

Sommaire

Adding a Contact Form to Your VitePress Site	3
Creating the Form	3
Integrating the Component on Your Site	7
Server Side	8
License	9

Adding a Contact Form to Your VitePress Site

Creating the Form

We're going to create a Vue component for our contact form.

This component should be placed in `.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       </div>
93     </form>
94   </div>
95 </template>
```

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

Integrating the Component on Your Site

Edit `.vitepress/theme/index.js` to add:

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



Note: If you already have `enhanceApp` in this file, for example in the form `enhanceApp({ router })`, complete it by integrating the above entry and adding `app` as follows: `enhanceApp({ app, router })`.

Now you can integrate the component into any Markdown page like this:

```
1 # Contact
2
```

```

3 Write to me using the form below:
4
5 <ContactForm />

```

Server Side

For the actual sending of emails, we'll use the PHP script below.

In the header, you need to configure your contact information by modifying `CONTACT_INFO`: - **to_name** and **to_email**: your name and email address where you want to receive messages from your visitors, - **cc_name** and **cc_email**: name and email address for carbon copy of messages (optional), - **site_email**: the address from which your site sends you messages (you can use a non-existent address, such as `no-reply@your-domain.tld`).



Note: Your site will send you visitor messages on its behalf, using the **site_email** that you configured. However, it configures a `Reply-To` field in emails, which will allow you to respond to messages seamlessly.

This script should be placed in `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 ?>

```

You can modify the Vue component as well as the PHP script if needed to add anti-spam protection.

License

All code presented on this page is licensed under [CC0 1.0 Universal](#).

You can copy, modify, distribute and use it, even for commercial purposes, without asking permission.