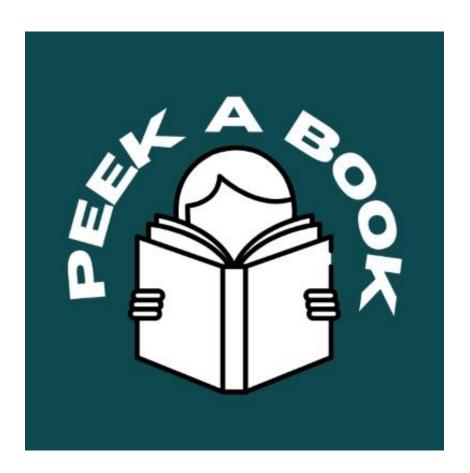


TEST EXECUTION REPORT DOCUMENT "Peek A Book"



Versione	1.0
Data	19/08/2024
Presentato da	lacomino Domenico, De Luca Ciro



Revision History

Data	Versione	Descrizione	Autori
19/08/2024	1.0	Stesura completa	Iacomino Domenico, De Luca Ciro

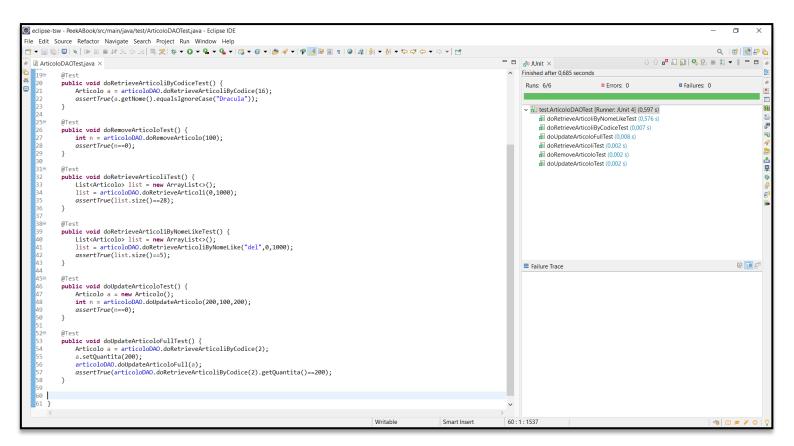
<u>Indice</u>

1.	ArticoloDAOTest	4
	LibroDAOTest	
3.	UtenteDAOTest	6
	OrdineDAOTest	
5.	Test Login	8
6.	Test Registrazione	11
7.	Test Aggiunta prodotti al catalogo	18
8.	Test Ricerca titoli nel catalogo	23



