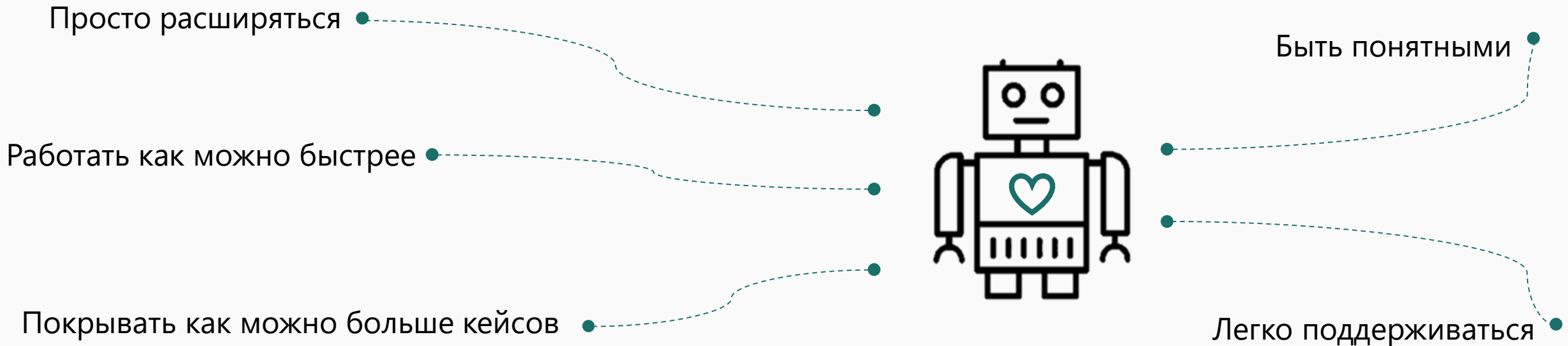


Тестирование интернет-магазина

<https://www.saucedemo.com/>

Цель проекта - создать автотесты, которые должны:





Планирование



Контроль качества кода



Проектирование



Документация



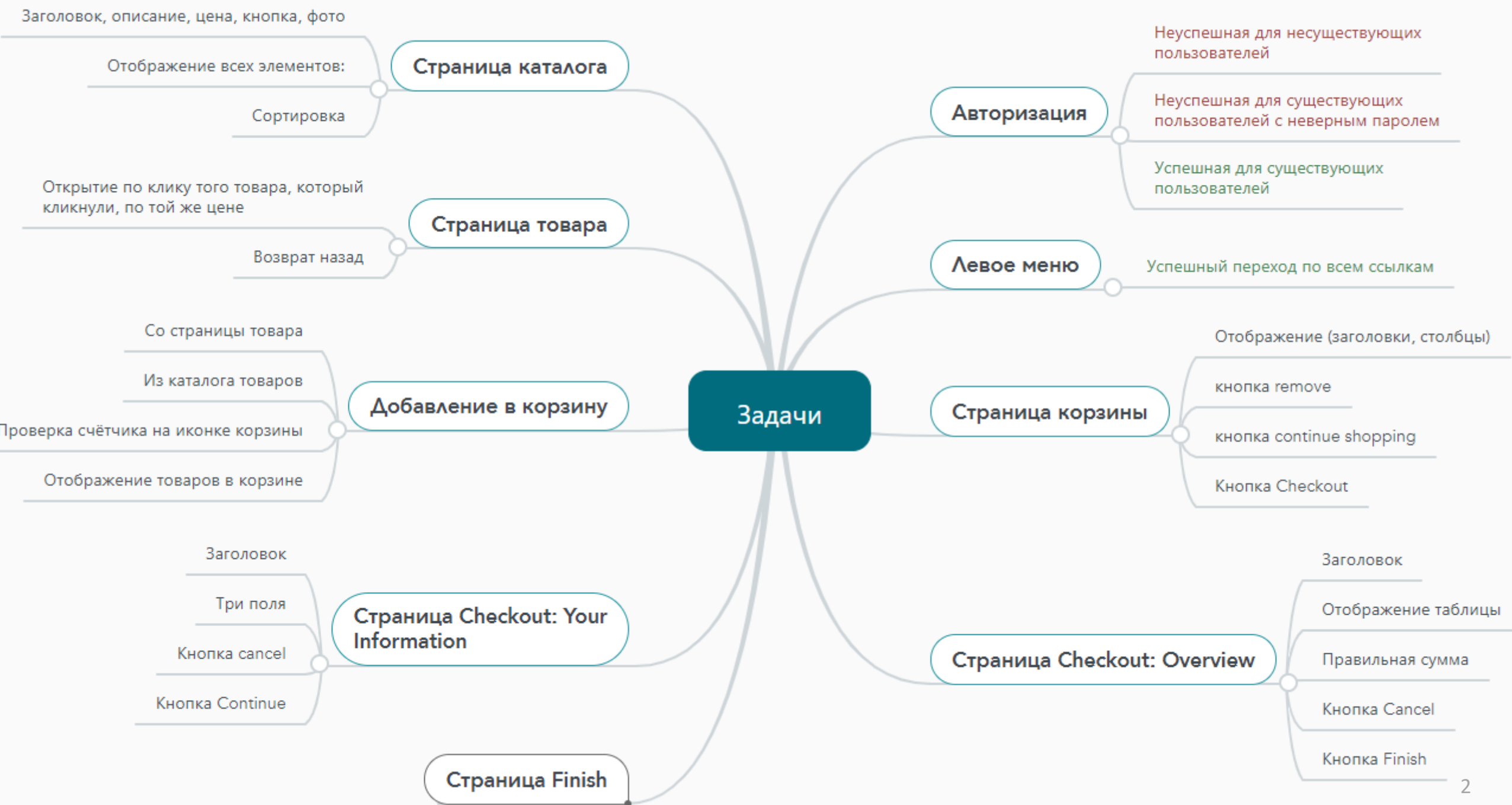
Инструменты





Отчёты



https://github.com/unotux/test_swag_shop









Планирование

Projects /  SwagLabs Testing /  SWAG-1

 1   

Тесты на авторизацию

 Attach  Create issue in epic  Link issue   Show Git development panel 









Description

Покрывать автотестами функционал авторизации разных пользователей в разных условиях

Issues in this epic

25% Done

- | | |
|---|--|
| <input checked="" type="checkbox"/> SWAG-2 Авторизация Happy path |   DONE |
| <input checked="" type="checkbox"/> SWAG-3 Неуспешная авторизация несуществующего пользователя |   BACKLOG |
| <input checked="" type="checkbox"/> SWAG-6 Существующие пользователи + неверный пароль |   BACKLOG |
| <input checked="" type="checkbox"/> SWAG-27 Негативный тест на доступ к страницам без авторизации |   IN PROGRESS |


Activity

Show: **Comments** History Work log Git Commits Git Roll Up



Add a comment...

Pro tip: press **M** to comment

Backlog 

Assignee  Анастасия


Reporter  Анастасия

Labels Auth

Priority  High

Epic Name Авторизация

Git Integration  Open Git Integration

 Show 5 more fields

Story Points, Original estimate, Time tracking, Components and Fix versions

Created February 7, 2021, 10:37 PM

Updated 1 minute ago

 Configure

Проектирование

Заранее думаем о том, каким образом структурировать код автотестов, чтобы избежать проблем с запутанностью, упростить поддержку, избежать дубликатов.

