Vue Workshop (1/2)

Workshop goal

Workshop goal

- Part 1
 - Migrate from Vue 2 to Vue 3
 - Learn about the Vue Migration build
 - Upgrade the Vue Router
 - Create reusable components
 - Learn about the composition API
 - Create composition functions
- Part 2
 - Use components Slots
 - Manage state using VueX
 - Explore the Vue Router
 - Use Vuetify components
 - Type check the code using TypeScript
 - Server side rendering of Vue applications using Nuxt





- Maurice de Beijer
- The Problem Solver
- Microsoft MVP
- Freelance lead/developer/instructor
- Twitter: <a>@mauricedb
- Web: http://www.TheProblemSolver.nl
- E-mail: <u>maurice.de.beijer@gmail.com</u>



Type it out by hand?

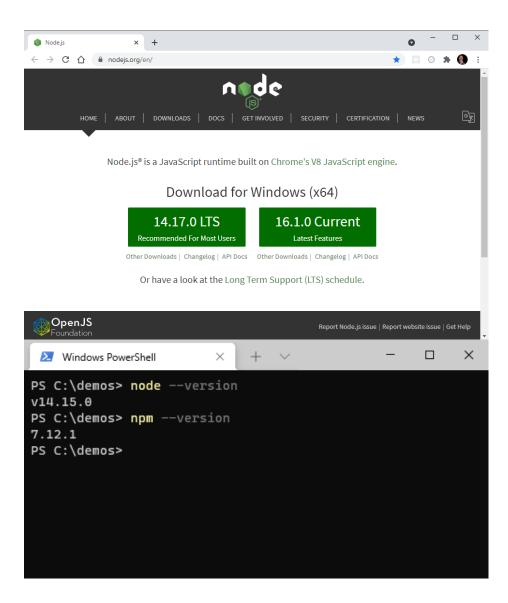
"Typing it drills it into your brain much better than simply copying and pasting it. You're forming new neuron pathways. Those pathways are going to help you in the future. Help them out now!"