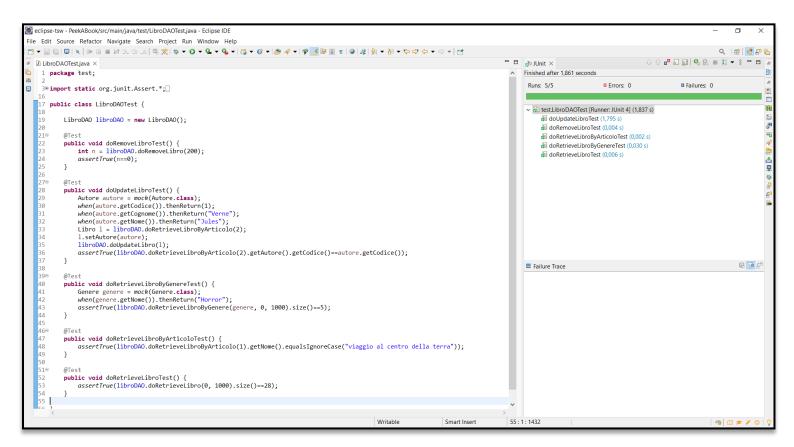


1. ArticoloDAOTest





2. LibroDAOTest





3. UtenteDAOTest

```
File Edit Source Refactor Navigate Search Project Run Window Help
 Q 🔡 😭 🗗 🦫
                                                                                                                                                                                            □ □ Junit ×
                                                                                                                                                                                                                                                   UtenteDAOTest.java ×
                                                                                                                                                                                                   Finished after 1,928 seconds
      3⊕import static org.junit.Assert.*;[.
                                                                                                                                                                                                    Runs: 6/6 Errors: 0
                                                                                                                                                                                                                                                               ☐ Failures: 0
    17 public class UtenteDAOTest {
             UtenteDAO utenteDAO = new UtenteDAO();

→ 

☐ UtenteDAOTest [Runner: JUnit 5] (1,846 s)

    ■ doDeleteUtenteTest (1,790 s)

    ■ doRetrieveByUsernamePasswordTest (0.044 s)

             public void doRetrieveIdByUsernameTest() {
    assertTrue(utenteDAO.doRetrieveIdByUsername("mimmo")==1);
                                                                                                                                                                                                          doRetrieveldByUsernameTest (0,002 s)
                                                                                                                                                                                                          ☐ doRetrieveByIdTest (0,002 s)
☐ doRetrieveAIIUtentiTest (0,003 s)
            @Test
public void doDeleteUtenteTest() {
    Utente utente = mock(Utente.class);
    when(utente.getId()) thenReturn(1000);
    assertTrue(utenteDAO.doDeleteUtente(utente)==0);
             public void doRetrieveAllUtentiTest() {
    assertTrue(utenteDAO.doRetrieveAllUtenti(0, 1000).size()==6);
             public void doRetrieveByIdTest() {
    assertTrue(utenteDAO.doRetrieveById(1).getEmail().equalsIgnoreCase("d.iacomino@studenti.unisa.it"));
                                                                                                                                                                                                                                                                                       9 7 s
                                                                                                                                                                                                   Failure Trace
              public void doVerifyDuplicateTest() {
   Utente utente = mock(Utente.class);
   when(utente.getUsename()).thenReturn("mimmo");
   assertTrue(utenteDAO.doVerifyDuplicate(utente));
              public void doRetrieveByUsernamePasswordTest() {
                  Utente utente = new Utente();
String password = "Domenico_01";
String username = "mimmo3";
utente = utenteDAO.doRetrieveByUsernamePassword(username, password);
assertTrue(utente.getEmail().equalsIgnoreCase("mimmo.iac@hotmail.it"));
```



4. Ordine DAOTest

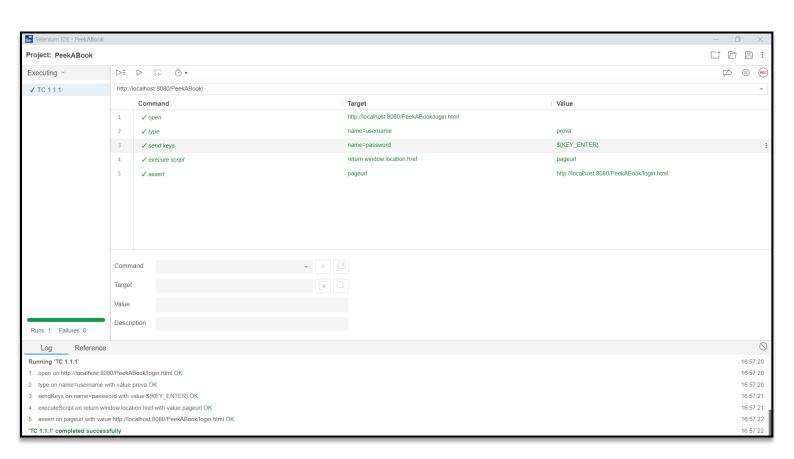
```
geclipse-tsw - PeekABook/src/main/java/test/OrdineDAOTest.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
    Q : E | 😭 🗗 🖟
                                                                                                                                                                                                                 □ □ Junit ×
                                                                                                                                                                                                                                                                              ⚠ OrdineDAOTest.java ×
        package test;
                                                                                                                                                                                                                         Finished after 1,939 seconds
       3⊕ import static org.junit.Assert.*;
     13 public class OrdineDAOTest {
                                                                                                                                                                                                                          OrdineDAO ordineDAO = new OrdineDAO();
    15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 40 42 43 44 45 46 47 }

    ■ doRetrieveOrdineByNumeroTest (0,004 s)
    ■ doRetrieveAllOrdiniTest (0,008 s)
               public void doRetrieveAllOrdiniTest() {
    assertTrue(ordineDAO.doRetrieveAllOrdini(0, 1000).size()==3);
    assertTrue(ordineDAO.doRetrieveAllOrdini(0, 1000).get(0).getStato().equalsIgnoreCase("in consegna"));
}
              @Test
public void doRetrieveOrdineByNlumeroTest() {
    assertTrue(ordineDAO.doRetrieveOrdineByNlumero(3).getCivico()==45);
    assertTrue(ordineDAO.doRetrieveOrdineByNlumero(100)==null);
}
               @lest
public void doRetrieveOrdiniByUtenteTest() {
    Utente utente = mock(Utente.class);
    when(utente.getId()).thenReturn(2);
    assertTrue(ordineDAO.doRetrieveOrdiniByUtente(utente).size()==2);
    assertFalse(ordineDAO.doRetrieveOrdiniByUtente(utente).get(1).getStato().equalsIgnoreCase("in consegna"));
}
                                                                                                                                                                                                                                                                                                                       B 7 6
                                                                                                                                                                                                                         Failure Trace
                public void doUpdateStatoOrdineTest() {
                     11c Vala GoUpdateStatoUrdInelest() {
Ordine o = mock(Ordine.class);
when(o.getStato()).thenReturn("Consegnato");
when(o.getNumero()).thenReturn(3);
ordineDAO.doUpdateStatoOrdine(o);
assertTrue(ordineDAO.doRetrieveOrdineByNumero(3).getStato().equals("Consegnato"));
```



