

УПРАВЛЕНИЕ НА КАЧЕСТВОТО

Лабораторни упражнения

Тема: Езикови конструкции на Selenium IDE
доц. д-р Десислава Петрова-Антонова

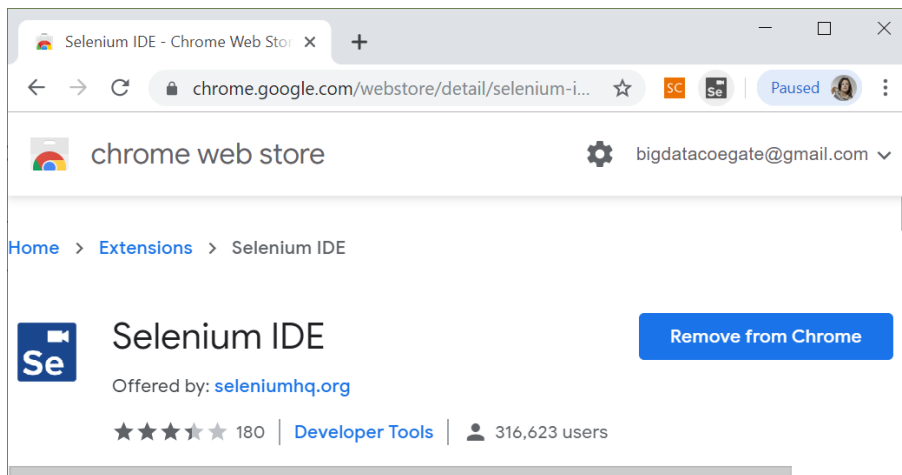
Съдържание

Създаване на проект и тестов сценарий.....	2
Дефиниране на променливи и команда If-Else.....	5
Локализиране на елементи	7
Команди Verify Element Present, Assert Element Present и Echo	12
Команда Do-Repeat If.....	15
Команди Wait	18

Ноември, 2019 г.

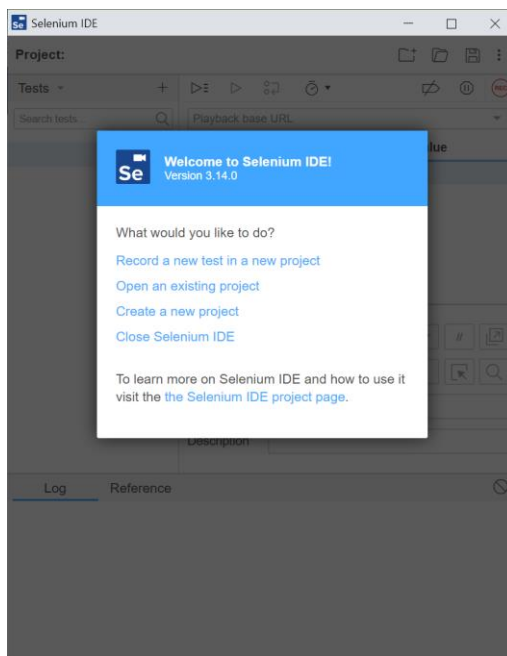
Създаване на проект и тестов сценарий

За да стартирате Selenium IDE, натиснете бутона  от лентата на браузъра, както е показано на Фиг. 1.



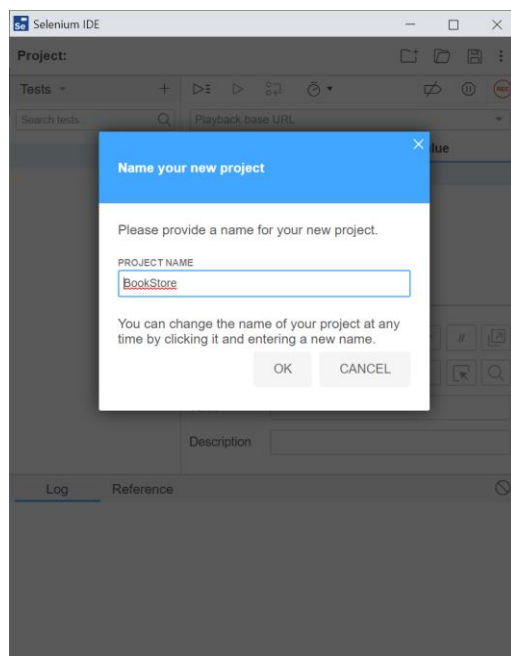
Фигура 1 Достъп до Selenium IDE от браузъра

На екрана ще се появи прозорец, показан на Фиг. 2.



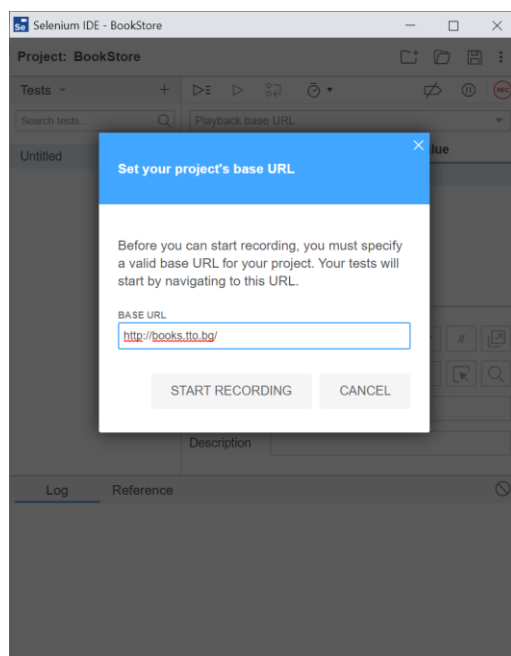
Фигура 2 Стартиране на Selenium IDE

Изберете опцията **Record a new test in a new project**, за да създадете нов проект. На екрана ще се появи прозорец, показан на Фиг. 3.



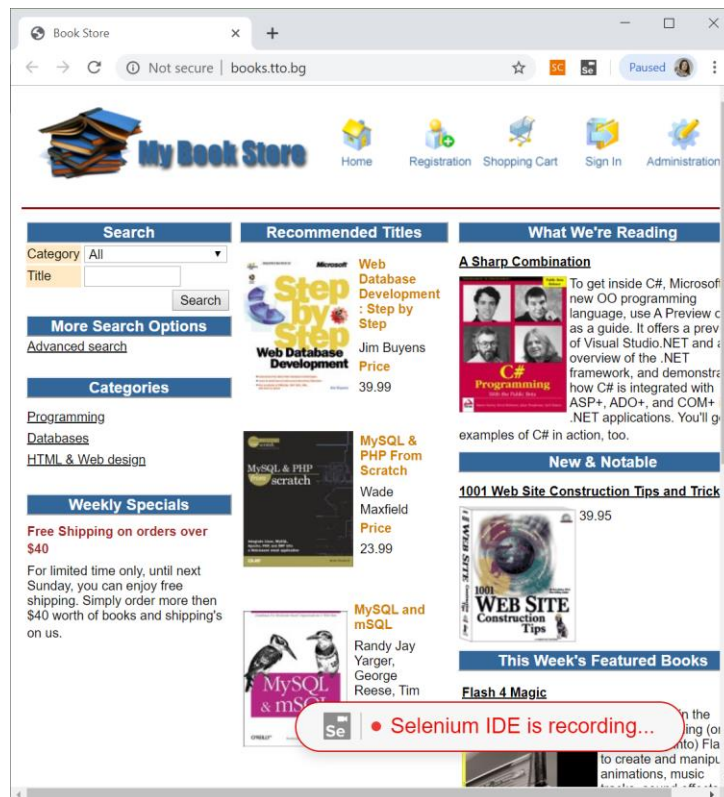
Фигура 3 Създаване на нов проект

Задайте име на проекта и натиснете бутона **OK**. На екрана ще се появи прозорец, показан на Фиг. 4, в който е необходимо да зададете основен URL адрес на приложението, което ще тествате.



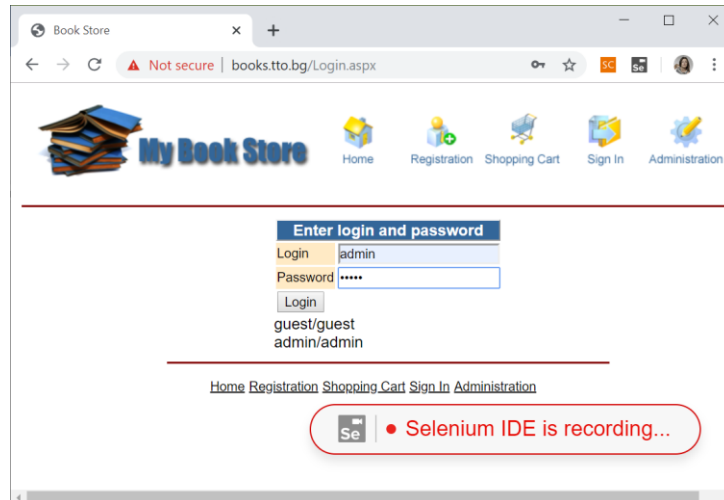
Фигура 4 Задаване на основен URL адрес

Въведете <http://books.tto.bg/> и натиснете бутона **Start recording**. В прозореца на браузъра ще се зареди приложението BookStore, както е показано на Фиг. 5.