Page-object

Отделение логики тестов от логики браузера

Понимание, на какой странице находимся из кода

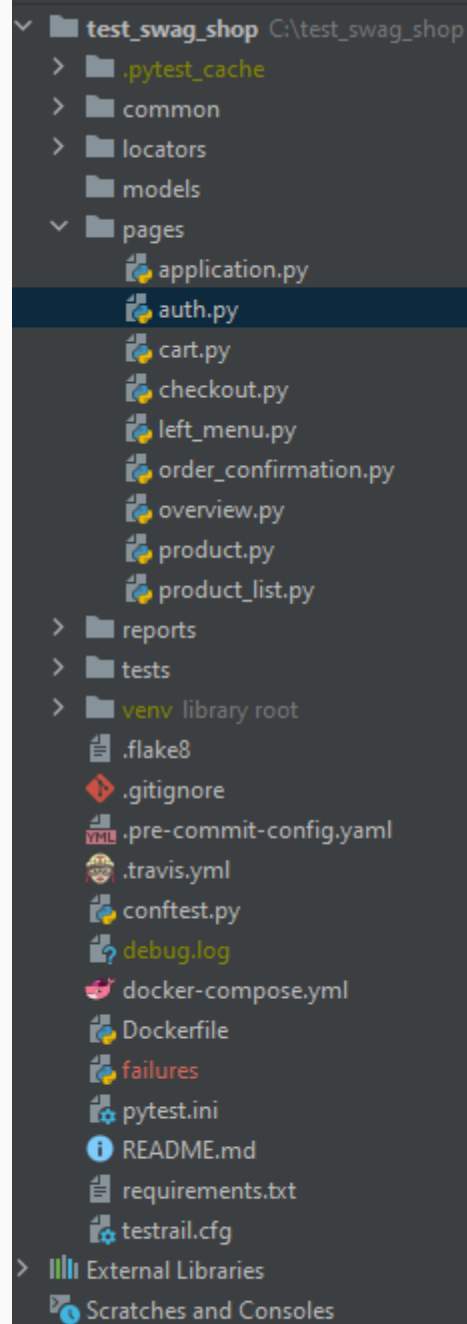
Переиспользование страниц

Page-element

Выделение элементов, повторяющихся на многих страницах, в отдельные объекты для переиспользования



Page-object



```
6 logger = logging.getLogger("app")
7
8
9 class LoginPage:
10     def __init__(self, app):
11         self.app = app
12
13     def username_field(self):
14         return self.app.driver.find_element(*p.USERNAME_INPUT)
15
16     def password_field(self):
17         return self.app.driver.find_element(*p.PASSWORD_INPUT)
18
19     def submit_button(self):
20         return self.app.driver.find_element(*p.LOGIN_SUBMIT)
21
22     def error_icon(self):
23         return self.app.driver.find_elements(*p.ERROR_ICON)
24
25     def error_text(self):
26         return self.app.driver.find_element(*p.ERROR_TEXT)
27
28     def get_error_text(self):
29         WebDriverWait(self.app.driver, 5).until(
30             EC.presence_of_element_located(p.ERROR_TEXT)
31         )
32         error_text = self.error_text().text
33         logger.info(f"Возникла ошибка {error_text}")
34         return error_text
35
36     def do_login(self, login, passwd):
37         logger.info(f"Вход с логином {login} и паролем {passwd}")
38         self.username_field().send_keys(login)
39         self.password_field().send_keys(passwd)
```

Инструменты

- Python + pytest – фреймворк для написания тестов
- Jira – управление поставленными задачами
- Teamcity – автоматический запуск тестов
- Testrail – хранение тест-кейсов
- Allure – построение отчётов по успешным\неуспешным тестам
- GitHub – хранение кода и управление версиями



Контроль качества кода





- Прекоммит хук
- Линтеры в CI

Teamcity CI

+ Add build step

Reorder build steps

✎ Auto-detect build steps

Build Step	Parameters Description	
1. Lint	Python Command: Flake8 Execute: If all previous steps finished successfully	Edit 
2. Run selenium container	Command Line Custom script: docker run -d --name selenium -p 4444:44... Execute: If all previous steps finished successfully	Edit 
3. Run tests	Python Command: Pytest Execute: If all previous steps finished successfully	Edit 
4. Stop and remove container	Command Line Custom script: docker stop selenium (and 1 more line) Execute: Always, even if build stop command was issued	Edit 