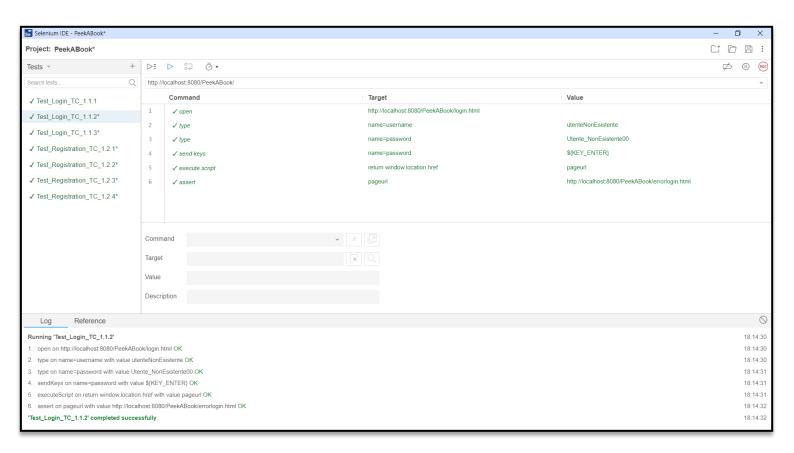
5. Test Login

Test Case ID	Test_Login_TC_1.1.1
Descrizione dell'esecuzione	L'utente inserisci solo l'username.
Risultati della procedura	Il login non viene effettuato
Anomalie	Nessuna
Risultato	PASS



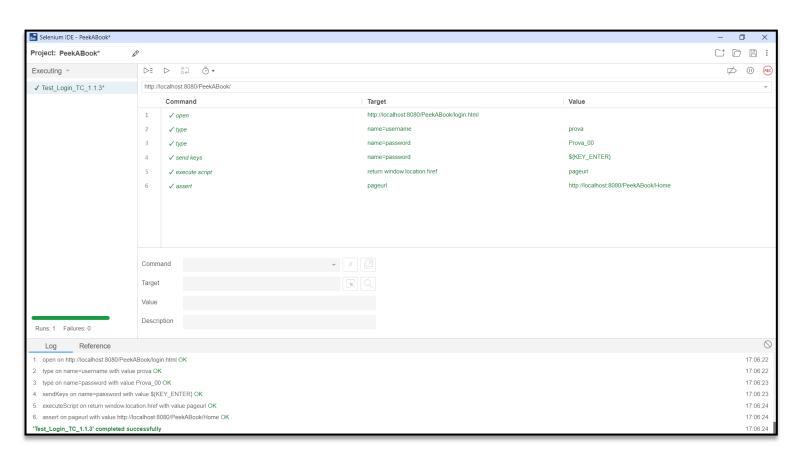


Test Case ID	Test_Login_TC_1.1.2
Descrizione dell'esecuzione	L'utente inserisce username e password che non
	hanno una corrispondenza nel database.
Risultati della procedura	Il login non viene effettuato
Anomalie	Nessuna
Risultato	PASS





