

report.html

Report generated on 22-Aug-2024 at 11:17:33 by `pytest-html` v4.1.1

Environment

Python	3.12.3
Platform	Windows-10-10.0.19045-SP0
Packages	<ul style="list-style-type: none">• <code>pytest</code>: 8.3.2• <code>pluggy</code>: 1.5.0
Plugins	<ul style="list-style-type: none">• <code>html</code>: 4.1.1• <code>metadata</code>: 3.1.1
JAVA_HOME	C:\Program Files\Java\jdk-17

Summary




9 tests took 00:01:33.

(Un)check the boxes to filter the results.

☒ 4 Failed, ☒ 5 Passed, ☒ 0 Skipped, ☒ 0 Expected failures, ☒ 0 Unexpected passes, ☒ 0 Errors, ☒ 0 Reruns

[Show all details](#) / [Hide all details](#)

Result	Test	Duration	Links
Failed	Tests/TestTenantCreation.py::TestTenantCreation::test_tenantCreation_with_invalidData[firefox-712-24-vivekas@gmail.com-viveka]	00:00:08	
<pre>self = <Tests.TestTenantCreation.TestTenantCreation object at 0x00000222E0EF1340>, Name = 712, MoId = 24, MoMail = 'vivekas@gmail.com', DomainName = 'viveka'</pre> <div>expand [+]</div> <pre>@pytest.mark.parametrize("Name,MoId,MoMail,DomainName",get_data_from_excel(r"C:\Users\HP\PycharmProjects\BUILDER-OSS2.0\TestFiles\TenantCreation.xlsx", "Sheet1")) def test_tenantCreation_with_invalidData(self,Name,MoId,MoMail,DomainName): self.tenantcreat = TenantCreation(self.driver) message = self.tenantcreat.tenantCreation_with_invalidData(Name,MoId,MoMail,DomainName) message1=self.tenantcreat.tenantCreation_with_invalidData_MoId() message2 = self.tenantcreat.tenantCreation_with_invalidData_MoMail() message3 = self.tenantcreat.tenantCreation_with_invalidData_DomainName() > assert (message.__contains__("Please fill out this field.") or message1.__contains__("Please fill out this field.") or message2.__contains__("Please fill out this field.") or message3.__contains__("Please fill out this field.)) E AssertionError: assert (False or False or False or False)</pre>			
Failed	Tests/TestTenantCreation.py::TestTenantCreation::test_tenantCreation_with_invalidData[firefox-mahi-sriyas-madhuram@gmail.com-madhuka]	00:00:08	
<pre>self = <Tests.TestTenantCreation.TestTenantCreation object at 0x00000222E0EF13D0>, Name = 'mahi', MoId = 'sriyas', MoMail = 'madhuram@gmail.com', DomainName = 'madhuka'</pre> <div>expand [+]</div> <pre>@pytest.mark.parametrize("Name,MoId,MoMail,DomainName",get_data_from_excel(r"C:\Users\HP\PycharmProjects\BUILDER-OSS2.0\TestFiles\TenantCreation.xlsx", "Sheet1")) def test_tenantCreation_with_invalidData(self,Name,MoId,MoMail,DomainName): self.tenantcreat = TenantCreation(self.driver) message = self.tenantcreat.tenantCreation_with_invalidData(Name,MoId,MoMail,DomainName) message1=self.tenantcreat.tenantCreation_with_invalidData_MoId() message2 = self.tenantcreat.tenantCreation_with_invalidData_MoMail() message3 = self.tenantcreat.tenantCreation_with_invalidData_DomainName() > assert (message.__contains__("Please fill out this field.") or message1.__contains__("Please fill out this field.") or message2.__contains__("Please fill out this field.") or message3.__contains__("Please fill out this field.)) E AssertionError: assert (False or False or False or False)</pre>			
Failed	Tests/TestTenantCreation.py::TestTenantCreation::test_tenantCreation_with_invalidData[firefox-sriv-26-sivas@gmail.c-sriv]	00:00:08	
<pre>self = <Tests.TestTenantCreation.TestTenantCreation object at 0x00000222E0EF07A0>, Name = 'sriv', MoId = 26, MoMail = 'sivas@gmail.c', DomainName = 'sriv'</pre> <div>expand [+]</div> <pre>@pytest.mark.parametrize("Name,MoId,MoMail,DomainName",get_data_from_excel(r"C:\Users\HP\PycharmProjects\BUILDER-OSS2.0\TestFiles\TenantCreation.xlsx", "Sheet1")) def test_tenantCreation_with_invalidData(self,Name,MoId,MoMail,DomainName): self.tenantcreat = TenantCreation(self.driver) message = self.tenantcreat.tenantCreation_with_invalidData(Name,MoId,MoMail,DomainName) message1=self.tenantcreat.tenantCreation_with_invalidData_MoId() message2 = self.tenantcreat.tenantCreation_with_invalidData_MoMail() message3 = self.tenantcreat.tenantCreation_with_invalidData_DomainName() > assert (message.__contains__("Please fill out this field.") or message1.__contains__("Please fill out this field.") or message2.__contains__("Please fill out this field.") or message3.__contains__("Please fill out this field.)) E AssertionError: assert (False or False or False or False)</pre>			
Failed	Tests/TestTenantCreation.py::TestTenantCreation::test_tenantCreation_with_invalidData[firefox-vijayan-27-ijay@gmail.com-679]	00:00:08	
<pre>self = <Tests.TestTenantCreation.TestTenantCreation object at 0x00000222E0EF06B0>, Name = 'vijayan', MoId = 27, MoMail = 'ijay@gmail.com', DomainName = 679</pre> <div>expand [+]</div>			

Result 	Test	Duration	Links
	<pre> OSS2.0\TestFiles\TenantCreation.xlsx", "Sheet1")) def test_tenantCreation_with_invalidData(self, Name, MoId, MoMail, DomainName): self.tenantcreat = TenantCreation(self.driver) message = self.tenantcreat.tenantCreation_with_invalidData(Name, MoId, MoMail, DomainName) message1=self.tenantcreat.tenantCreation_with_invalidData_MoId() message2 = self.tenantcreat.tenantCreation_with_invalidData_MoMail() message3 = self.tenantcreat.tenantCreation_with_invalidData_DomainName() > assert (message.__contains__("Please fill out this field.") or message1.__contains__("Please fill out this field.") or message2.__contains__("Please fill out this field.") or message3.__contains__("Please fill out this field.")) E AssertionError: assert (False or False or False or False) </pre>		
Passed	Tests/TestTenantCreation.py::TestTenantCreation::test_validate_title[firefox]	00:00:07	
Passed	Tests/TestTenantCreation.py::TestTenantCreation::test_verify_TenantButton_is_displayed[firefox]	00:00:07	
Passed	Tests/TestTenantCreation.py::TestTenantCreation::test_validate_TenantCreation_without_credentials[firefox]	00:00:12	
Passed	Tests/TestTenantCreation.py::TestTenantCreation::test_tenantCreation_with_validData[firefox]	00:00:18	
Passed	Tests/TestTenantCreation.py::TestTenantCreation::test_tenantCreation_with_DuplicATEDData[firefox]	00:00:18	