Prerequisites

Install Node & NPM
Install the GitHub repository

Install Node.js & NPM

- Minimal:
 - Node version 14
 - NPM version 7

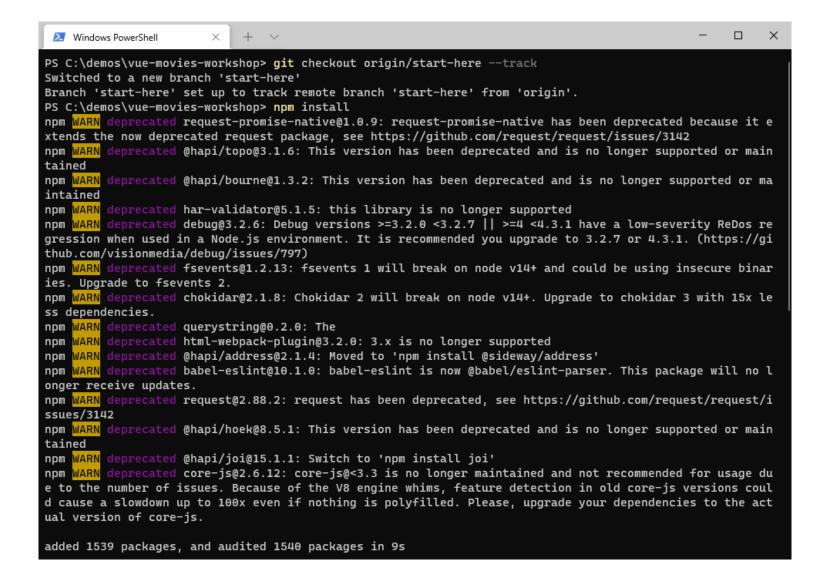


Clone the GitHub Repository

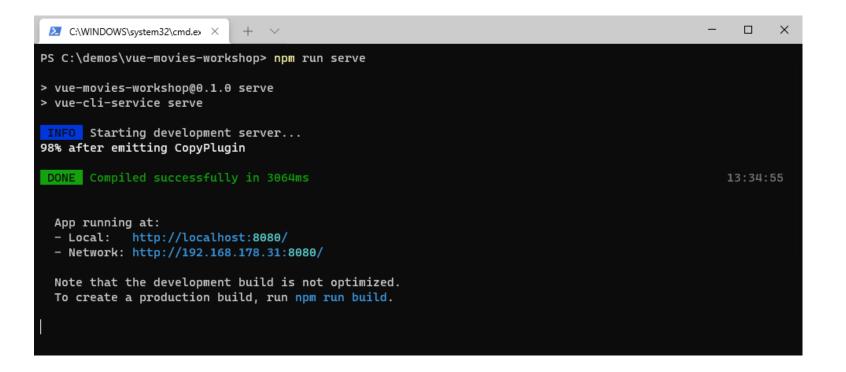
```
PS C:\demos> git clone git@github.com:mauricedb/vue-movies-workshop.git
Cloning into 'vue-movies-workshop'...
remote: Enumerating objects: 292, done.
remote: Counting objects: 100% (292/292), done.
remote: Compressing objects: 100% (137/137), done.
Receiving obj 292 (delta 159), reused 261 (delta 128), pack-reused 0Receiving objects: 80% (234/292)
Receiving objects: 100% (292/292), 314.82 KiB | 1.17 MiB/s, done.
Resolving deltas: 100% (159/159), done.
PS C:\demos>
```



Checkout branch & install NPM packages



Start the application



Following Along



- Repository: https://github.com/mauricedb/vue-movies-workshop
- Slides: http://theproblemsolver.nl/vue-workshop-part-1.pdf

From Vue 2 to Vue 3

From Vue 2 to Vue 3

- Vue 3 is not 100% backward compatible with Vue2
 - Make sure to check the 3rd party Vue libraries for compatibility
- Mostly adding new features
 - Composition API
 - Teleport
 - Multiple root nodes in a template
- But changing or dropping some features
 - Application initialization
 - Reactivity
 - Slot naming
 - Filters
- Vue 3 dropped support for Internet Explorer 11

Vue Migration Build

- A **special version** of Vue 3
 - With support for most Vue 2 features

Workflow

- First upgrade vue-cli
 - vue upgrade
- Uninstall Vue 2
 - npm uninstall vue vue-template-compiler @vue/testutils
- Install Vue 3
 - npm install vue@3
 - npm install -D @vue/compiler-sfc @vue/test-utils@next
- Install the Vue migration build
 - npm install @vue/compat
- Note: There is a Vue CLI command that automates this
 - vue add vue-next
 - Only works on simple applications

Add/Update vue.config.js

```
JS vue.config.js U X
JS vue.config.js > ...
      module.exports = {
         chainWebpack: (config) ⇒ {
           config.resolve.alias.set("vue", "@vue/compat");
    5
           config.module
    6
              .rule("vue")
              .use("vue-loader")
              .tap((options) \Rightarrow \{
   9
                return {
  10
                   ... options,
  11
                  compilerOptions: {
  13
                     compatConfig: {
                       MODE: 2,
  14
                     },
  15
  16
                  },
  17
              });
  18
  19
        },
  20
```