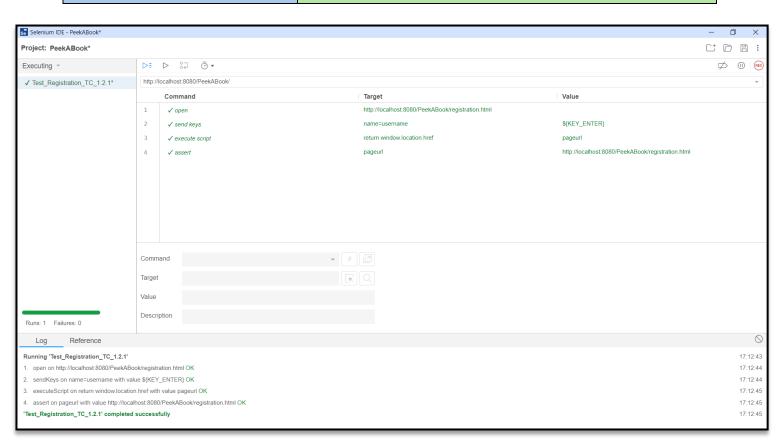
Test Case ID	Test_Login_TC_1.1.3
Descrizione dell'esecuzione	L'utente inserisce username e password che hanno
	una corrispondenza nel database.
Risultati della procedura	Il login viene effettuato
Anomalie	Nessuna
Risultato	PASS





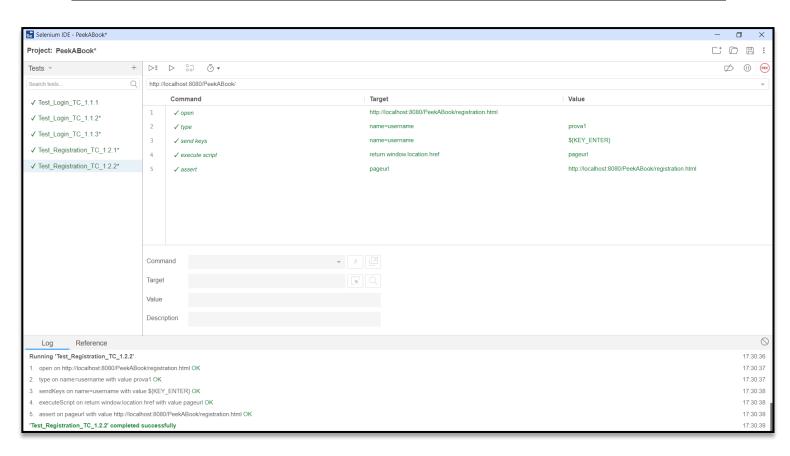
6. Test Registrazione

Test Case ID	Test_Registration_TC_1.2.1
Descrizione dell'esecuzione	L'utente non inserisce alcun dato.
Risultati della procedura	La registrazione non viene effettuata.
Anomalie	Nessuna
Risultato	PASS



