

Sommaire

Ajouter un formulaire de contact à votre site Vitepress	3
Création du formulaire	3
Intégration du composant sur le site	7
Côté serveur	8
Licence	9

Ajouter un formulaire de contact à votre site Vitepress

Création du formulaire

Nous allons créer un composant Vue pour notre formulaire de contact.

Ce composant est à placer dans `.vitepress/theme/components/ContactForm.vue`:

```
1 <script setup>
2 import { ref, computed } from 'vue'
3 import { useData } from 'vitepress'
4
5 const formData = ref({
6   name: '',
7   email: '',
8   subject: '',
9   message: ''
10 })
11
12 const isSubmitting = ref(false)
13 const isSuccess = ref(false)
14 const errorMessage = ref('')
15
16 const isValidEmail = (email) => {
17   const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/
18   return emailRegex.test(email)
19 }
20
21 const isFormValid = computed(() => {
22   return formData.value.name.trim() !== '' &&
23     formData.value.email.trim() !== '' &&
24     isValidEmail(formData.value.email) &&
25     formData.value.subject.trim() !== '' &&
26     formData.value.message.trim() !== ''
27 })
28
29 const submitForm = async () => {
30   errorMessage.value = ''
31   if (!formData.value.name.trim()) {
32     errorMessage.value = 'Name is required'
33     return
34   }
35   if (!formData.value.email.trim()) {
36     errorMessage.value = 'Email address is required'
37     return
38   }
39   if (!isValidEmail(formData.value.email)) {
40     errorMessage.value = 'Email address is invalid'
41     return
42   }
```

```

43   if (!formData.value.subject.trim()) {
44     errorMessage.value = 'Message subject is required'
45     return
46   }
47   if (!formData.value.message.trim()) {
48     errorMessage.value = 'Message is required'
49     return
50   }
51   isSubmitting.value = true
52   try {
53     const formDataToSend = new FormData()
54     formDataToSend.append('name', formData.value.name)
55     formDataToSend.append('email', formData.value.email)
56     formDataToSend.append('subject', formData.value.subject)
57     formDataToSend.append('message', formData.value.message)
58     const response = await fetch('/sendmail.php', {
59       method: 'POST',
60       body: formDataToSend
61     })
62     const message = await response.text()
63     if (response.ok) {
64       isSuccess.value = true
65       errorMessage.value = ''
66     } else {
67       errorMessage.value = message || 'An error occurred. Please try
68         again.'
69     }
70   } catch (error) {
71     errorMessage.value = 'Network error'
72   } finally {
73     isSubmitting.value = false
74   }
75 </script>
76
77 <template>
78   <div class="contact-form">
79     <form @submit.prevent="submitForm">
80       <div class="form-group">
81         <label for="name">Name <span class="required">*</span></label>
82         <input
83           id="name"
84           v-model="formData.name"
85           type="text"
86           :disabled="isSuccess"
87           placeholder="Your name"
88         />
89       </div>
90       <div class="form-group">
91         <label for="email">Email <span class="required">*</span></label>

```

```

92     <input
93         id="email"
94         v-model="formData.email"
95         type="email"
96         :disabled="isSuccess"
97         placeholder="Your email address"
98     />
99 </div>
100 <div class="form-group">
101     <label for="subject">Subject <span class="required">*</span></
        label>
102     <input
103         id="subject"
104         v-model="formData.subject"
105         type="text"
106         :disabled="isSuccess"
107         placeholder="Message subject"
108     />
109 </div>
110 <div class="form-group">
111     <label for="message">Message <span class="required">*</span></
        label>
112     <textarea
113         id="message"
114         v-model="formData.message"
115         :disabled="isSuccess"
116         rows="5"
117         placeholder="Your message"
118     ></textarea>
119 </div>
120 <!-- Error message area -->
121 <div v-if="errorMessage" class="error-message">
122     {{ errorMessage }}
123 </div>
124 <!-- Success message area -->
125 <div v-if="isSuccess" class="success-message">
126     Thank you! Your message has been sent successfully.
127 </div>
128 <button
129     v-if="!isSuccess"
130     type="submit"
131     :disabled="!isFormValid || isSubmitting || isSuccess"
132     class="submit-button"
133 >
134     {{ isSubmitting ? 'Sending...' : 'Send message' }}
135 </button>
136 </form>
137 </div>
138 </template>
139
140 <style scoped>

```

```
141 .contact-form {
142   max-width: 600px;
143   margin: 2rem auto;
144 }
145 .form-group {
146   margin-bottom: 1.5rem;
147 }
148 label {
149   display: block;
150   margin-bottom: 0.5rem;
151   font-weight: 500;
152   color: var(--vp-c-text-1);
153 }
154 .required {
155   color: #e74c3c;
156 }
157 input,
158 textarea {
159   width: 100%;
160   padding: 0.75rem;
161   border: 1px solid var(--vp-c-divider);
162   border-radius: 4px;
163   font-size: 1rem;
164   font-family: inherit;
165   background-color: var(--vp-c-bg);
166   color: var(--vp-c-text-1);
167   transition: border-color 0.2s;
168 }
169 input:focus,
170 textarea:focus {
171   outline: none;
172   border-color: var(--vp-c-brand);
173 }
174 input:disabled,
175 textarea:disabled {
176   opacity: 0.6;
177   cursor: not-allowed;
178   background-color: var(--vp-c-bg-soft);
179 }
180 textarea {
181   resize: vertical;
182 }
183 .error-message {
184   padding: 0.75rem 1rem;
185   margin-bottom: 1rem;
186   background-color: #fee;
187   border-left: 4px solid #e74c3c;
188   color: #c0392b;
189   border-radius: 4px;
190 }
191 .success-message {
```

```

192 padding: 0.75rem 1rem;
193 margin-bottom: 1rem;
194 background-color: #d4edda;
195 border-left: 4px solid #28a745;
196 color: #155724;
197 border-radius: 4px;
198 font-weight: 500;
199 }
200 .submit-button {
201   width: 100%;
202   padding: 0.75rem 1.5rem;
203   background-color: var(--vp-c-brand-1);
204   color: white;
205   border: none;
206   border-radius: 4px;
207   font-size: 1rem;
208   font-weight: 500;
209   cursor: pointer;
210   transition: background-color 0.2s;
211 }
212 .submit-button:hover:not(:disabled) {
213   background-color: var(--vp-c-brand-2);
214 }
215 .submit-button:disabled {
216   opacity: 0.6;
217   cursor: not-allowed;
218 }
219 </style>

