isServer && $isServed.

Note I'm submitting this knowing that there is a good chance that this will be tossed, so feel free to do so if this is not what is wanted.

# fix #9467, Custom validation message in custom prop validator #9470

**What kind of change does this PR introduce?** Feature.

**Does this PR introduce a breaking change?** No.

**The PR fulfills these requirements:** It's submitted to the dev branch for v2.x (or to a previous version branch), not the master branch. When resolving a specific issue, it's referenced in the PR's title (e.g. fix #xxx[,#xxx], where "xxx" is the issue number). All tests are passing: <https://github.com/vuejs/vue/blob/dev/.github/CONTRIBUTING.md#development-setup>. New/updated tests are included.

If adding a **new feature**, the PR's description includes: A convincing reason for adding this feature (to avoid wasting your time, it's best to open a suggestion issue first and wait for approval before working on it).

**Other information:** [#9467](https://github.com/vuejs/vue/issues/9467).

# refactor: remove duplicate import sources #9469

**What kind of change does this PR introduce?** Refactor.

**Does this PR introduce a breaking change?** No.

**The PR fulfills these requirements:** It's submitted to the dev branch for v2.x (or to a previous version branch), not the master branch. When resolving a specific issue, it's referenced in the PR's title (e.g. fix #xxx[,#xxx], where "xxx" is the issue number). All tests are passing: <https://github.com/vuejs/vue/blob/dev/.github/CONTRIBUTING.md#development-setup>. New/updated tests are included.

# feat: add origin prop for <transition-group>, fix #8424 #9430

**What kind of change does this PR introduce?** Bugfix. Feature.

**Does this PR introduce a breaking change?** No.

**The PR fulfills these requirements:** It's submitted to the dev branch for v2.x (or to a previous version branch), not the master branch. When resolving a specific issue, it's referenced in the PR's title (e.g. fix #xxx[,#xxx], where "xxx" is the issue number). All tests are passing: <https://github.com/vuejs/vue/blob/dev/.github/CONTRIBUTING.md#development-setup>.

If adding a **new feature**, the PR's description includes: A convincing reason for adding this feature (to avoid wasting your time, it's best to open a suggestion issue first and wait for approval before working on it).

As described in [#8424](https://github.com/vuejs/vue/issues/8424), <transition-group> now records the items' position relative to the viewport (via getBoundingClientRect()). While page scroll happens just before the items start to perform transition, it will look like that they are flashed out of the container and slowly move to their original position.

This PR added a new prop: origin for <transition-group>, which is a string value that indicates how we are gonna calculate the position of transition items. Available values are viewport (the current behavior) and document. When set to document, the position of the items are always calculated relative to the top left corner of the document element, which will retain their position in the document when page scroll occurs.

**Other information:** Here is a show case of this PR: <https://codepen.io/Justineo/project/editor/XxEKgb>. Scroll to the bottom and click - to see the transition.

# chore: add `repository.directory` field to each `package.json` #9403

**What kind of change does this PR introduce?** Other, please describe: I created a similar PR in vue-cli repo, so I will just copy&paste my comment here: This field allows you specify path of each package in a monorepo thanks to [this npm RFC](https://github.com/npm/rfcs/blob/latest/accepted/0010-monorepo-subdirectory-declaration.md). Main benefit of providing this field is a more accurate link to the package's source code from its npmjs.com page.

**Does this PR introduce a breaking change?** No.

**The PR fulfills these requirements:** It's submitted to the dev branch for v2.x (or to a previous version branch), not the master branch.