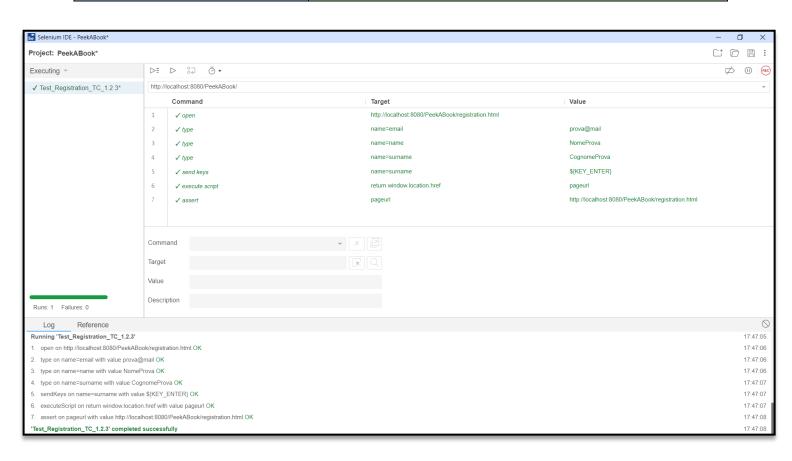


Test Case ID	Test_Registration_TC_1.2.2
Descrizione dell'esecuzione	L'utente inserisce solo l'username.
Risultati della procedura	La registrazione non viene effettuata.
Anomalie	Nessuna
Risultato	PASS



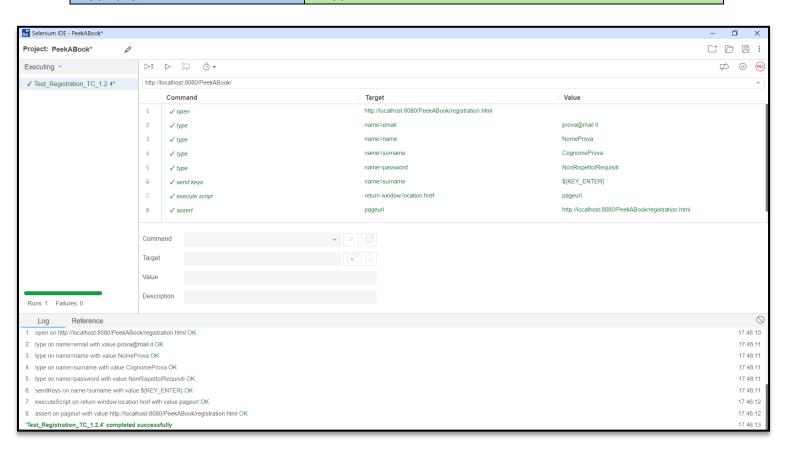


Test Case ID	Test_Registration_TC_1.2.3
Descrizione dell'esecuzione	L'utente inserisce solo email (con formato errato),
	nome e cognome.
Risultati della procedura	La registrazione non viene effettuata.
Anomalie	Nessuna
Risultato	PASS



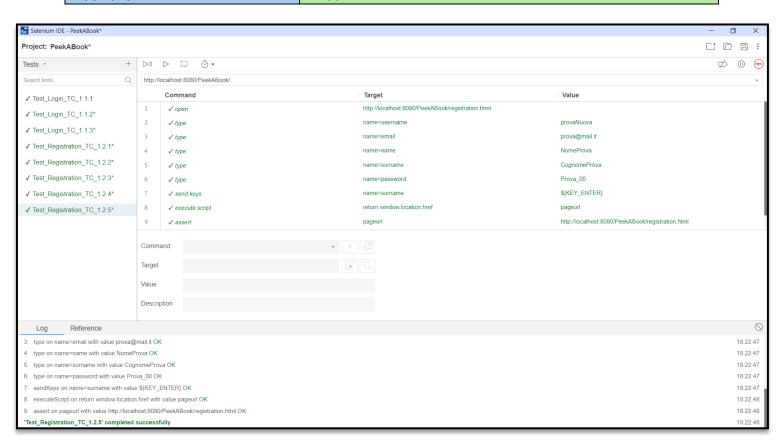


Test Case ID	Test_Registration_TC_1.2.4
Descrizione dell'esecuzione	L'utente inserisce solo email, nome, cognome e
	password (che non rispetta i requisiti minimi).
Risultati della procedura	La registrazione non viene effettuata.
Anomalie	Nessuna
Risultato	PASS