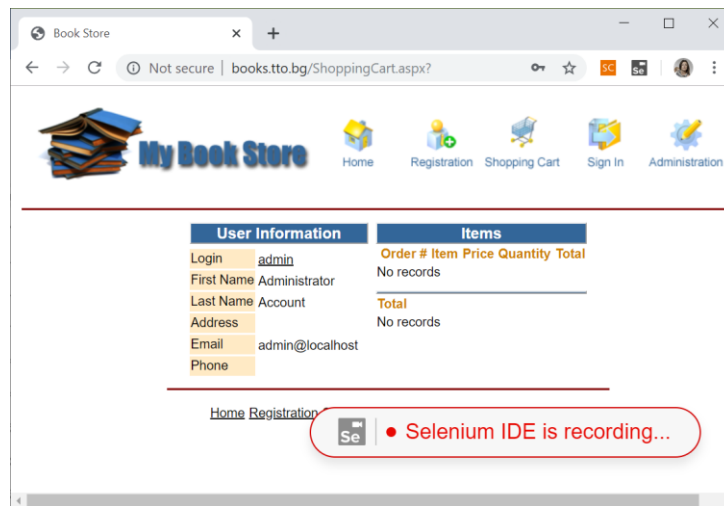
Фигура 5 Начална страница на приложението BookStore

Натиснете бутона **Sign in**. В прозореца на браузъра ще се отвори нова страница, показана на Фиг. 6.



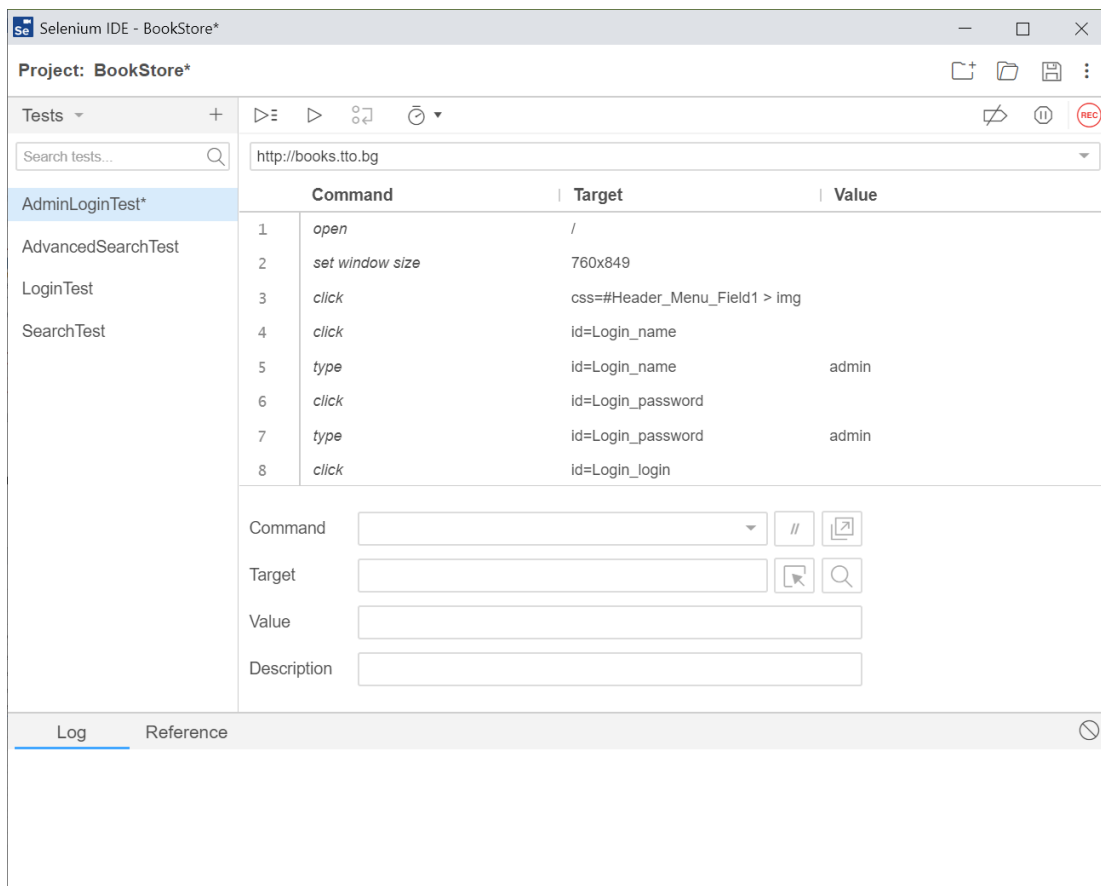
Фигура 6 Форма за вход в приложението BookStore

Въведете стойност „admin“ в полетата **Login** и **Password**. Натиснете бутона **Login**. В прозореца на браузъра ще се отвори нова страница, показана на Фиг. 7.



Фигура 7 Вход с потребител администратор

Върнете се в средата на Selenium IDE и спрете записа с бутона **Stop recording** (Ctrl+U). Задайте име на тестовия сценарий, например **AdminLoginTest**, и натиснете бутона **OK**. Създаденият тестов сценарий ще изглежда, както е показано на Фиг. 8.



Фигура 8 Тестов сценарий за вход на администратор

Дефиниране на променливи и команда If-Else

Редактирайте тестовия сценарий **AdminLoginTest**, както следва:

- 1) Добавете променлива **loginpassword** със стойност **"admin"** след ред 6 като използвате командата **store**.
- 2) На елемента **Login_password** задайте стойност от променливата **loginpassword**.
- 3) Добавете команда **if** след инициализирането на елемента **Login_password**. Условният израз е от вида **\${loginpassword} === "admin"**.
- 4) Добавете JavaScript, който инициализира променлива **output** със стойност **„correct password“**.
- 5) Добавете команда **else**.
- 6) Добавете JavaScript, който инициализира променлива **output** със стойност **„wrong password“**.
- 7) Добавете команда **end**.
- 8) Като последен ред на тестовия сценарий добавете команда **assert**, която проверява дали стойността на променливата **output** е **„correct password“**.

Кодът на тестовия сценарий след промяната е показан на Фиг. 9.

The screenshot shows the Selenium IDE interface for a project named 'BookStore*'. The URL is 'http://books.tto.bg'. The test scenario 'AdminLoginTest*' is selected, showing 15 steps in a table:

	Command	Target	Value
1	open	/	
2	set window size	838x826	
3	click	css=#Header_Menu_Field1 > img	
4	click	id=Login_name	
5	type	id=Login_name	admin
6	click	id=Login_password	
7	store	admin	loginpassword
8	type	id=Login_password	\${loginpassword}
9	if	\${loginpassword} === "admin"	
10	execute script	return "correct password"	output
11	else		
12	execute script	return "wrong password"	output
13	end		
14	click	id=Login_login	
15	assert	output	correct password

Below the table, there are input fields for Command (open), Target (/), Value, and Description.

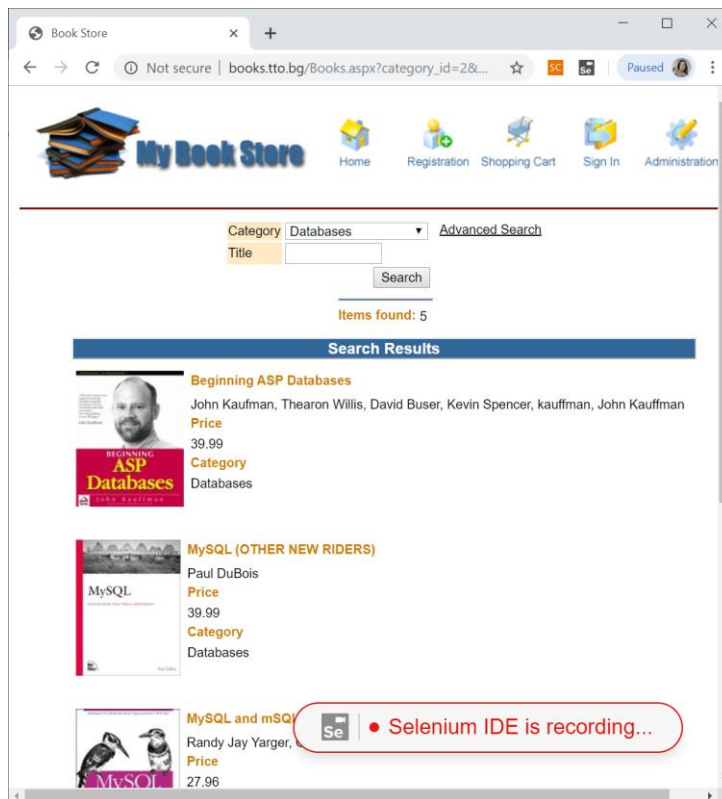
Фигура 9 Тестов сценарий с команда IF-ELSE

Изпълнете тестовия сценарий.