Start the app npm run serve





Home | Top Rated Movies | Popular Movies | About



Welcome to Your Vue.js App

For a guide and recipes on how to configure / customize this project,

🖟 🗴 Elements Console Sources Network Performance Memory Application Security Lighthouse	<u>△</u> 15 🌣 🚼 >
Default levels ▼ No Issues	×
[HMR] Waiting for update signal from WDS	log.js?lafd:24
△ ▶ [Vue warn]: (deprecation CONFIG_OPTION_MERGE_STRATS) config.optionMergeStrategies no longer exposes internal strategies. Use custom merge functions instead.	vue.runtime.esm-bundler.js?1786:1237
▲ r (We warn): (deprecation GLOBAL_POUNT) The global app bootstrapping API has changed: wm.\$mount() and the "el" option have been removed. Use create4pp(RootComponent).mount() instead. Obtails: https://via.webi.org/publichi.	vue.runtime.esm-bundler.js?1786:1237
△ Y (We warn): (deprecation GLGRAL_PROTOTYPE) We, prototype is no longer available in Vue 3. Use app. config.globalProperties instead. Details: https://via.vue/pi.org/guid/schips/attion/global=app.html/pi.org/guid/schips/attion/global=app. This interval in the prototype is no longer available in Vue 3. Use app. config.globalProperties interval inter	vue.runtime.esm-bundler.js?1786:1237
▲ ►[Vue warn]: (deprecation RENDER_FUNCTION) Vue 3's render function API has changed. You can opt-in to the new API with:	vue.runtime.esm-bundler.js?1786:1237
<pre>configureCompat((ReNDER_FUNCTION: false)) (This can also be done per-component via the "compationfig" option.) Details: https://via.vergationfiguredinor.render-function.add.html</pre>	
A F[We warn]: (deprecation INSTANCE_EVENT_MOOKS) "nook:destroyed" lifecycle events are no longer supported. From templates, use the "wnode" prefix instead of "hook:". For example, @hook:destroyed should be changed to @wnode-destroyed. From JavaScript, use Composition API to dynamically register lifecycle hooks. Details: https://www.weis.org/guide/mispation/mode-lifecycle-events.ntml	vue.runtime.esm-bundler.js?1786:1237
△ YUke warn]: (deprecation GLORA_EXTEND) Vue.extend() has been removed in Vue 3. Use defineComponent() instead. Details: Intos://vi>.vue.org.ac/glora/eals.ac/intal8definecomponent	vue.runtime.esm-bundler.js?1786:1237
▲ ►[Vue warn]: (deprecation GLOBAL_PRIVATE_UTIL) Vue.util has been removed. Please refactor to avoid its usage since it was an internal API even in Vue 2.	vue.runtime.esm-bundler.js?1786:1237
△ ►[Vue warn]: (deprecation OPTIONS_DESTROYED) 'destroyed' has been renamed to 'unmounted'.	vue.runtime.esm-bundler.js?1786:1237
▲ ►[Vue warn]: (deprecation OPTIONS_DESTROYED) (2) at (App> at (App>)	vue.runtime.esm-bundler.js?1786:1237
A P(New warm): (deprecation COMPONENT_FUNCTIONAL) Functional component <pre>A plus function in Vue 3. The "functional" option has been removed. NOTE: Before migrating to use plain functions for functional components, first make sure that all async components usage have been migrated and its compat behavior has been disabled. Details: https://viv.vue/s.org/guide/migration/functional-components.html at (App) at (App)</pre>	vue.runtime.esm-bundler.js}1786:1237
A FIVe warn]: (deprecation RENDER_FUNCTION) (2) st «SQUELINK to»" > st «SQUELINK to»" > st «Quellink to»" > st (App> st	yue.runtime.esm-bundler.js?1786:1237
A YUwe warn]: (deprecation OPTIONS_DESTROYED) (3) at <nouterlint to="/"> a</nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint></nouterlint>	vue.runtime.esm-bundler.js?1786:1237
△ *[Vue warn]: (deprecation INSTANCE_SCOPED_SLOTS) vm.\$scopedSlots has been removed. Use vm.\$slots instead. Details: https://www.js.org/guide/migration/slots-unification.html	vue.runtime.esm-bundler.js?1786:1237

Update mount code

main.js



```
src > Js main.js > ...

You, seconds ago | 1 author (You)

import { createApp } from "vue";

import App from "./App.vue";

import router from "./router";

createApp({ ... App, router }).mount("#app");
```

Vue Router 3 to 4

Vue Router 4

- Update Vue router to version 4
 - npm uninstall vue-router
 - npm install vue-router@4

router/index.js

```
JS index.js M X
   1 import { createRouter, createWebHistory } from "vue-router";
     import Home from "../views/Home.vue";
     import MovieList from "../views/MovieList.vue";
     import MovieDetails from "../views/MovieDetails.vue";
   6 const routes = [...
  47 ];
  48
     const router = createRouter({
        history: createWebHistory(process.env.BASE_URL),
  50
  51
        routes,
  52
  53
     export default router;
```

main.js

```
src > Js main.js

You, seconds ago | 1 author (You)

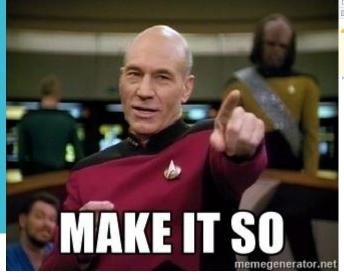
import { createApp } from "vue";

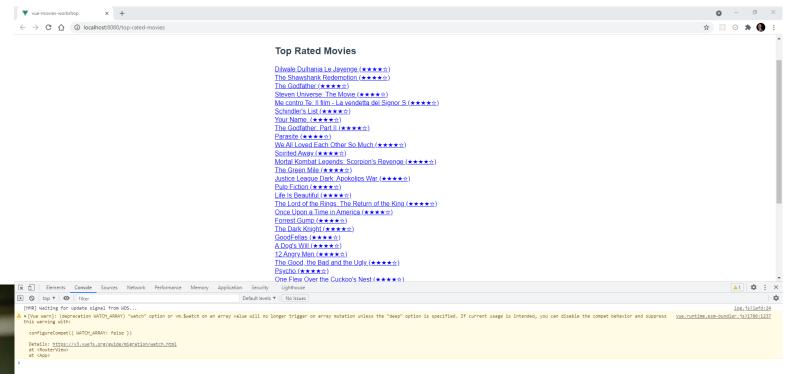
import App from "./App.vue";

import router from "./router";

createApp(App).use(router).mount("#app");
```