unotux remove travis config

● ce7035b 3 minutes ago ⌚ 64 commits



unotux remove travis config

✗ ce7035b 4 minutes ago ⌚ 64 commits

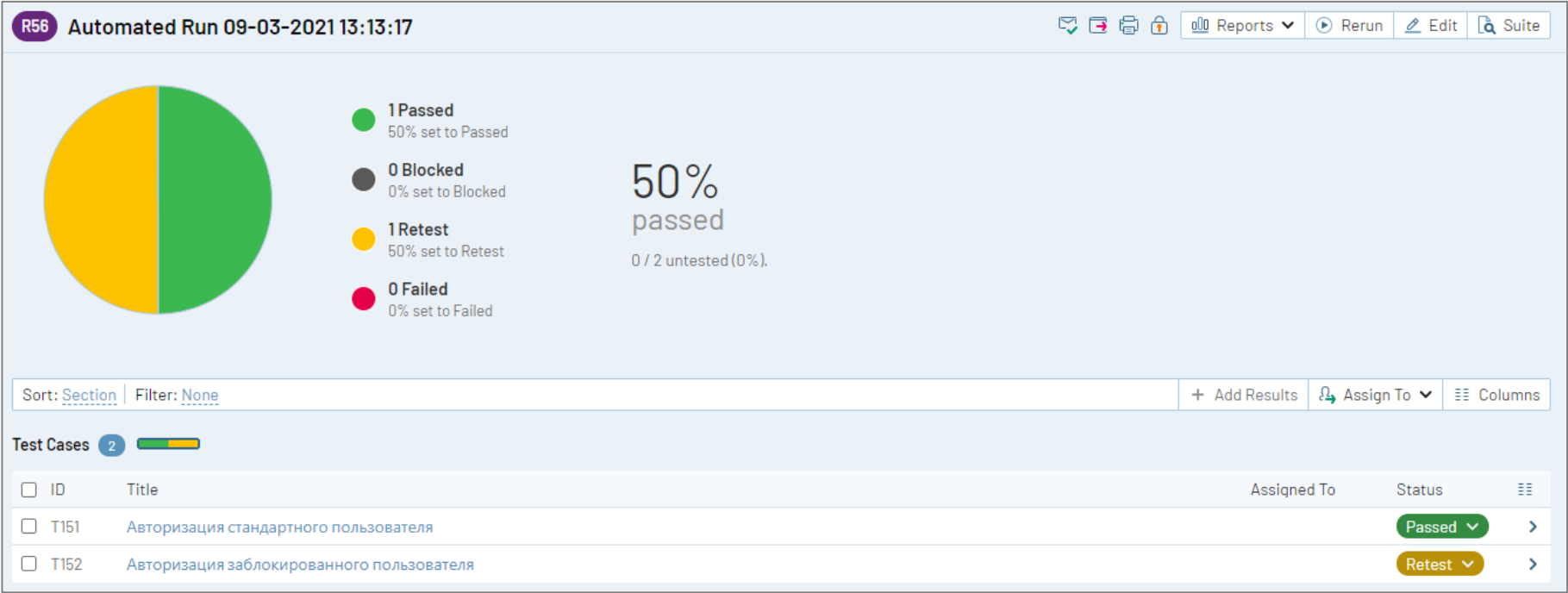


unotux remove travis config

✓ ce7035b 22 minutes ago ⌚ 64 commits

Документация

Testrail




Описание функций


```
def test_do_login_negative(self, app, login, passwd, expected_error):  
    """  
    Тесты для случаев, когда авторизация не должна пройти  
    1. Заблокированный пользователь  
    2. Несуществующий пользователь  
    3. Стандартный пользователь с неверным паролем  
  
    Во всех случаях результат: ошибка  
    """
```


Логирование


```
----- live log call -----  
2021-03-09 16:13:28 INFO Добавление товаров в корзину  
2021-03-09 16:13:29 INFO Переход в корзину  
2021-03-09 16:13:29 INFO Сумма товаров в корзине: 129.94  
2021-03-09 16:13:30 INFO Корзина очищена  
2021-03-09 16:13:30 INFO Сумма товаров в корзине: 0.0  
PASSED
```


Отчёты


**Allure**


 Overview


 Categories

 Suites

 Graphs





 Timeline



 Behaviors

 Packages

En

< Collapse

order  name  duration  status 

Status: 0 0 21 0 0 Marks:  

> Авторизация	5
▼ Корзина	3
> Возврат в каталог	1
> Переход к оформлению заказа	1
▼ Удаление из корзины	1
✓ #1 test_remove_from_cart	1s 675ms
▼ Оформление заказа	4
> Заполнение данных клиента	1
> Просмотр сформированного заказа	2
> Шаг заполнения данных клиента	1
▼ Страница каталога	6
> Добавление в корзину из каталога	1
> Наличие товаров на странице	1
> Сортировка	4
▼ Страница товара	3
> Переход из каталога на страницу товара	1

tests.test_cart.TestCart#test_remove_from_cart

Passed

test_remove_from_cart

Overview

HistoryRetries

Severity: blocker

Duration: 1s 675ms

Description

1. Добавляем все отображаемые товары в корзину
2. Переходим в корзину
3. Проверяем, что сумма не 0
2. Удаляем все товары из корзины
3. Отображается сумма 0

Execution

> Set up

▼ Test body

> log

> Tear down