Променете тестовия сценарий, за да тествате потребител с име guest.

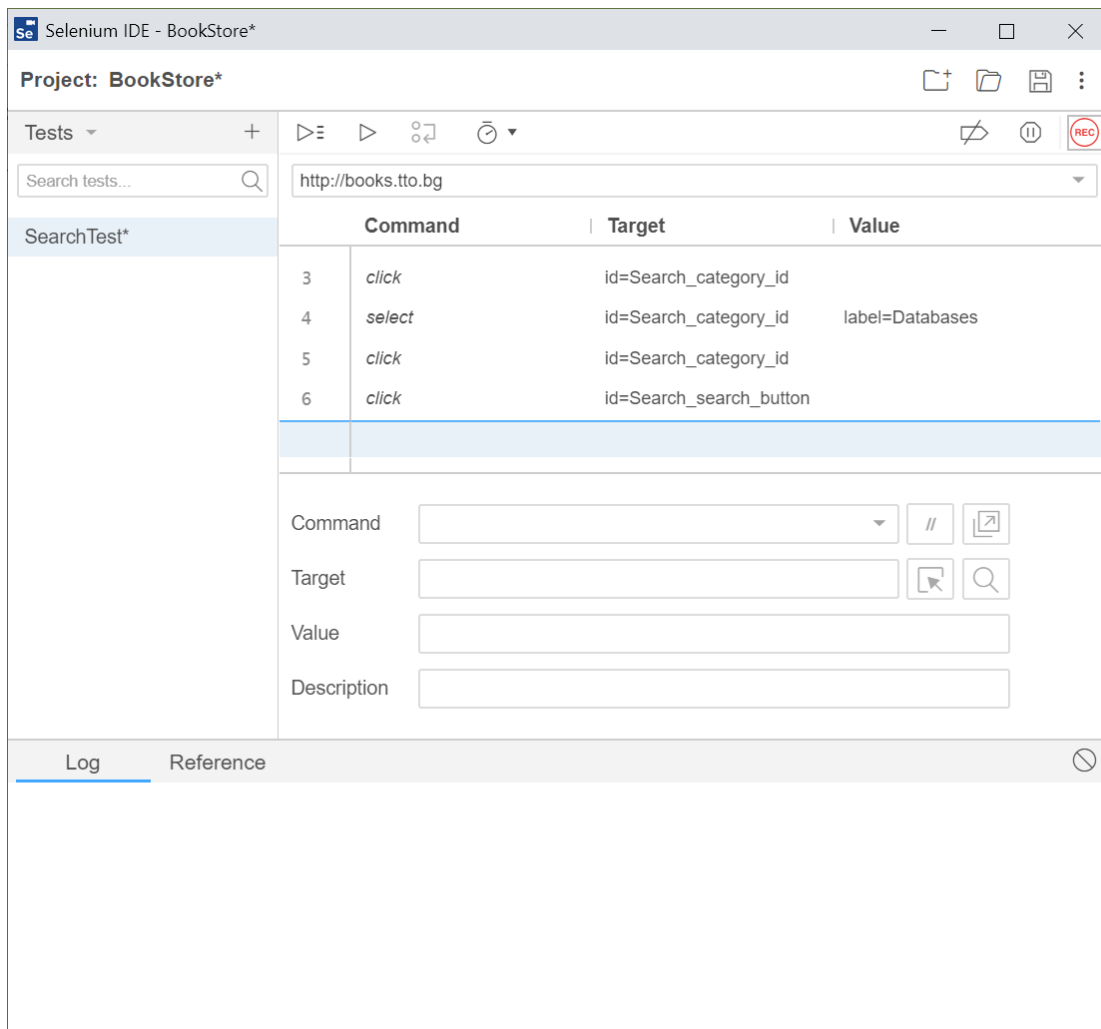
Локализиране на елементи

Създайте нов тестов сценарий с име **SearchTest**. Натиснете бутона за запис (Ctrl+U). В прозореца на брауъра ще се зареди приложението BookStore, както е показано на Фиг. 5. Изберете категория Databases в приложението BookStore и натиснете бутона Search. В прозореца на брауъра ще се отвори нова страница, показана на Фиг. 10.



Фигура 10 Резултат от търсене в приложението BookStore

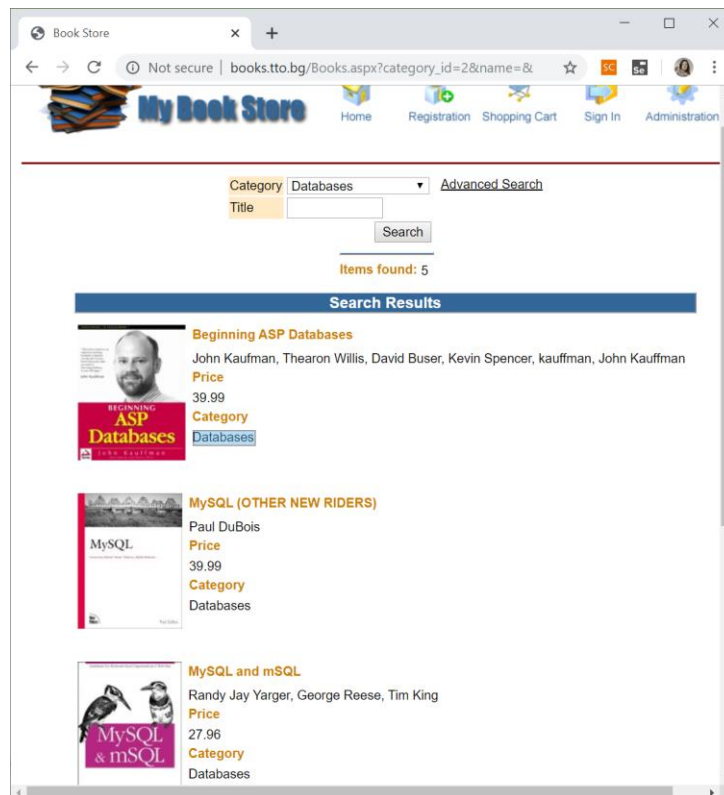
Върнете се в средата на Selenium IDE и спрете записа с бутона **Stop recording** (Ctrl+U). Създаденият тестов сценарий ще изглежда, както е показано на Фиг. 11.



Фигура 11 Тестов сценарий за търсене на книга

Редактирайте тестовия сценарий **SearchTest**, както следва:

- 1) Добавете променлива **searchcategory** със стойност **"Programming"** след ред 3 като използвате командата **store**.
- 2) Като последен ред на тестовия сценарий добавете команда **verify text**.
- 3) Натиснете бутона **Select target in page** вдясно от полето **Target**.
- 4) Върнете се в приложението **BookStore** и маркирайте текста **Databases**, както е показано на Фиг. 12. Маркираният текст се оцветява в светлосин цвят.



Фигура 12 Избор на елемент с идентификатор

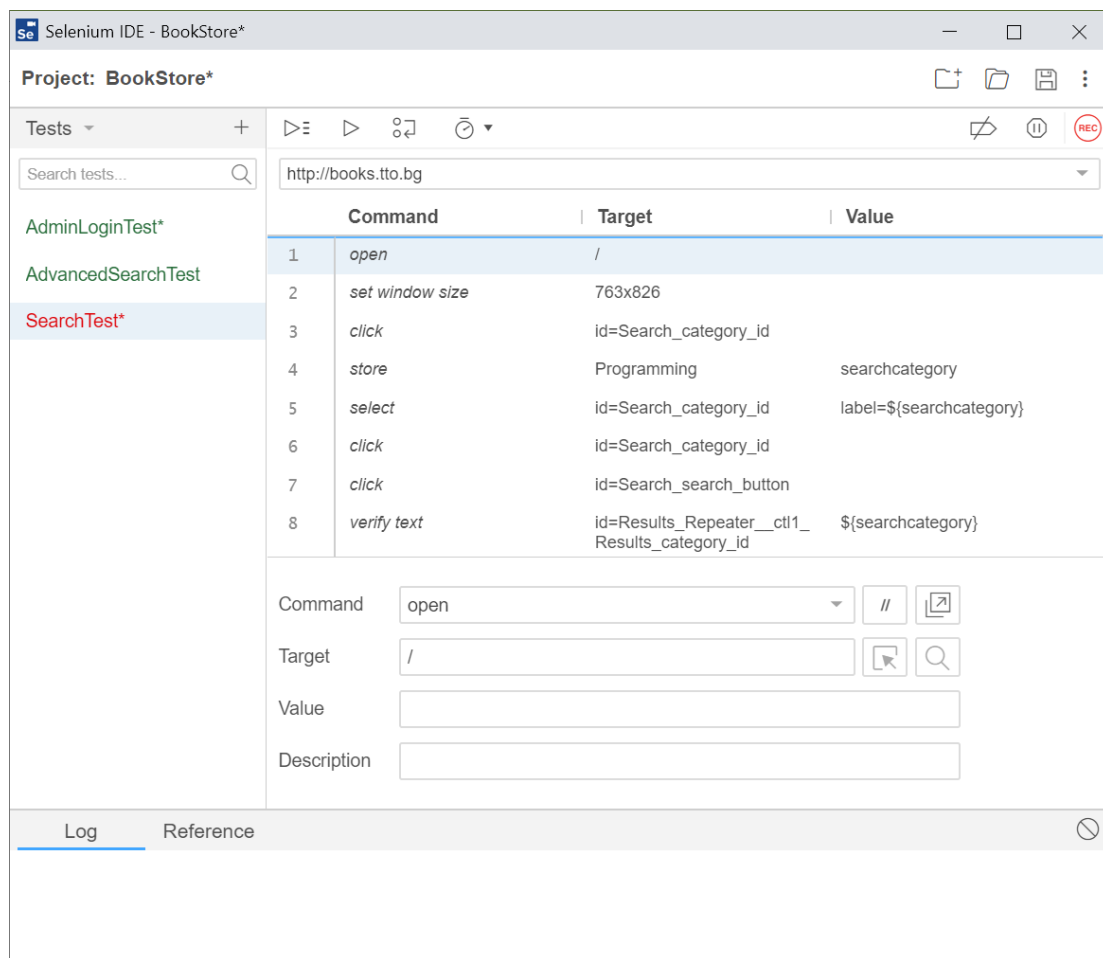
В полето **Target** ще се попълни идентификатора на избрания елемент от страницата – `id=Results_Repeater__ctl1_Results_category_id`.