Start the app npm run serve

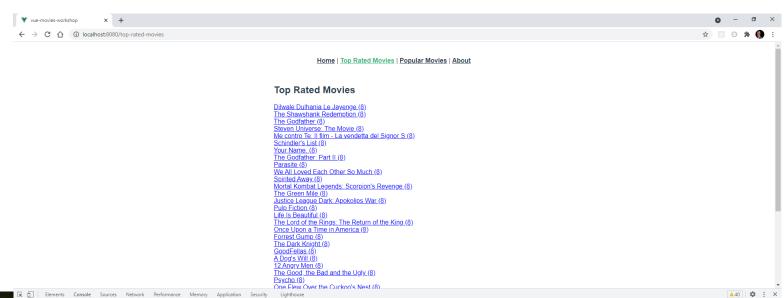


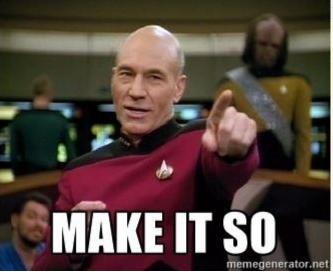


Remove the migration build

Workflow

- Remove the migration build
 - npm uninstall @vue/compat
 - Remove/Update vue.config.js
- Note: The rating stars filter is now broken





► O top ▼ O Filter Default levels ▼ No Issues ▶ [Vue warn]: Property "stars" was accessed during render but is not defined on instance. runtime-core.esm-bundler.js?5c40:38 *|Vow Baffig: Froperty Saffig: Map accessed using refine. **

**A (Baffig: Froperty Saffig: Map accessed using refine. **

**A (Baffig: Froperty Saffig: Map accessed using refine. **

**A (Baffig: Froperty Saffig: Map accessed using refine. **

**A (Baffig: Froperty Saffig: Map accessed using refine. **

**A (Baffig: Froperty Saffig: Map accessed using refine. **

**A (Baffig: Froperty Saffig: Map accessed using refine. **

**A (Baffig: Froperty Saffig: Map accessed using refine. **

**A (Baffig: Froperty Saffig: Map accessed using refine. **

**A (Baffig: Froperty Saffig: Map accessed using refine. **

**A (Baffig: Froperty Saffig: Map accessed using refine. **

**A (Baffig: Froperty Saffig: Map accessed using refine. **

**A (Baffig: Froperty Saffig: Map accessed using refine. **

**A (Baffig: Froperty Saffig: Map accessed using refine. **

**A (Baffig: Froperty Saffig: Map accessed using refine. **

**A (Baffig: Froperty Saffig: Map accessed using refine. **

**A (Baffig: Froperty Saffig: Map accessed using refine. **

**A (Baffig: Froperty Saffig: Map accessed using refine. **

**A (Baffig: Froperty Saffig: Map accessed using refine. **

**A (Baffig: Froperty Saffig: Map accessed using refine. **

**A (Baffig: Froperty Saffig: Map accessed using refine. **

**A (Baffig: Froperty Saffig: Map accessed using refine. **

**A (Baffig: Froperty Saffig: Map accessed using refine. **

**A (Baffig: Froperty Saffig: Map accessed using refine. **

**A (Baffig: Froperty Saffig: Map accessed using refine. **

**A (Baffig: Froperty Saffig: Map accessed using refine. **

**A (Baffig: Froperty Saffig: Map accessed using refine. **

**A (Baffig: Froperty Saffig: Map accessed using refine. **

**A (Baffig: Froperty Saffig: Map accessed using refine. **

**A (Baffig: Froperty Saffig: Map accessed using refine. **

**A (Baffig: Froperty Saffig: Map accessed using refine. **

**A (Baffig: Froperty Saffig: Map accessed using refine. **

**A (Baffig: Froperty Saffig: Map accessed using refine. **

**A (Baffig: Froperty Saffig: Ma at <RouterView> at <App> ▶[Vue warn]: Property "stars" was accessed during render but is not defined on instance. runtime-core.esm-bundler.js?5c40:38 at (RouterLink to= \frame: "MovieDetails", parass: (.)) > at (MovieSard key=7) movie= \frame figure (1) = (1 at <RouterView> at <App> ▶ [Vue warn]: Property "stars" was accessed during render but is not defined on instance. runtime-core.esm-bundler.js?5c40:38 at (Routerlink to= \ (name: "NovieDetails", params: {_-}) > at (NovieCera key=1891 movie= \ (popularity: 23.833, vote_count: 11614, video: false, poster_path: "/78uH8itoSrlExs2YZ5sH8IQk2no.jpg", id: 1891, __} type="top-rated" > at «MovieList moviesUrl="https://the-problem-solver-sample-data.azurewebsites.net/top-rated-movies" title="Top Rated Movies" type="top-rated" ...) at <App> ►[Vue warn]: Property "stars" was accessed during render but is not defined on instance.