# feat(types): add DOM type definition for JSX element #9369

**What kind of change does this PR introduce?** Feature.

**Does this PR introduce a breaking change?** No.

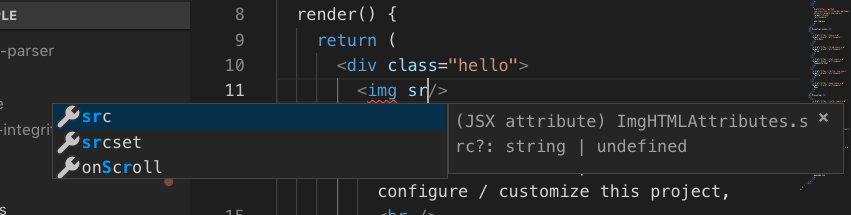
**The PR fulfills these requirements:** It's submitted to the dev branch for v2.x (or to a previous version branch), *not* the master branch. All tests are passing: <https://github.com/vuejs/vue/blob/dev/.github/CONTRIBUTING.md#development-setup>.

**Other information:**

**Descriptions**

* Inspired by [vue-tsx-support](https://github.com/wonderful-panda/vue-tsx-support) by [**@wonderful-panda**](https://github.com/wonderful-panda) who was also inspired by [@types/react](https://github.com/DefinitelyTyped/DefinitelyTyped/tree/master/types/react/v15)
* This PR is not aiming for perfection. I'd love to hear feedbacks from Vue core team as this is my very first PR.

By providing officially maintained HTML Attributes Types, developers can get a huge benefit when they are writing TSX.

[](https://user-images.githubusercontent.com/8381075/51739106-07f28280-20d4-11e9-849f-ce708eef0765.png)

All you need is just a few tweaks on shims-tsx.d.ts

# feat(props): allow defining a required prop as null #9358

Based on [#1961](https://github.com/vuejs/vue/issues/1961). It wasn't possible to specify a required prop of type object and pass null as the value.

**What kind of change does this PR introduce?** Feature.

**Does this PR introduce a breaking change?** No.

**Other information:** The tests cases reflect what I'm trying to fix. I remember this was pointed out multiple times in the past. We cannot explicitely pass a null prop when the type is Object and the prop is required. This would keep current behaviour but allow having the type null for explicit null values. I want to add more tests and maybe refactor but I remember this was problematic as it could break some validation, so I would like so feedback about types you may use that would break with this.

# fix #9284: Retrigger propsData getter before current watcher teardown #9327

**What kind of change does this PR introduce?** Bugfix.

**Does this PR introduce a breaking change?** No.

**The PR fulfills these requirements:** It's submitted to the dev branch for v2.x (or to a previous version branch), not the master branch. When resolving a specific issue, it's referenced in the PR's title (e.g. fix #xxx[,#xxx], where "xxx" is the issue number). All tests are passing: <https://github.com/vuejs/vue/blob/dev/.github/CONTRIBUTING.md#development-setup>. New/updated tests are included.

**Other information:** You can test in this case <https://github.com/MR-fisher/vue-memory-leak-case>.

# Fix #9224 Warn type when not a constructor #9241

Closes [#9241](https://github.com/vuejs/vue/pull/9241)

**What kind of change does this PR introduce?** Other, please describe: Error Handling.

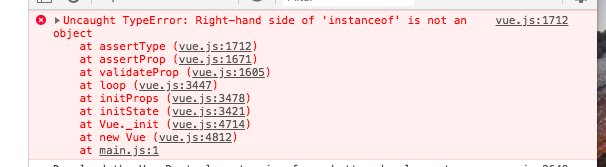
**Does this PR introduce a breaking change?** No.

**The PR fulfills these requirements:** It's submitted to the dev branch for v2.x (or to a previous version branch), not the master branch. When resolving a specific issue, it's referenced in the PR's title (e.g. fix #xxx[,#xxx], where "xxx" is the issue number). All tests are passing: <https://github.com/vuejs/vue/blob/dev/.github/CONTRIBUTING.md#development-setup>. New/updated tests are included.

**Other information:**

### Proof

**Before**

[](https://user-images.githubusercontent.com/18662248/50373943-af205d80-05ee-11e9-930d-ae1e798b03fc.png)

**After**

[screen shot 2018-12-22 at 1 37 43 pm](https://user-images.githubusercontent.com/18662248/50373959-045c6f00-05ef-11e9-8eb8-5715e099cdde.png)