5) Задайте в полето Value стойността на променливата `${searchcategory}`.

Кодът на тестовия сценарий след промяната е показан на Фиг. 13.

В така създадения тестов сценарий за локализиране на елемент в уеб страницата е използван идентификатора – атрибут ID. В допълнение Selenium IDE поддържа следните възможности за локализиране на елемент:

- Name – атрибутът Name е подобен на атрибута ID и служи за локализиране на елемент по име;
- Link – използва се за локализиране на хипервръзки;
- CSS – използва се за локализиране на елементи, които нямат име или идентификатор. Локализирането на елемент е базирано на шаблон, при което разпознаването се извършва посредством комбинация от HTML tag (tag), идентификатор (Id), клас (class) и атрибути (attributes).
- XPath – използва се XPath израз за локализиране на елемент, чрез който се извършва търсене в DOM (Document Object Model) дървото.
- DOM – Йерархично локализиране на елементи в DOM дървото.



Фигура 13 Тестов сценарий с команда verify text

Изпълнете тестовия сценарий. Резултатът е както следва:

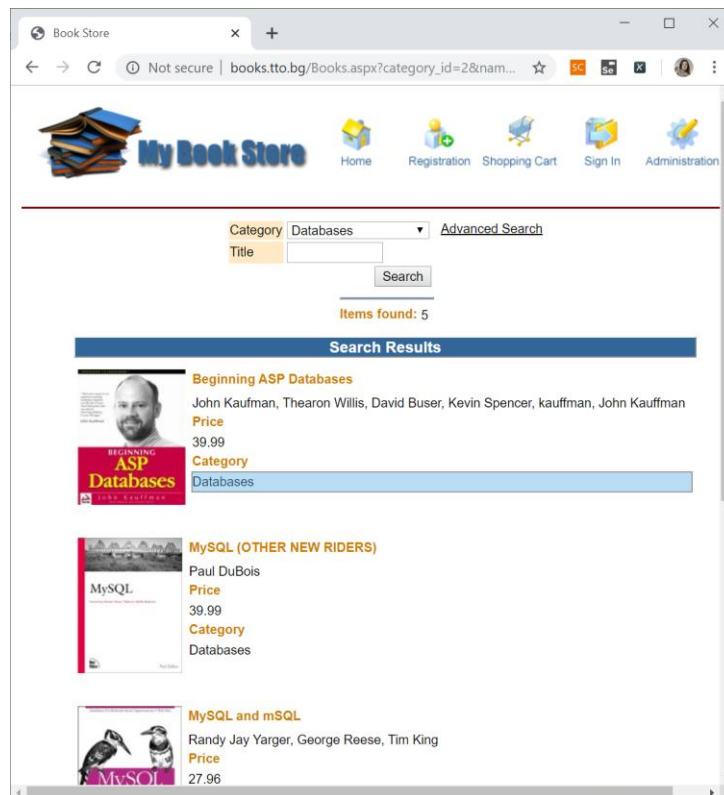
verifyText on css=tr:nth-child(3) tr:nth-child(7) > td:nth-child(1) with value \${searchcategory} **Failed:** 18:05:18
Actual value "Databases" did not match "Programming"

Таблица 1 обобщава методите за локализиране на елемент.

Таблица 1 Локализиране на елементи

Метод	Target синтаксис
ID	id = id_of_the_element
Name	name = name_of_the_element
Name с филтър	name = name_of_the_element filter = value_of_filter
Link	link = link_text
Tag и ID	css=tag#id
Tag и клас	css=tag.class
Tag и атрибут	css=tag[attribute=value]
Tag, клас и атрибут	css=tag.class[attribute=value]

Променете селектора с бутона **Select target in page** така, че да получите локализиран елемент със съдържание „Databases“ в уеб страницата с каскаден стил (CSS) от вида `css=tr:nth-child(3) tr:nth-child(7) > td:nth-child(1)`. Резултатът от локализирането на елемента е показан на Фиг. 14.



Фигура 14 Локализиране на елемент с каскаден стил


Изпълнете тестовия сценарий. Резултатът е както следва:

```
verifyText on css=tr:nth-child(3) tr:nth-child(7) > td:nth-child(1) with value ${searchcategory} Failed: 17:29:16
Actual value "Databases" did not match "Programming"
```

Върнете се в средата на Selenium IDE, изберете последния ред от теста и отворете падащото меню на полето Target. Изберете XPath селектор за локализиране на елемента и отново изпълнете тестовия сценарий. Резултатът е както следва:

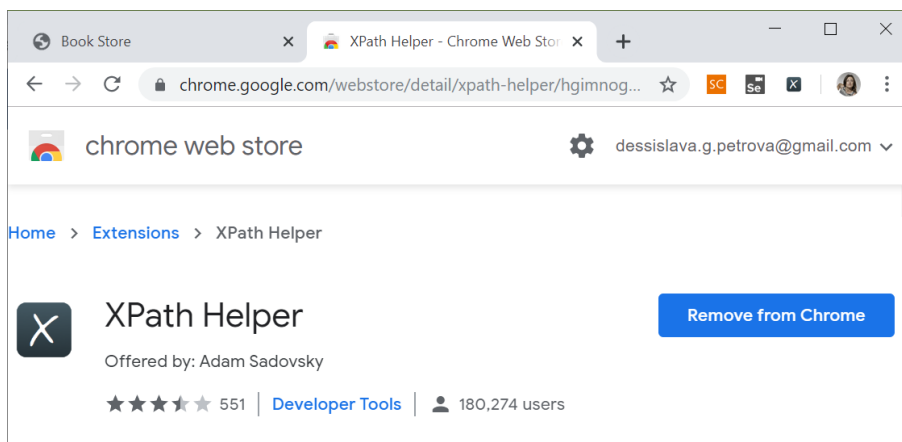
```
verifyText on xpath=//table[@id='Results_holder']/tbody/tr[3]/td/table/tbody/tr[2]/td[2]/table/tbody/tr[7]/td with value
${searchcategory} Failed: 17:31:39
Actual value "Databases" did not match "Programming"
```

Chrome и Firefox поддържат разширения, за откриване на XPath изрази в уеб страница. Примерно разширение е XPath Helper¹. Подобни разширения са FireBug и FirePath².

Инсталирайте XPath Helper. За да стартирате XPath Helper, натиснете бутона  от лентата на браузъра, както е показано на Фиг. 15.