at Robuterlink to ► \[\frac{Noneer "Norlederalis", params: (_i) \} \)

at \[\frac{Novigut (avg = 0.00 \)

the \[\frac{Novigut (avg = 0.00 \)
the \[\frac{Novigut (avg = 0.00 \)
the \[\frac{Novigut (avg = 0.00 \)
the \[\frac{Novigut (avg = 0.00 \)
the \[\frac{Novigut (avg = 0.00 \)
the \[\frac{Novigut (avg = 0.00 \)
the \[\frac{Novigut (avg = 0.00 \)
the \[\frac{Novigut (avg = 0.00 \)
the \[\frac{Novigut (avg = 0.00 \)
the \[\frac{Novigut (avg = 0.00 \)
the \[\frac{Novigut (avg = 0.00 \)
the \[\frac{Novigut (avg = 0.00 \)
the \[\frac{Novigut (avg = 0.00 \)
the \[\frac{Novigut (avg = 0.00 \)
the \[\frac{Novigut (avg = 0.00 \)
the \[\frac{Novigut (avg = 0.00 \)
the \[\frac{Novigut (avg = 0.00 \)
the \[\frac{Novigut (avg = 0.00 \)
the \[\frac{Novigut (avg = 0.00 \)
the \[\frac{Novigut (avg = 0.00 \)
the \[\frac{Novigut (avg = 0.00 \)
the \[\frac{Novigut (avg = 0.00 \)
the \[\frac{Novigut (avg = 0.00 \)
the \[\frac{Novigut (avg = 0.00 \)
the \[\frac{Novigut runtime-core.esm-bundler.js?5c40:38 at <RouterView> at <App>

Fix the rating stars

Removed APIs

Removed APIs

- keyCode support as v-on modifiers
- . \$on, \$off and \$once instance methods
- Filters
- Inline templates attributes
- \$children instance property
- propsData option
- \$destroy instance method. Users should no longer manually manage the lifecycle of individual
 Vue components.
- Global functions set and delete, and the instance methods \$set and \$delete. They are no longer required with proxy-based change detection.

Filters

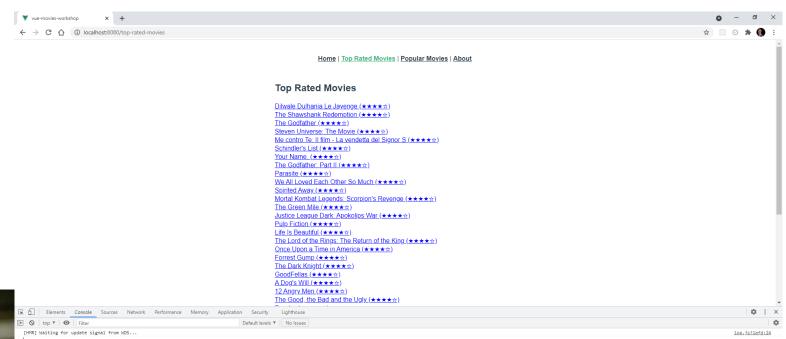
- Filters have been removed from Vue 3
 - Replace with computed property
 - Or with a function call

MovieCard.vue

```
src > views > ♥ MovieCard.vue > {} "MovieCard.vue"
     You, 2 minutes ago | 1 author (You)
  1 <template>
       <div>
         <router-link :to="{ name: 'MovieDetails', params: { id: movie.id, type } }">
            {{ movie.title }}
            <span :title="movie.vote_average"> ({{ voteAsStars }}) 
          ⟨router-link>
       </div>
     </template>
     <script>
     export default {
 12
       props: {
 13
         movie: Object,
 14
         type: String,
 15
 16
       computed: {
 17
         voteAsStars() {
            return "".padEnd(this.movie.vote_average / 2, "★").padEnd(5, "☆");
 18
 19
          },
 20
       },
     };
 21
 22 </script>
```