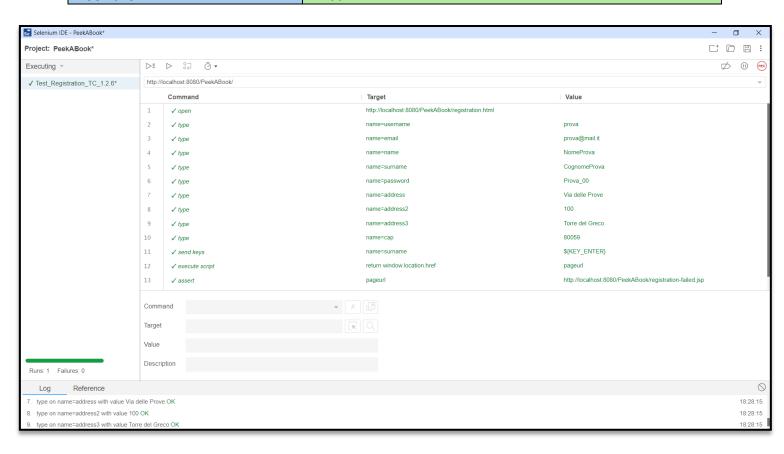


Test Case ID	Test_Registration_TC_1.2.5
Descrizione dell'esecuzione	L'utente inserisce solo i dati personali ma non quelli
	relativi all'indirizzo di residenza.
Risultati della procedura	La registrazione non viene effettuata.
Anomalie	Nessuna
Risultato	PASS



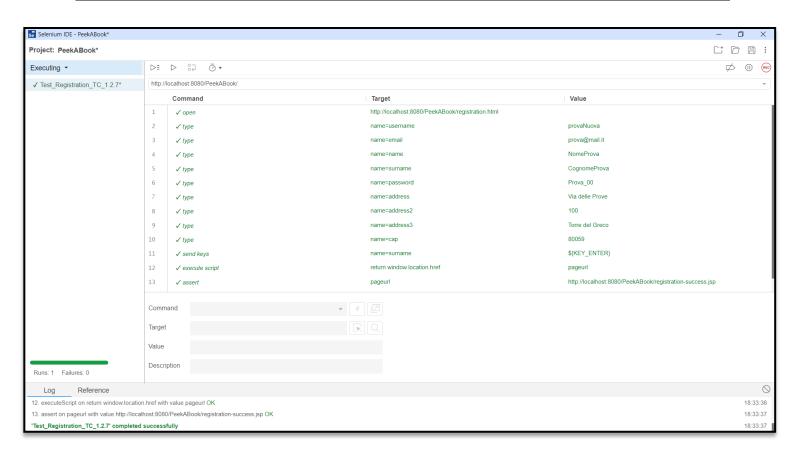


Test Case ID	Test_Registration_TC_1.2.6
Descrizione dell'esecuzione	L'utente inserisce tutti i dati correttamente ma
	l'username scelto è già in uso da un altro utente.
Risultati della procedura	La registrazione non viene effettuata.
Anomalie	Nessuna
Risultato	PASS





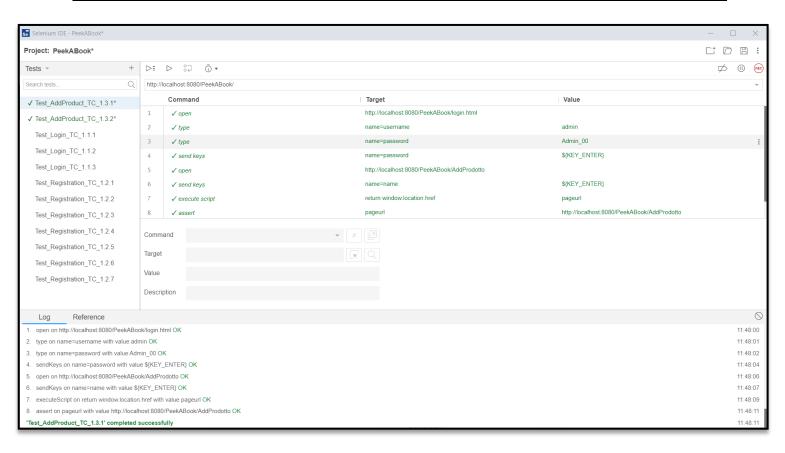
Test Case ID	Test_Registration_TC_1.2.7
Descrizione dell'esecuzione	L'utente inserisce tutti i dati correttamente.
Risultati della procedura	La registrazione viene effettuata.
Anomalie	Nessuna
Risultato	PASS