¹ <https://chrome.google.com/webstore/detail/xpath-helper/hgimnogillphhkhlmeebmlgioejdpjl?hl=en>

² <https://www.toolsqa.com/selenium-webdriver/xpath-firebug-firepath/>

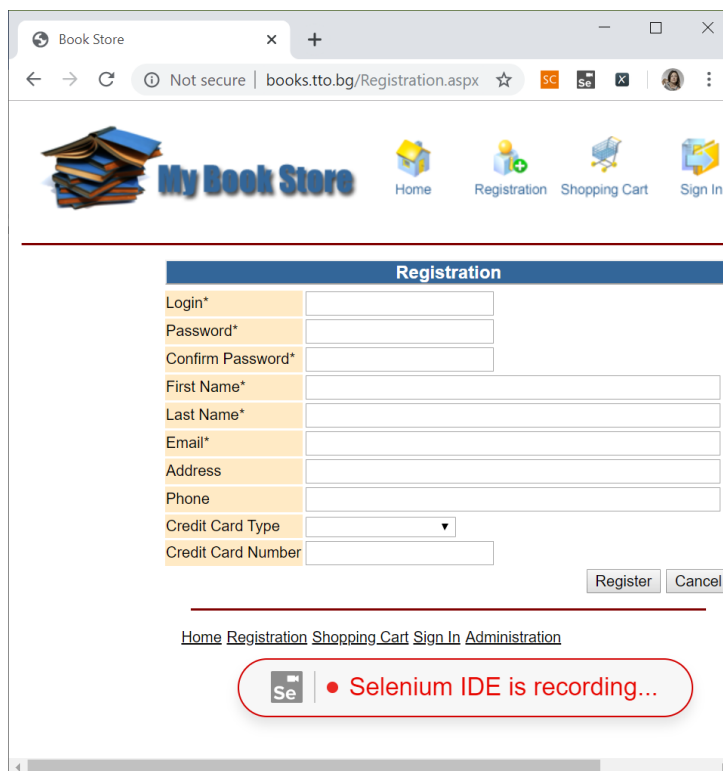


Фигура 15 XPath Helper

Променете тестовия сценарий, за да тествате категории Databases и HTML & Web Design.

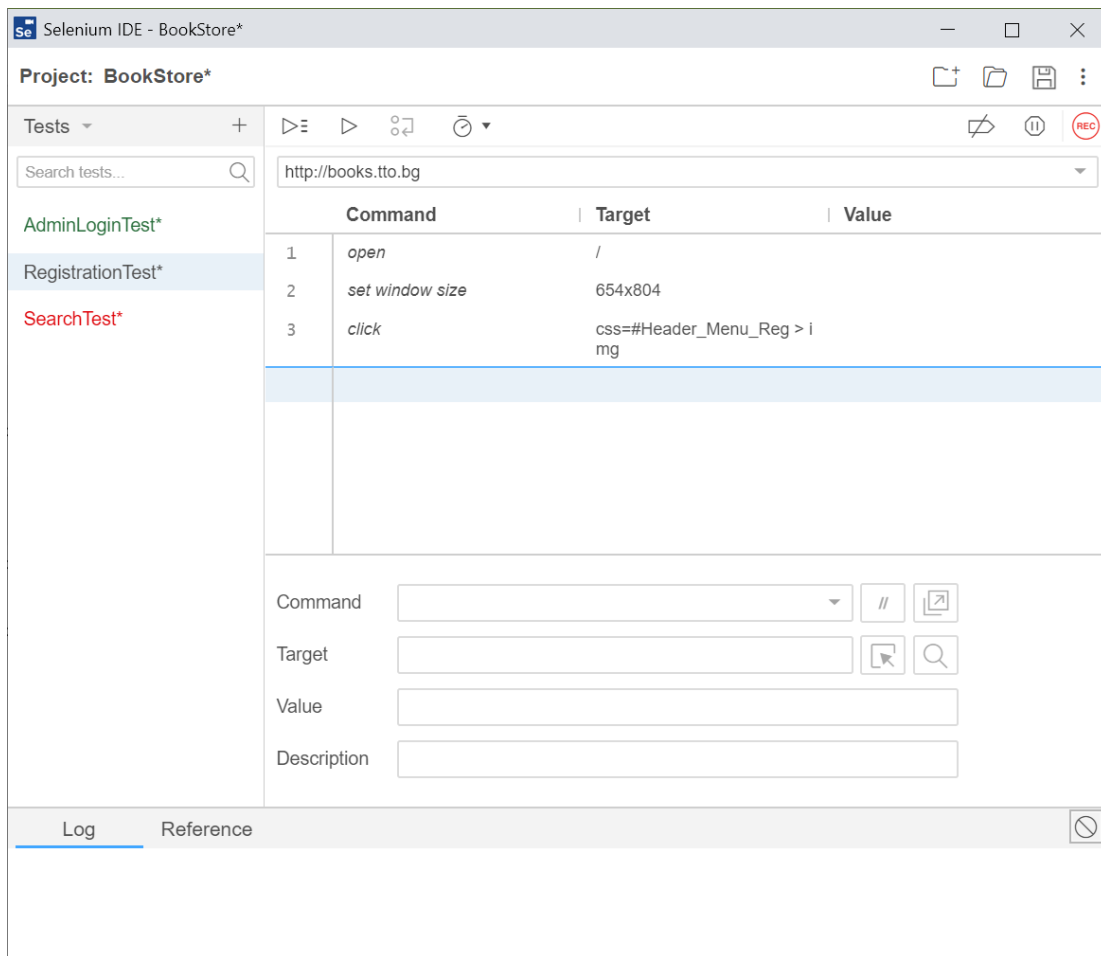
Команди Verify Element Present, Assert Element Present и Echo

Създайте нов тестов сценарий с име **RegistrationTest**. Натиснете бутона за запис (Ctrl+U). В прозореца на брауъра ще се зареди приложението BookStore, както е показано на Фиг. 5. Натиснете бутона **Registration**. В прозореца на брауъра ще се отвори нова страница, показана на Фиг. 10.



Фигура 16 Форма за регистрация в приложението BookStore

Върнете се в средата на Selenium IDE и спрете записа с бутона **Stop recording** (Ctrl+U). Създаденият тестов сценарий ще изглежда, както е показано на Фиг. 17.

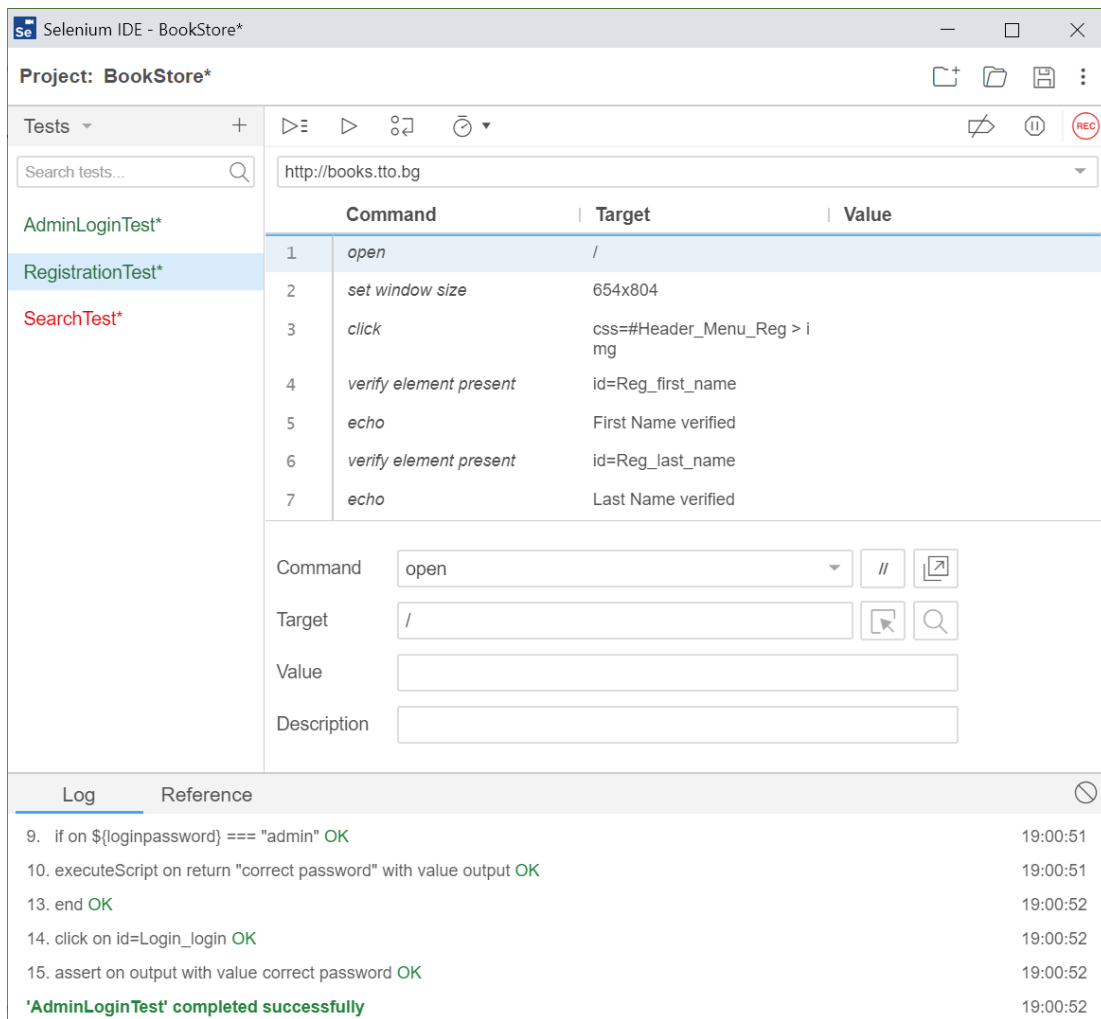


Фигура 17 Тестов сценарий за регистрация на потребител

Редактирайте тестовия сценарий **RegistrationTest**, както следва:

- 1) Добавете нов ред в края на тестовия сценарий.
- 2) Изберете команда **verify element present** от полето **Comand**.
- 3) Натиснете бутона **Select target in page** вдясно от полето **Target**.
- 4) Върнете се в приложението **BookStore** и маркирайте текстовото поле с етикет **First Name** в уеб страницата.
- 5) Добавете команда ехо в средата на Selenium IDE със стойност „**First Name verified**“ за полето **Target**.
- 6) Добавете нов ред в края на тестовия сценарий в средата на Selenium IDE.
- 7) Изберете команда **verify element present** от полето **Comand**.
- 8) Натиснете бутона **Select target in page** вдясно от полето **Target**.
- 9) Върнете се в приложението **BookStore** и маркирайте текстовото поле с етикет **Last Name** в уеб страницата.
- 10) Добавете команда ехо в средата на Selenium IDE със стойност „**Last Name verified**“ за полето **Target**.

