

Setting Up

Take the first steps into learning how to validate Vue 3 forms with Vee-Validate

4

Create a free account to unlock it.



Unlock Content

Validating at form level

Learn how to set up a validation schema using vee-validate's useField method

Unlock this lesson by subscribing to a plan.



Unlock Content

Submitting bigger forms

Learn how to validate a big form with Vee-validate by leveraging the power of the `FormComposition` function.

Unlock this lesson by subscribing to a plan.



Unlock Content

Using YUP for validations

Learn how to leverage the power of YUP's validation methods to validate your forms with *Yup*.

Unlock this lesson by subscribing to a plan.



Unlock Content

Lazy Validation

Learn how to lazy validate a field with vee-validate on a Vue 3 form

Unlock this lesson by subscribing to a plan.



Unlock Content

Why Vee-Validate?

A form is only as good as the data it collects from your user.

Allowing your users to submit forms with required fields left blank, or not checking that an input's content is what you are expecting, can lead to unusable data and unreliable databases.

Simply put, a form that does not collect the data you need, as you need it, is not serving its full purpose.

In this course we are going to learn how to validate any form inside a Vue 3 application by leveraging the power of `Vee-Validate`.

Why use a library? And why Vee-Validate?

Form validation can get very complex, very quickly. To do it right, you'd need to hand-craft everything from connecting your inputs into a central parser to check the validity of the data, to creating custom rules, and handling different input states.

While you could do all this from scratch, your precious development time is better spent on building application logic, polishing the user experience of your form, and ensuring it's highly accessible. By implementing a plug-and-play solution like Vee-Validate, you'll free up more time that can be devoted to these considerations.

What you'll learn

- **Validate single fields and full forms** - Throughout this course we will learn how to use Vee-Validate to validate both a single input in our applications, and then grow that to validating a full form with a defined set of rules.
- **Handle errors** - Next, we will learn together how to check and present validation errors to our users as they progress through filling out or forms.
- **Key API differences** - We will explore the different ways of obtaining information about the state of our forms, for example, how to tell when an error is present within a field with directly vs extracting all the form's errors at once.
- **YUP** - Later in the course, we will explore the benefits of using an out of the box validation methods library, YUP, and it's benefits as opposed to writing your own.
- **Best practices** - We will implement and understand together the libraries best practices such as how to handle form submit events or the benefits of lazy vs aggressive validation.

So if you're ready to feel confident validating any form with this handy validation library, I'll see you in the rest of the course.