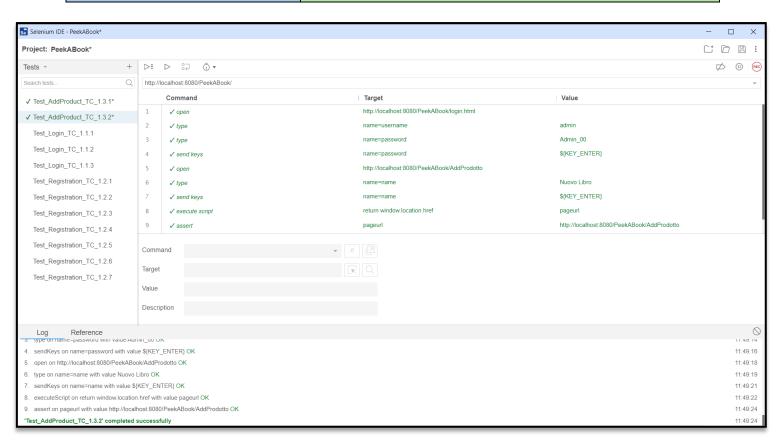
7. Test Aggiunta prodotti al catalogo

Test Case ID	Test_AddProduct_TC_1.3.1
Descrizione dell'esecuzione	L'admin non inserisce alcun dato.
Risultati della procedura	Il prodotto non viene aggiunto al catalogo.
Anomalie	Nessuna
Risultato	PASS



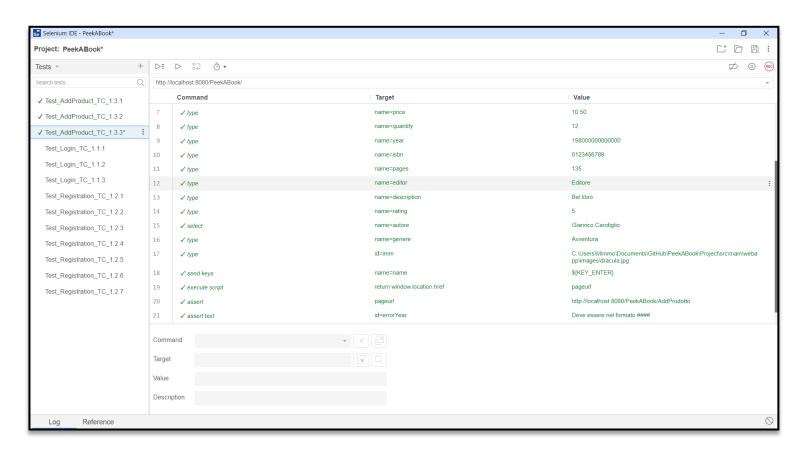


Test Case ID	Test_AddProduct_TC_1.3.2
Descrizione dell'esecuzione	L'admin inserisce solo il nome del libro.
Risultati della procedura	Il prodotto non viene aggiunto al catalogo.
Anomalie	Nessuna
Risultato	PASS



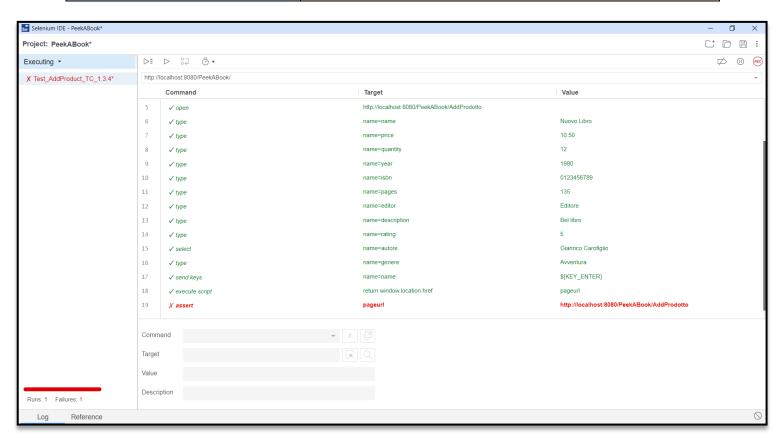


Test Case ID	Test_AddProduct_TC_1.3.3
Descrizione dell'esecuzione	L'admin inserisce tutti i dati correttamente tranne
	l'anno di pubblicazione che non rispetta il formato.
Risultati della procedura	Il prodotto non viene aggiunto al catalogo.
Anomalie	Nessuna
Risultato	PASS