Кодът на тестовия сценарий след промяната е показан на Фиг. 18.



Фигура 18 Тестов сценарий за регистрация на потребител с команда *verify element present*

Изпълнете тестовия сценарий. Променете тестовия сценарий като зададете погрешни идентификатори за командите **verify element present**, например `id=Reg_first_name_wrong` и `id=Reg_last_name_wrong`. Изпълнете тестовия сценарий. Резултатът за командата **verify element present** е както следва:

```
verifyElementPresent on id=Reg_first_name_wrong Failed: 19:05:17
Element with locator id=Reg_first_name_wrong could not be found
```

```
verifyElementPresent on id=Reg_last_name_wrong Failed: 19:05:17
Element with locator id=Reg_last_name_wrong could not be found
```

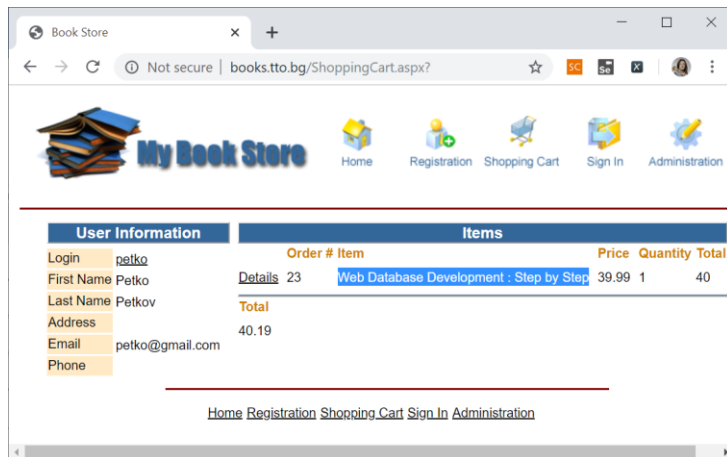
Заменете командите **verify element present** с **assert element present**. Изпълнете тестовия сценарий. Резултатът за командата **assert element present** е както следва:

```
assertElementPresent on id=Reg_first_name_wrong Failed: 19:10:26
Element with locator id=Reg_first_name_wrong could not be found
```

ВАЖНО: Разликата между командите **verify** и **assert** се състои в това, че при неуспешно изпълнение на командата **assert** тестовият сценарий спира да се изпълнява на съответната стъпка, докато при неуспешно изпълнение на командата **verify** тестовият сценарий се изпълнява до край. Поради тази причина командите **echo** не се изпълняват след замяната на командите **verify** и **assert**.

Команда Do-Repeat If

Създайте нов тестов сценарий с име **ShopingCartTest**. Натиснете бутона за запис (Ctrl+U). В прозореца на браузъра ще се зареди приложението BookStore, както е показано на Фиг. 5. Влезте в приложението като потребител с име guest. Върнете се към началната страница и добавете книга със заглавие „Web Database Development: Step by Step“ в потребителската кошница. В прозореца на браузъра ще се отвори нова страница, показана на Фиг. 19.

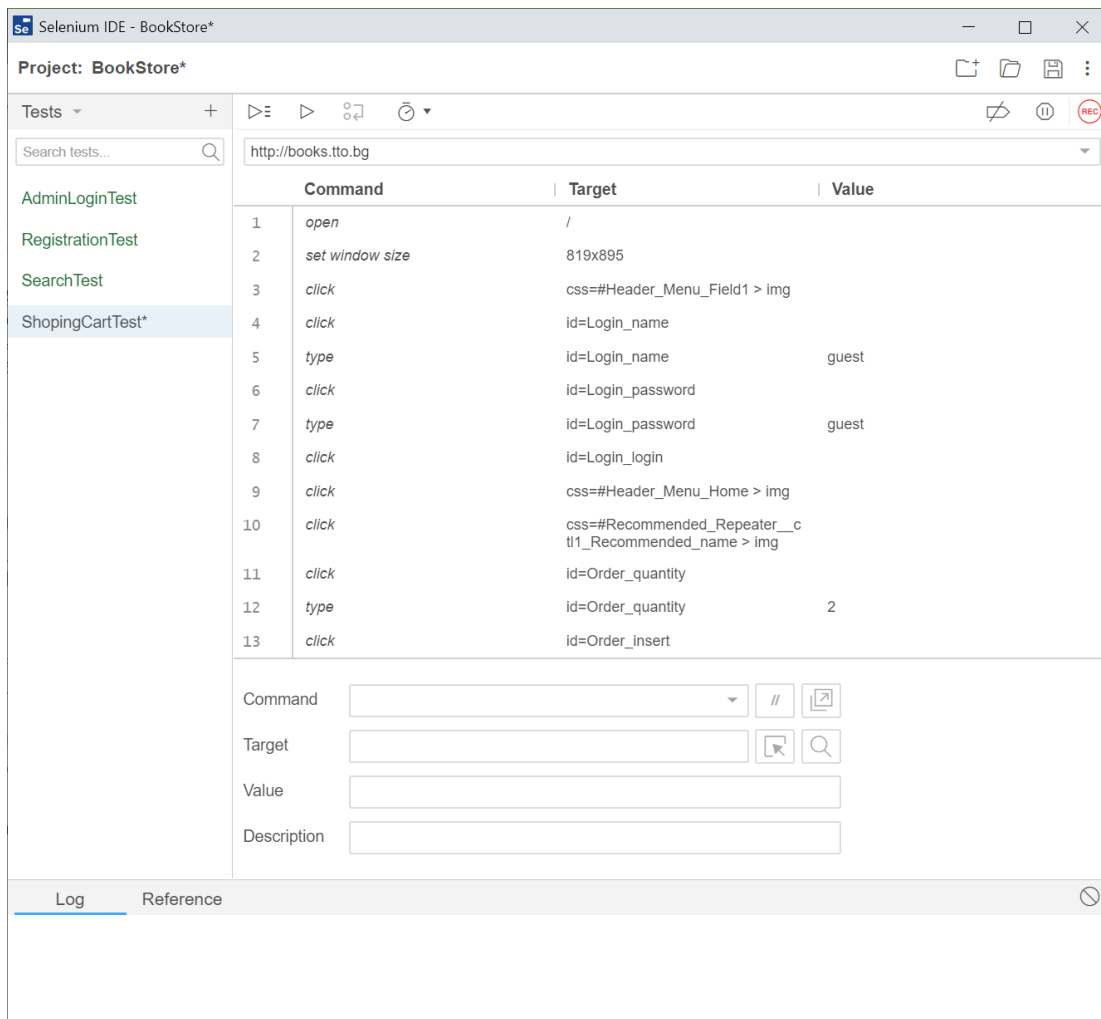


Фигура 19 Потребителска кошница с един артикул в приложението BookStore

Върнете се в средата на Selenium IDE и спрете записа с бутона **Stop recording** (Ctrl+U). Създаденият тестов сценарий ще изглежда, както е показано на Фиг. 20.

