Test Email-Versand

|  |
| --- |
| namespace Tests\Feature;  use Tests\TestCase;  use Illuminate\Support\Facades\Mail;  use App\Mail\ApplicationMail;  use Illuminate\Http\UploadedFile;  class MailTest extends TestCase  {      /\*\*       \* Test Email       \*/      public function test\_Mail(): void      {          Mail::fake();          $pdfData = UploadedFile::fake()->create('fake.pdf');          $to\_email = "i@onm.de";          $mailable = new ApplicationMail(              'Testfirstname',              'Testlastname',              'Testemail',              'Testphone',              'Testhtml\_css',              'Testphp',              'Testmysql',              'Testjs',              'TestfrontendFrameworks',              'Testwordpress',              'Testtypo3',              $pdfData,              'TestbonusText',              'Testjobapplication'          );          Mail::to($to\_email)->send($mailable);          $mailable->assertTo('i@onm.de');          $mailable->assertSeeInOrderInHtml([              'Testjobapplication',              'Testfirstname',              'Testlastname',              'Testemail', 'Testphone',              'Testhtml\_css',              'Testphp',              'Testmysql',              'Testjs',              'TestfrontendFrameworks',              'Testwordpress',              'Testtypo3',              'TestbonusText'          ]);      }  } |

Beispiel für Unit Tests

|  |
| --- |
| /\*\*   \* @vitest-environment happy-dom   \*/  import { describe, it, expect } from 'vitest'  import { mount } from '@vue/test-utils'  import ContentHeader from './ContentHeader.vue'  describe('bigtext test', () => {      it('should render', () => {          const wrapper = mount(ContentHeader, {              propsData:{                  textAlignCenter: 'true'              }          })          expect(wrapper.find('h3.textAlignCenter').exists()).toBeTruthy()      })  }) |