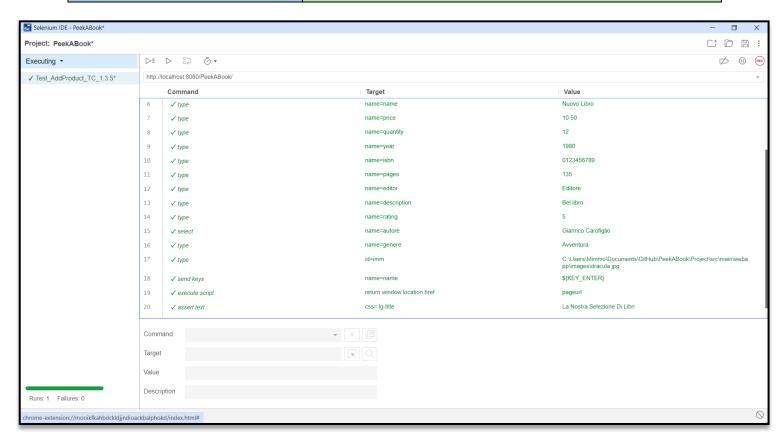


Test Case ID	Test_AddProduct_TC_1.3.4
Descrizione dell'esecuzione	L'admin inserisce tutti i dati correttamente ma non
	carica il file della copertina del libro.
Risultati della procedura	Il prodotto viene aggiunto al catalogo.
Anomalie	Nonostante non venga inserita la copertina il
	prodotto viene aggiunto al catalogo.
Risultato	FAIL





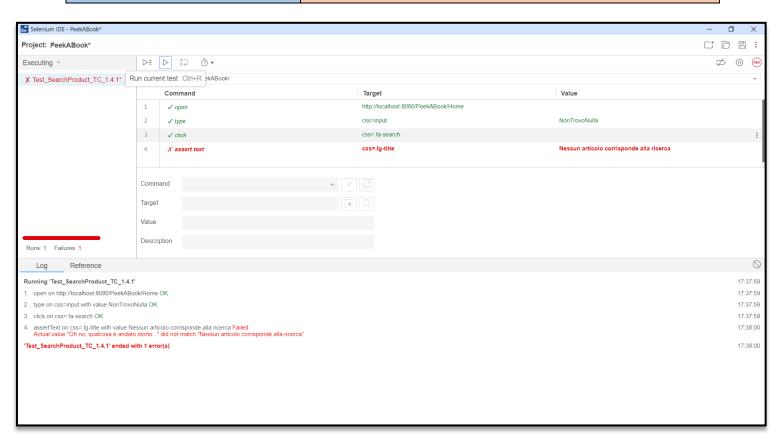
Test Case ID	Test_AddProduct_TC_1.3.5
Descrizione dell'esecuzione	L'admin inserisce tutti i dati correttamente.
Risultati della procedura	Il prodotto viene aggiunto al catalogo.
Anomalie	Nessuna
Risultato	PASS





8. Test Ricerca titoli nel catalogo

Test Case ID	Test_SearchProduct_TC_1.4.1
Descrizione dell'esecuzione	L'utente inserisce il titolo di un libro non presente nel
	catalogo.
Risultati della procedura	Viene visualizzata una pagina di errore generico.
Anomalie	Dovrebbe essere mostrata una pagina specifica che
	indica che il titolo ricercato non ha prodotto risultati.
Risultato	FAIL





Test Case ID	Test_SearchProduct_TC_1.4.2
Descrizione dell'esecuzione	L'utente inserisce il titolo di un libro presente nel
	catalogo.
Risultati della procedura	Viene visualizzata una pagina contenente il titolo
	ricercato.
Anomalie	Nessuna
Risultato	PASS