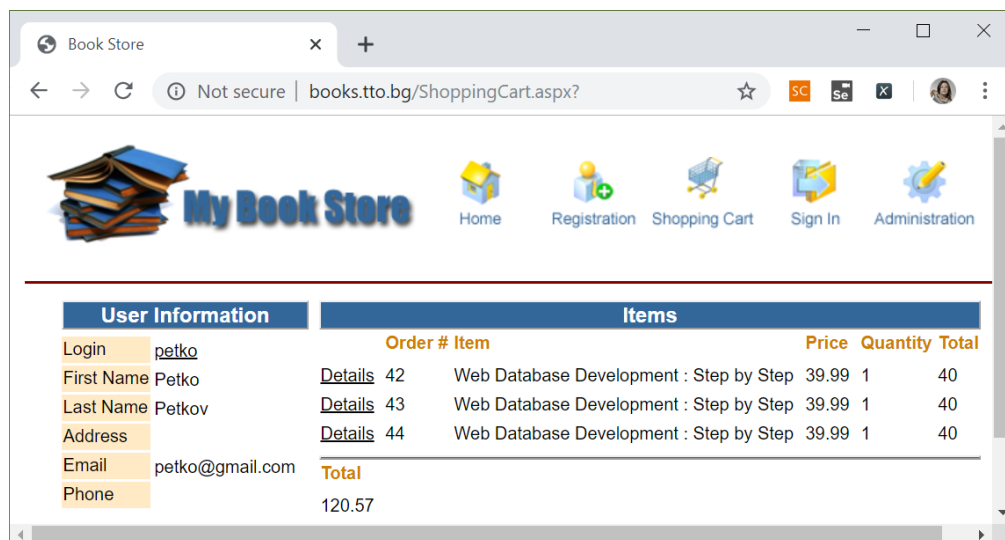
Редактирайте тестовия сценарий **ShopingCartTest**, както следва:

- 1) Добавете нов ред след ред 8, като добавите скрипт с командата **execute script** за инициализиране на променлива **loopvariable** с 1. В полето **Target** запишете return 1.
- 2) Създайте цикъл като добавите команда **do**. В тялото на цикъла се включват стъпките за връщане към начална страница, избирането на книга и добавянето ѝ в потребителската кошница с количество 2.
- 3) В края на тестовия сценарий добавете нова стъпка за извеждане на стойността на променливата **loopvariable** с помощта на командата **echo** в **Log** панела на Selenium IDE.
- 4) Добавете скрипт с командата **execute script** за инкрементиране на стойността на променлива **loopvariable** с 1. В полето **Target** запишете return \${loopvariable}+1.
- 5) Добавете команда **repeat if**, като поставите условие за край на цикъла в полето Target – \${loopvariable} < 4.



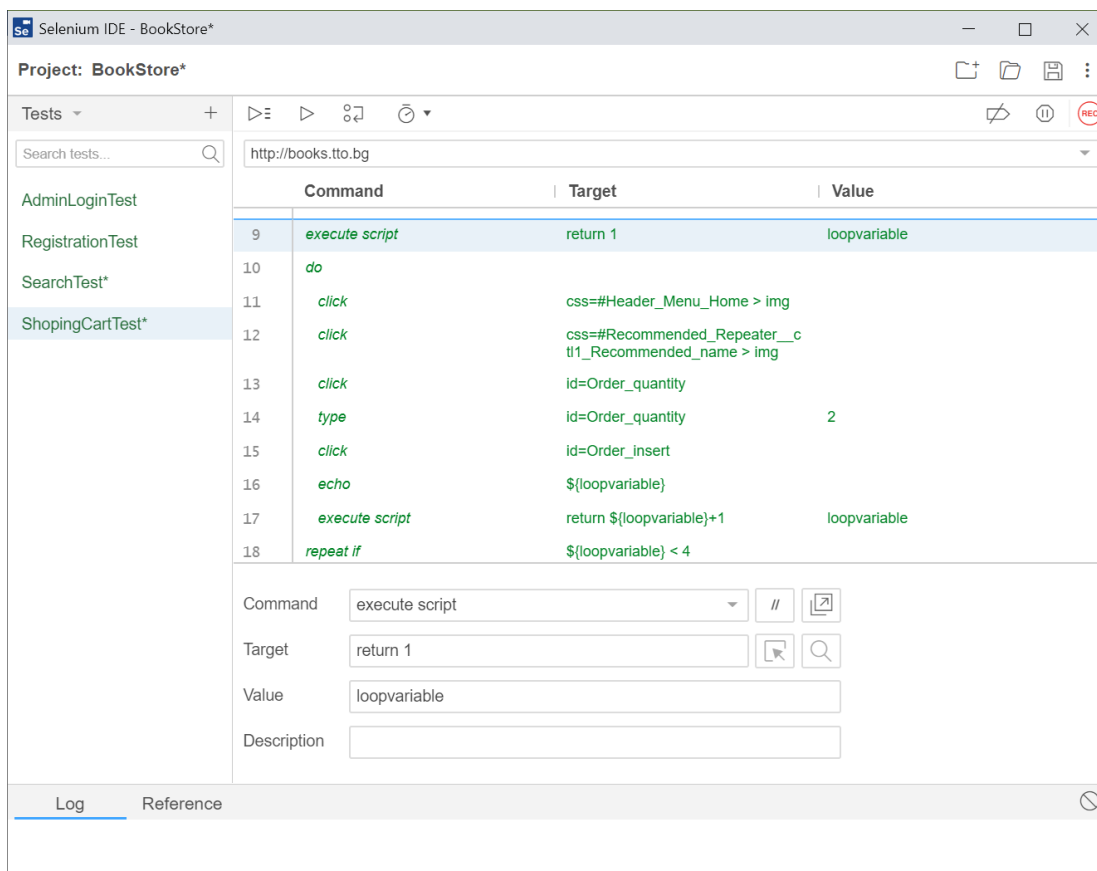
Фигура 20 Тестов сценарий за потребителска кошница

В резултат на добавения цикъл, при изпълнението на тестовия сценарий в потребителската кошница ще бъдат добавени 3 артикула, както е показано на Фиг. 21. Кодът на тестовия сценарий след промяната е показан на Фиг. 22.



Фигура 21 Потребителска кошница с три артикула в приложението BookStore

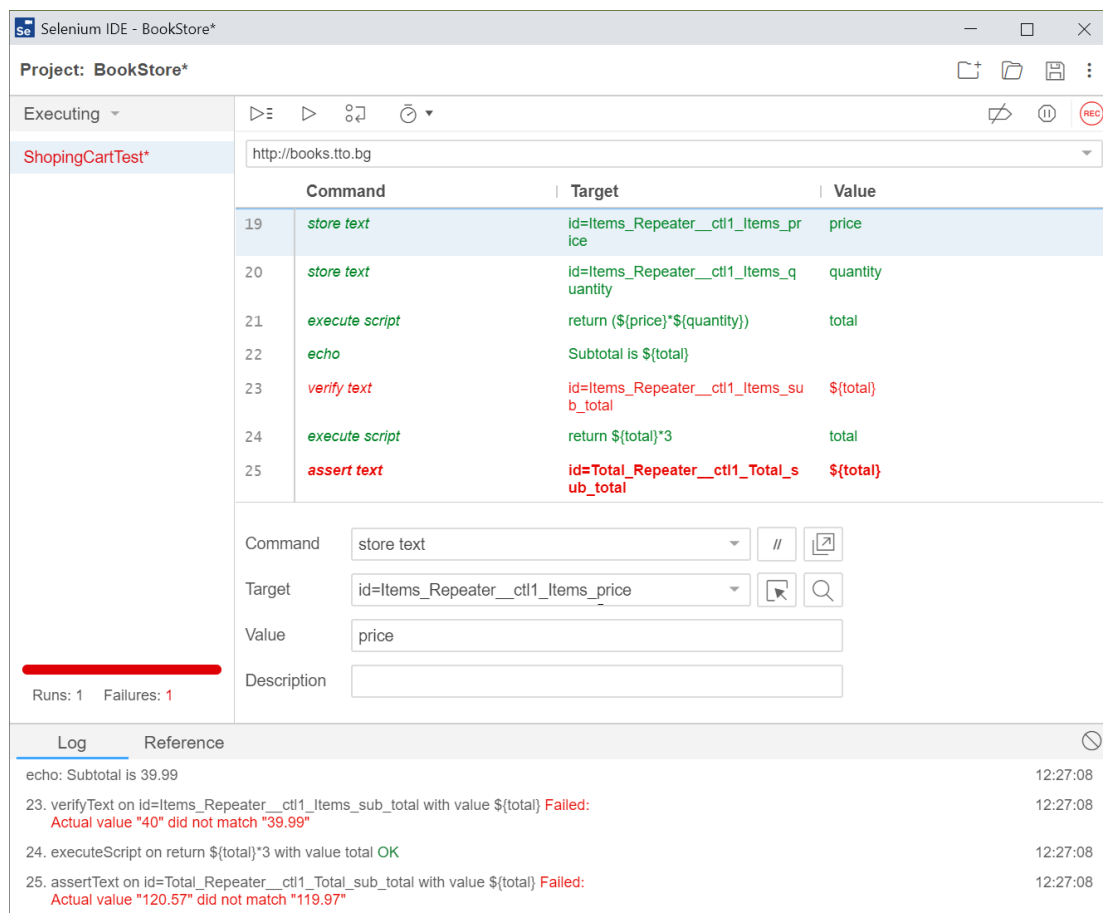
Премахнете от потребителската кошница добавената книга по време на записа на тестовия сценарий. Изпълнете тестовия сценарий.