Computed stars





Reusable components

Reusable components

- Create **reusable components** as building blocks
- Add v-model support where needed
 - Receive the modelValue prop
 - Submit the update:modelValue event

LabeledInput.vue

```
₩ LabeledInput.vue M 🗶
src > views > V LabeledInput.vue > { } "LabeledInput.vue"
     You, seconds ago | 1 author (You)
  1 <template>
        <div>
          <label class="input-label" :for="$attrs.id">{{ label }}
          <input
            type="text"
            class="form-control"
            : value="modelValue"
            @input="$emit('update:modelValue', $event.target.value)"
          />
       </div>
 10
     </template>
 12
 13 <script>
 14 export default {
       props: ["label", "modelValue"],
 15
 16
       emits: ["update:modelValue"],
 17 };
 18 </script>
 19
 20 <style scoped>
 21 .input-label {
       display: inline-block;
       margin-top: 1em;
 23
 24
       margin-bottom: 0.5em;
 25
 26
 27 .form-control {
 28 width: 100%:
```

MovieDetails.vue

```
▼ MovieDetails.vue M ×
rc > views > ♥ MovieDetails.vue > {} "MovieDetails.vue" > � template > � div > � form > � fieldset.fieldset
      You, seconds ago | 1 author (You)
   1 <template>
        <div>
          <div v-if="error">{{ error }}</div>
          <div v-else-if="loading">Loading ... </div>
          <form v-else @submit.prevent="submitForm" @reset="resetForm" novalidate>
             <fieldset <pre>class="fieldset" :disabled="saving">
               <LabeledInput label="Title" v-model="movie.title" />
               <div>
                 <label>
                    <span class="input-label">Overview</span>
  10
                   <textarea v-model="movie.overview" rows="5" class="form-control" />
  11
  12
                 </label>
  13
               </div>
               <LabeledInput</pre>
  15
                  label="Vote average"
  16
                  v-model.number="movie.vote_average"
  17
                  type="number"
               <LabeledInput</pre>
  19
                  label="Release date"
  20
 21
                  v-model="movie.release_date"
 22
                  type="date"
  23
  24
  25
               <div class="button-row">
  26
                  <button type="submit">Save</button>
 27
                 <button type="reset">Reset</button>
               </div>
  28
```

Using inheritAttrs

- Use inheritAttrs to control where additional attributes are added
 - By default at the components root HTML tag
- Use v-bind="\$attrs" to add them to any required HTML tag

LabeledInput.vue



```
▼ LabeledInput.vue M ×
src > views > V LabeledInput.vue > { } "LabeledInput.vue"
     You, seconds ago | 1 author (You)
  1 <template>
        <div>
          <label class="input-label" :for="$attrs.id">{{ label }}</label>
          <input
            type="text"
            class="form-control"
            : value="modelValue"
            ainput="$emit('update:modelValue', $event.target.value)"
  9
            v-bind="$attrs"
 10
        </div>
 11
     </template>
 13
 14 <script>
 15 export default {
        inheritAttrs: false,
 17
        props: ["label", "modelValue"],
        emits: ["update:modelValue"],
 19
     </script>
 21
     <style scoped>
     .input-label {
        display: inline-block;
 24
        margin-top: 1em;
 25
 26
        margin-bottom: 0.5em;
 27
```

Composition API

Composition API

- The Composition API is an alternative way of creating components
 - Doesn't replace the Options API!
- Makes most sense in more complex components
 - But can be used everywhere
- Export a setup() function
 - Receives the props and context as parameters
 - · Returns an object with items available in the template
- Note: There is a <u>Vue 2 plugin</u> for Composition API

Lifecycle Hooks

Lifecycle Hooks

- Import and call with a callback function in the setup() function
- All functions start with on...
 - onMounted()
 - onUnmounted()
 - Etc.
- The setup() function itself replaces the created() hook

Reactivity

Reactivity

- The Vue reactivity is **completely rewritten** in Vue 3
 - Now based on JavaScript proxy objects
- More flexible and fewer edge cases
 - But only works in modern browsers



Reactive Helper Functions

- ref()
- reactive()
- toRefs()
- computed()
- watch()
- watchEffect()

ref()

- Use a ref() to create a reactive variable
 - Use the ref.value in the code
 - Use the ref directly in the template
- Works both for objects and primitive values

reactive()

- Can be used to create a **reactive object**
 - No wrapping .value property
- Using a ref() calls reactive() internally for an object
 - Not often used as ref() achieves almost the same result

computed()

- Creates a reactive value based on other reactive values
 - Automatically recomputes the result when the dependencies update
- Computed values are usually read only
 - But they can be read/write when needed

watch() watchEffect()