```

Intégration du composant sur le site

Éditez `.vitepress/theme/index.js` pour ajouter:

```

1 enhanceApp({ app }) {
2   app.component('ContactForm', ContactForm)
3 }

```



Note: Si vous avez déjà `enhanceApp` dans ce fichier, par exemple sous la forme `enhanceApp({ router })`, complétez-le en y intégrant l'entrée ci-dessus et en ajoutant `app` comme suit: `enhanceApp({ app, router })`.

À présent, vous pouvez intégrer le composant dans n'importe quelle page Markdown comme ceci:

```

1 # Contact
2
3 Écrivez-moi avec le formulaire ci-dessous:

```

```

4
5 <ContactForm />

```

Côté serveur

Pour l'envoi effectif d'emails, nous allons utiliser le script PHP ci-dessous.

Dans l'en-tête, vous devez configurer vos coordonnées en modifiant `CONTACT_INFO`: - **to_name** et **to_email**: votre nom, et votre adresse email sur laquelle vous souhaitez recevoir les messages de vos visiteurs, - **cc_name** et **cc_email**: nom et adresse email de copie carbone des messages (facultatif), - **site_email**: l'adresse depuis laquelle votre site vous envoie les messages (vous pouvez mettre une adresse qui n'existe pas, comme `no-reply@votre-domaine.tld`).



Note: Votre site vous enverra les messages des visiteurs en son nom, avec l'email **site_email** que vous aurez configuré. Cependant il configure un champ **Reply-To** dans les emails, qui vous permettra de répondre aux messages en toute transparence.

Ce script est à placer dans `public/sendmail.php`:

```

1 <?php
2
3 $CONTACT_INFO['to_email'] = 'me@example.com';
4 $CONTACT_INFO['to_name'] = 'My name';
5 $CONTACT_INFO['cc_email'] = '';
6 $CONTACT_INFO['cc_name'] = '';
7 $CONTACT_INFO['site_email'] = 'My website <no-reply@example.com>';
8
9 header("Content-Type: text/plain");
10
11 $from_name = trim($_POST['name']);
12 $from_email = trim($_POST['email']);
13 $subject = trim($_POST['subject']);
14 $message = trim($_POST['message']);
15
16 $regex_head = '/[\n\r]/';
17 $regex_mail = '/^[+.\w]{1,64}@[+.\w]{1,64}\.[+.\w]{2,6}$/i';
18
19 if (empty($from_name)
20     || empty($from_email)
21     || empty($subject)
22     || empty($message)) {
23     http_response_code(400);
24     echo 'Required fields must be filled';
25 } else if (!preg_match($regex_mail, $from_email)) {
26     http_response_code(400);

```

```

27     echo 'Invalid email address';
28 } else if (preg_match($regex_head, $from_name)
29 || preg_match($regex_head, $from_email)
30 || preg_match($regex_head, $subject)) {
31     http_response_code(400);
32     echo 'Malformed data';
33 } else {
34     $to = $CONTACT_INFO['to_name']
35     . '<'. $CONTACT_INFO['to_email']. '>';
36     $headers = 'From: ' . $CONTACT_INFO['site_email'] . "\r\n";
37     $headers .= 'Reply-To: ' . $from_name
38     . ' <'. $from_email . '>' . "\r\n";
39     if (!empty($CONTACT_INFO['cc_email'])) {
40         $headers .= 'Cc: ' . $CONTACT_INFO['cc_name']
41         . ' <'. $CONTACT_INFO['cc_email']. '>' . "\r\n";
42     }
43     $headers .= "MIME-Version: 1.0\r\n";
44     $headers .= "Content-Type: text/plain; charset=UTF-8\r\n";
45     $headers .= "Content-Transfer-Encoding: base64\r\n";
46
47     $contents = vsprintf('
48 %s
49
50 -----
51 %s <%s>', array(
52     $message, $from_name, $from_email
53 ));
54     $contents = chunk_split(base64_encode($contents));
55     if(!mail($to, $subject, $contents, $headers)) {
56         http_response_code(500);
57         echo 'Unable to send message';
58     }
59 }
60
61 ?>

```

Vous pouvez si besoin modifier le composant Vue ainsi que le script PHP pour leur adjoindre une protection anti-spam.

Licence

Tout le code présenté sur cette page est sous [licence CC0 1.0 Universal](#).

Vous pouvez le copier, le modifier, le distribuer et l'utiliser, même à des fins commerciales, sans demander la permission.