Фигура 22 Тестов сценарий с цикъл DO-REPEAT IF

Редактирайте тестовия сценарий **ShoppingCartTest**, както следва:

- 1) Добавете нова стъпка в края на тестовия сценарий, за да съхраните единичната цена на артикула в променлива **price**, като използвате командата **store text**.
- 2) Добавете нова стъпка, за да съхраните количеството на артикула в променлива **total**, като използвате командата **store text**.
- 3) Добавете скрипт с командата **execute script**, за да пресметнете общата цена на артикула **total**. В полето **Target** запишете `return (${price}*${quantity})`.
- 4) Добавете нова стъпка за извеждане на стойността на променливата **total** с помощта на командата **echo** в **Log** панела на Selenium IDE.
- 5) Проверете дали пресметнатата обща цена на артикула от уеб приложението е равна на очакваната, която е изчислена и записана в променливата **total**. Използвайте командата **verify text**.
- 6) Добавете скрипт с командата **execute script**, за да пресметнете общата цена на поръчката **total**. В полето **Target** запишете `return ${total}*3`.
- 7) Проверете дали пресметнатата обща цена на поръчката от уеб приложението е равна на очакваната, която е изчислена и записана в променливата **total**. Използвайте командата **assert text**.



Фигура 23 Тестов сценарий за проверка на изчисленията в потребителската кошница

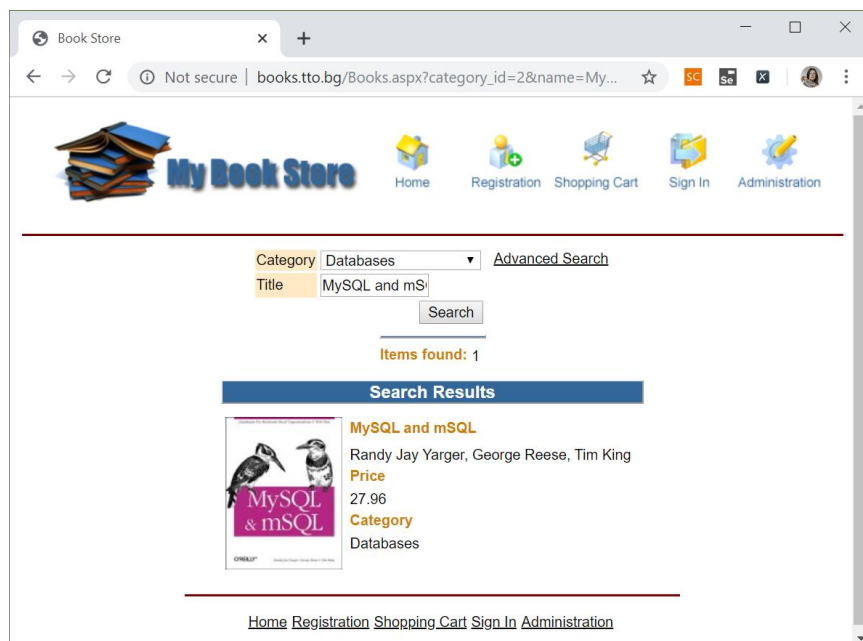
Изпълнете тестовия сценарий. Резултатът за командите **verify text** и **assert text** е както следва:

verifyText on id=Items_Repeater__ctl1_Items_sub_total with value \${total} **Failed:12:27:08**
Actual value "40" did not match "39.99"

assertText on id=Total_Repeater__ctl1_Total_sub_total with value \${total} **Failed:12:27:08**
Actual value "120.57" did not match "119.97"

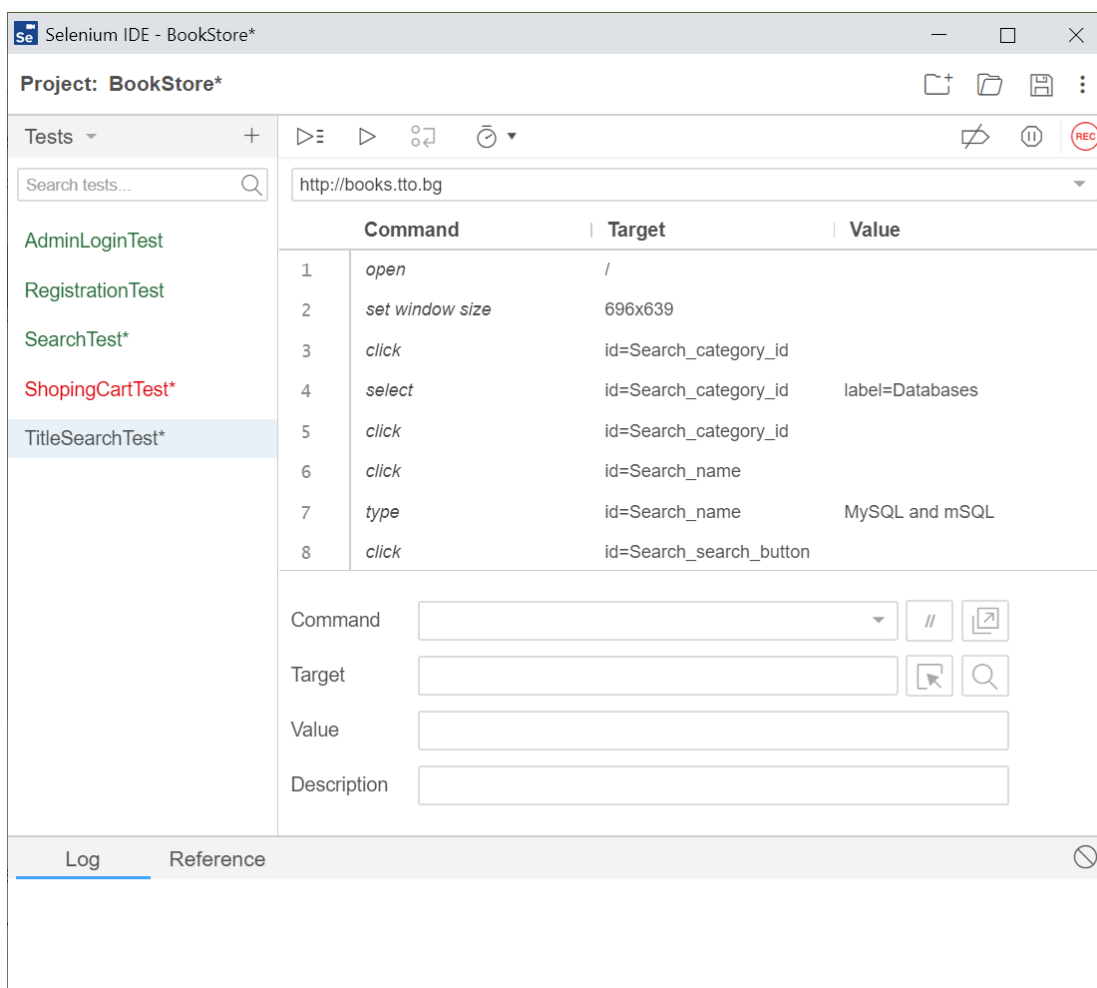
Команди Wait

Създайте нов тестов сценарий с име **TitleSearchTest**. Натиснете бутона за запис (Ctrl+U). В прозореца на браузъра ще се зареди приложението BookStore, както е показано на Фиг. 5. Изберете категория **Databases** и въведете заглавие „**MySQL and mSQL**“ в полето **Title**. В прозореца на браузъра ще се отвори нова страница, показана на Фиг. 24.



Фигура 24 Търсене на книга по категория и заглавие

Върнете се в средата на Selenium IDE и спрете записа с бутона **Stop recording** (Ctrl+U). Създаденият тестов сценарий ще изглежда, както е показано на Фиг. 25.



Фигура 25 Тестов сценарий за търсене по категория и заглавие на книга

Редактирайте тестовия сценарий **TitleSearchTest**, както следва:

- 1) Добавете нова стъпка в края на тестовия сценарий с команда **wait for element present**. Използвайте бутона **Select target in page**, за да локализиране изображението с корицата на книгата в полето **Target**. Променете стойността в полето **Value** на 1000 милисекунди.
- 2) Добавете стъпка, в която да проверите дали изображението с корицата на книгата е налично в уеб страницата. За целта използвайте командата **verify element present**.

Изпълнете тестовия сценарий.

Редактирайте тестовия сценарий **TitleSearchTest**, както следва:

- 1) Добавете нова стъпка след реда с командата **click** върху елемент с идентификатор **Search_category_id**, като дефинирате променлива **category** със стойност **Programming**.
- 2) Добавете нова стъпка след реда с командата **click** върху елемент с идентификатор **Search_name**, като дефинирате променлива **title** със стойност **A Sharp Combination**.

Изпълнете тестовия сценарий. Резултатът за командата **wait for element present** е както следва:

```
waitForElementPresent on css=#Results_Repeater__ctl1_Results_name > img with value 1000 Failed:16:42:59  
Unable to find the target element within the timeout specified.
```

ВАЖНО: Командата **wait for element present** е подходяща за използване при тестване на елементи с голям размер, например изображения.