- watch() a reactive value and execute when updated
 - The source can be a function or a ref
 - Pass in an array to watch multiple sources
 - Similar as this.\$watch() in the Options API
- There is a new and similar watchEffect() in Vue 3
 - Similar to computed()
 - Often the better option in Vue 3

toRefs()

- Converts a reactive object to an object with reactive properties
 - The toRefs() function turns a new object
- The **props** passed to the setup() function are not reactive
 - With toRefs() you can watch for prop updates

toRaw()

- Turn a reactive object back into a normal object
 - With the toRaw() function

MovieList using the Composition API

MovieList.vue

```
♥ MovieList.vue M 🗙
rc > views > ▼ MovieList.vue > { } "MovieList.vue"
 19 <script>
 20 import { ref, onMounted, watch } from "vue";
 21 import MovieCard from "./MovieCard.vue";
 22
 23 export default {
       components: {
 24
          MovieCard,
 25
 26
 27'>
       props: { ···
 40
 41
       setup(props) {
         const error = ref(null);
 42
 43
          const loading = ref(true);
          const movies = ref([]);
 44
          async function fetchMovies() {
 46
 47
            try {
              const rsp = await fetch(props.moviesUrl);
 48
 49
 50
              if (rsp.ok) {
 51
                movies.value = await rsp.json();
 52
              } else {
 53
                error.value = rsp.statusText ?? "Failed to load data";
            } catch (error) {
              error.value = error?.message ?? "Failed to load data";
 56
            } finally {
 57
              loading.value = false;
 58
 59
```

MovieList.vue



```
rc > views > V MovieList.vue > { } "MovieList.vue"
             const rsp = await fetch(props.moviesUrl);
 49
             if (rsp.ok) {
 50
               movies.value = await rsp.json();
 51
             } else {
 52
                error.value = rsp.statusText ?? "Failed to load data";
 53
           } catch (error) {
             error.value = error?.message ?? "Failed to load data";
 56
 57
           } finally {
             loading.value = false;
 58
 59
 60
 61
         onMounted(() ⇒ fetchMovies());
 62
 63
         watch(
 64
           () ⇒ props.moviesUrl,
 66
           () ⇒ fetchMovies()
 67
         );
 68
 69
         return {
 70
           error,
 71
           loading,
 72
           movies,
         };
 73
 74
       },
 75 };
 76 </script>
```

MovieDetails using the Composition API

```
> views > V MovieDetails.vue > { } "MovieDetails.vue"
35 <script>
   import { ref, onMounted, watch, computed } from "vue";
    import { useRouter } from "vue-router";
    import LabeledInput from "./LabeledInput.vue";
39
40 export default {
41
      components: { LabeledInput },
42
      props: {
43
        id: { type: [Number, String], required: true },
        type: { type: String, required: true },
44
45
      setup(props) {
46
        const router = useRouter();
47
48
        const error = ref(null);
        const loading = ref(true);
50
        const saving = ref(false);
51
        const movie = ref(null);
52
53
54
        const uriTypeFragment = computed(() ⇒
           props.type == "popular" ? "popular-movies" : "top-rated-movies"
56
         );
57
        const movieUrl = computed(
58
59
          () \Rightarrow
              ${process.env.VUE_APP_API_ORIGIN}/${uriTypeFragment.value}/${props.id}
60
         );
61
62
```

```
₩ MovieDetails.vue M X
src > views > ♥ MovieDetails.vue > {} "MovieDetails.vue"
           async function fetchMovie() {
  63
             try {
  64
               loading.value = true;
  65
               const rsp = await fetch(movieUrl.value);
  66
  67
               if (rsp.ok) {
  68
                  movie.value = await rsp.json();
  69
                } else {
  70
                  error.value = rsp.statusText ?? "Failed to load data";
  71
  72
             } catch (error) {
  73
               error.value = error?.message ?? "Failed to load data";
  74
             } finally {
  75
  76
               loading.value = false;
  77
  78
  79
           onMounted(() \Rightarrow fetchMovie());
  80
  81
           watch(movieUrl, () ⇒ fetchMovie());
  82
```

```
▼ MovieDetails.vue M ×
src > views > ♥ MovieDetails.vue > { } "MovieDetails.vue"
          async function saveMovie() {
  84
  85
             try {
  86
               saving.value = true;
               const rsp = await fetch(movieUrl.value, {
 87
  88
                 method: "put",
                 headers: {
  89
                    "Content-Type": "application/json",
  90
  91
                 body: JSON.stringify(movie.value),
  92
  93
               });
  94
  95
               if (rsp.ok) {
                 movie.value = await rsp.json();
  96
               } else {
  97
                 error.value = rsp.statusText ?? "Failed to save data";
  98
 99
             } catch (error) {
100
101
               error.value = error?.message ?? "Failed to save data";
             } finally {
102
               saving.value = false;
103
104
 105
```



```
▼ MovieDetails.vue M X

src > views > ♥ MovieDetails.vue > { } "MovieDetails.vue"
           async function submitForm() {
 107
              await saveMovie();
 108
              if (!error.value) {
 109
                router.push(`/${uriTypeFragment.value}`);
 110
 111
 112
 113
           function resetForm() {
              fetchMovie();
 114
 115
 116
 117
           return {
 118
              error,
              loading,
 119
 120
              saving,
 121
             movie,
              submitForm,
 122
 123
              resetForm,
           };
 124
 125
 126
      </script>
```

