

```
1 <template>
2   <div>
3     <div class="about">
4       <h1>Vee Validate</h1>
5     </div>
6
7     <v-card class="mx-auto pl-5 pr-5 pt-5" max-width="344" outlined>
8       <ValidationObserver ref="observer" v-slot="{ valid }">
9         <ValidationProvider
10          name="name"
11          ref="name"
12          rules="required|alpha|max:20"
13          v-slot="{ errors }"
14          :bails="false"
15        >
16          <v-text-field v-model="name" label="Name" outlined></v-text-field>
17
18          <ul style="color:red" class="overline text-left">
19            <li v-for="(error, index) in errors" :key="index">
20              <span>{{ error }}</span>
21            </li>
22          </ul>
23        </ValidationProvider>
24
25        <ValidationProvider
26          name="password"
27          ref="password"
28          rules="required|customPassword|min:8"
29          v-slot="{ errors }"
30          :bails="false"
31        >
32          <v-text-field
33            class="mt-5"
34            v-model="password"
35            :append-icon="show1 ? 'visibility' : 'visibility_off'"
36            :type="show1 ? 'text' : 'password'"
37            label="Password"
38            counter
39            @click:append="show1 = !show1"
40            outlined
41          ></v-text-field>
42
43          <ul style="color:red" class="overline text-left">
44            <li v-for="(error, index) in errors" :key="index">
45              <span>{{ error }}</span>
46            </li>
47          </ul>
48        </ValidationProvider>
49        <v-card-actions>
50          <v-spacer></v-spacer>
51          <!-- <v-btn :disabled="!valid" @click="signIn()" text>Login</v-btn> -->
52          <v-btn @click="signIn()" text>Login</v-btn>
53        </v-card-actions>
54      </ValidationObserver>
55    </v-card>
56  </div>
57 </template>
58
59 <script>
60 import {
```

```
61   ValidationProvider,
62   extend
63 } from "vee-validate/dist/vee-validate.full";
64
65 import { ValidationObserver } from "vee-validate/dist/vee-validate.full";
66
67 // can customize default error messages
68 extend("required", {
69   message: (field, values) => "You must enter a " + `${field}` + " value"
70 });
71
72 // create custom error message for custom rule
73 var errorMessage =
74   " min length 8 chars, and must include 1 lower-case, upper-case, numeral and
special character (!@%*-)";
75
76 // create custom rule
77 extend("customPassword", {
78   message: field => "The " + `${field}` + errorMessage,
79   validate: value => {
80     var notTheseChars = /["' ? & / < > \s ] /;
81     var mustContainTheseChars = /^(?=.*?[A-Z])(?=.*?[a-z])(?=.*?[0-9])(?=.*?
[#!@%*-]).{8,}$/;
82
83     var containsForbiddenChars = notTheseChars.test(value);
84     var containsRequiredChars = mustContainTheseChars.test(value);
85
86     if (containsRequiredChars && !containsForbiddenChars) {
87       return true;
88     } else {
89       if (containsForbiddenChars) {
90         errorMessage =
91           ' contains forbidden characters: " ' + " ' ? & / < > or space";
92       } else {
93         errorMessage =
94           " min length 8 chars, and must include 1 lower-case, upper-case, numeral
and special character ($@%!*#?&)";
95       }
96       return false;
97     }
98   }
99 });
100
101 export default {
102   components: {
103     ValidationProvider,
104     ValidationObserver
105   },
106   data() {
107     return {
108       show1: false,
109       name: "",
110       password: ""
111     };
112   },
113   computed: {},
114   created() {},
115   methods: {
116     async signIn() {
117       const isValid = await this.$refs.observer.validate();
```

```
118     if (!isValid) {
119         console.log("is not valid");
120         alert("Please fix errors first!");
121     } else {
122         console.log("is valid");
123         console.log(this.name + " signed in with password " + this.password);
124         // reset fields
125         this.name = "";
126         this.password = "";
127
128         // reset validation
129         // You should call it on the next frame
130         requestAnimationFrame(() => {
131             this.$refs.observer.reset();
132         });
133     }
134 }
135 }
136 };
137 </script>
138
```