Composition Functions

Composition Functions

- Extract functionality into reusable functions
 - Use anything you would use in the setup()
- More reliable than using mixins
 - No accidental naming collisions
- Composition API naming convention
 - Prefix a function with "use"

useFetchData.js

```
1 import { ref, onMounted, watch } from "vue";
   export default function useFetchData(url) {
     const error = ref(null);
      const loading = ref(true);
      const data = ref(null);
      async function fetchData() {
        try {
          const rsp = await fetch(url.value);
10
11
          if (rsp.ok) {
12
13
            data.value = await rsp.json();
          } else {
14
15
            error.value = rsp.statusText ?? "Failed to load data";
16
17
        } catch (error) {
18
          error.value = error?.message ?? "Failed to load data";
19
        } finally {
20
          loading.value = false;
21
22
23
      onMounted(() ⇒ fetchData());
24
25
      watch(url, () ⇒ fetchData());
26
27
      return { error, loading, data, fetchData };
28
29
```

MovieList.vue

```
▼ MovieList.vue M ×
src > views > ♥ MovieList.vue > { } "MovieList.vue" > � style
      <script>
      import MovieCard from "./MovieCard.vue";
     import useFetchData from "../composables/useFetchData";
      import { toRefs } from "@vue/reactivity";
 23
      export default {
        components: {
 26
          MovieCard,
 27
  28 >
        props: { ···
 41
        setup(props) {
 42
 43
          const url = toRefs(props).moviesUrl;
          const { error, loading, data: movies } = useFetchData(url);
 44
  45
          return {
 47
             error,
 48
             loading,
  49
             movies,
          };
  50
  51
  52
      </script>
```



```
▼ MovieDetails.vue M ×
src > views > ♥ MovieDetails.vue > {} "MovieDetails.vue" > � script > ❷ default > � setup > � saveMovie
        setup(props) {
 48
           const router = useRouter();
 49
 50
           const saving = ref(false);
 51
 52
 53
           const uriTypeFragment = computed(() ⇒
 54
             props.type == "popular" ? "popular-movies" : "top-rated-movies"
          );
 55
 57
           const movieUrl = computed(
 58
             () \Rightarrow
                 ${process.env.VUE_APP_API_ORIGIN}/${uriTypeFragment.value}/${props.id}
 59
           );
 60
 61
 62
           const {
 63
             error,
 64
             loading,
 65
             data: movie,
 66
             fetchData: fetchMovie,
 67
           } = useFetchData(movieUrl);
 68
 69
           async function saveMovie() {
```

watchEffect()

watchEffect()

- Automatically watches all dependencies
 - Even deep dependencies
 - Run the watch function immediately

useFetchData.js



```
JS useFetchData.js X
src > composables > JS useFetchData.js > ...
      You, 2 days ago | 1 author (You)
     import { ref, watchEffect } from "vue";
       export default function useFetchData(url) {
         const error = ref(null);
         const loading = ref(true);
         const data = ref(null);
         async function fetchData() { ...
   8 >
  22
  23
         watchEffect(() ⇒ fetchData());
  24
  25
         return { error, loading, data, fetchData };
  26
  27
```

Conclusion

- Migrating from Vue 2 to 3 is usually not hard
 - The migration build helps a lot
- Most external libraries need to be updated
 - Not all are compatible with Vue 3 yet
- The Composition API is an improvement over the Options API
 - But the Options API still works
 - Reuse code with composition functions
- The Reactive API has been completely rewritten
 - More flexible and fewer edge cases

Maurice de Beijer

@mauricedb

maurice.de.beijer @gmail.com

