

API: допинг и стабилизатор для UI-тестирования



Юлия
Атласова

Signant Health, Helsinki
Finland



@Julia_Atlasova



JuliaAtlasova@yandex.ru

HEISENBUG

Bio

- Peter-Service (Nexign): PL-SQL, Robot Framework + Selenium, Python
- Яндекс (Java, Selenium)
- Eram Systems (Java, C#, JavaScript, Selenium, Cypress)
- Signant Health (Python, Robot Framework, Selenium)

Agenda

- Проблемные точки и сильные стороны UI тестирования
- API слой – краткое описание возможностей
- API для авторизации
- API для тестирования бизнес-логики
- API для подготовки/ модификации/ удаления тестовых данных
- Вопросы и ответы

UI Tests: минусы и плюсы

МИНУСЫ

- Медленные
- Хрупкие
- Дорогие (долго разрабатываются)
- Трудно поддерживаемые

ПЛЮСЫ

- Наглядные
- Изложены на языке бизнеса
- Воспроизводимые вручную
- Симулируют реальные пользовательские сценарии
- Тестируют систему как единое целое

Есть способ ускорить и стабилизировать UI тесты!



Невероятно!... достаточно добавить простой...



С использованием API вызовов тесты:

- Работают быстрее
- Работают стабильнее
- Пишутся быстрее
- Поддерживаются легче
- Выполняют необходимые операции за меньшее количество шагов
- И это ещё не всё...

Проблема 1:

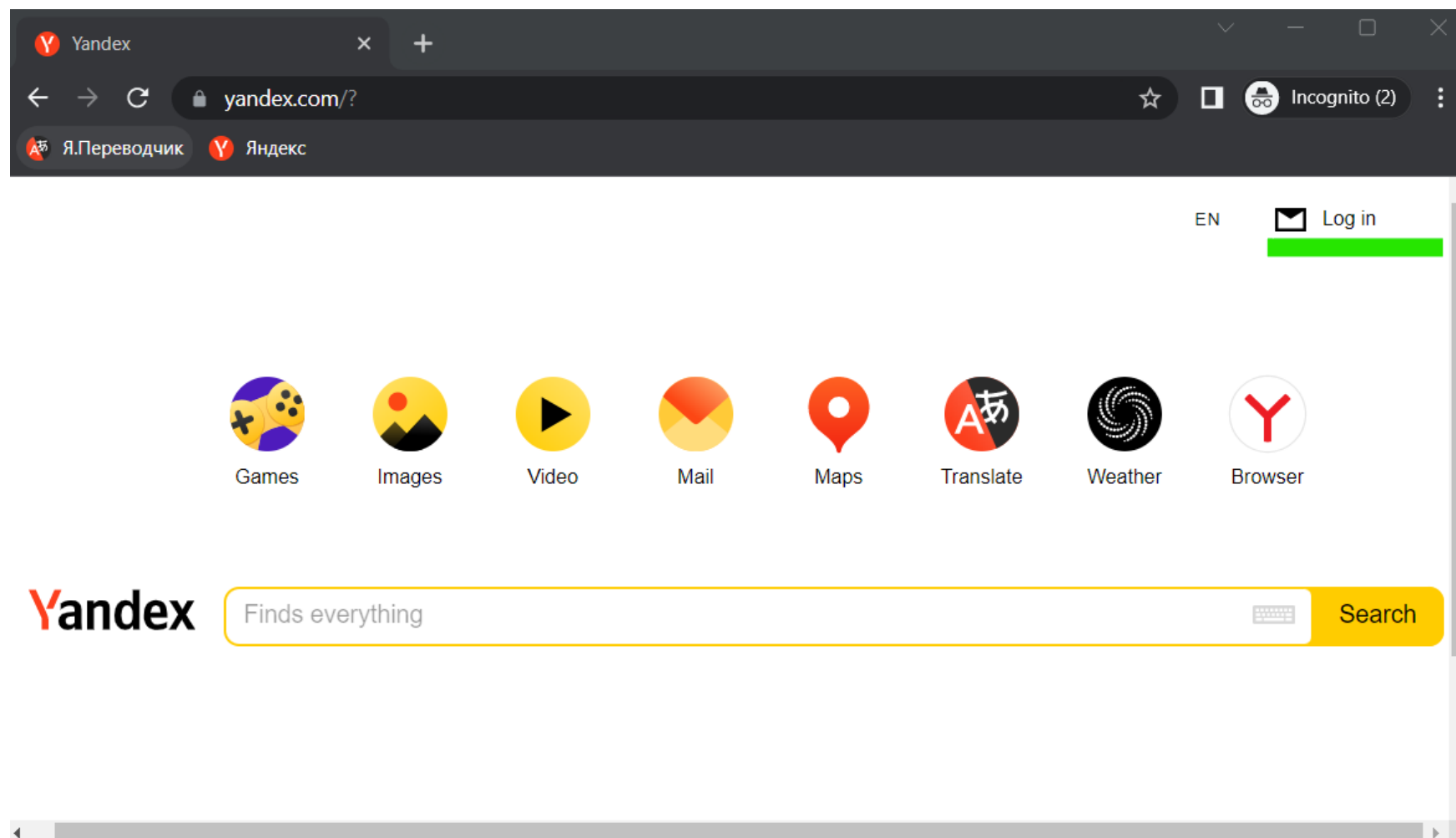
лишние UI шаги для
авторизации /
верификации пользователя

- Авторизация с логином и паролем
- Вход с подтверждающим кодом (смс или email)

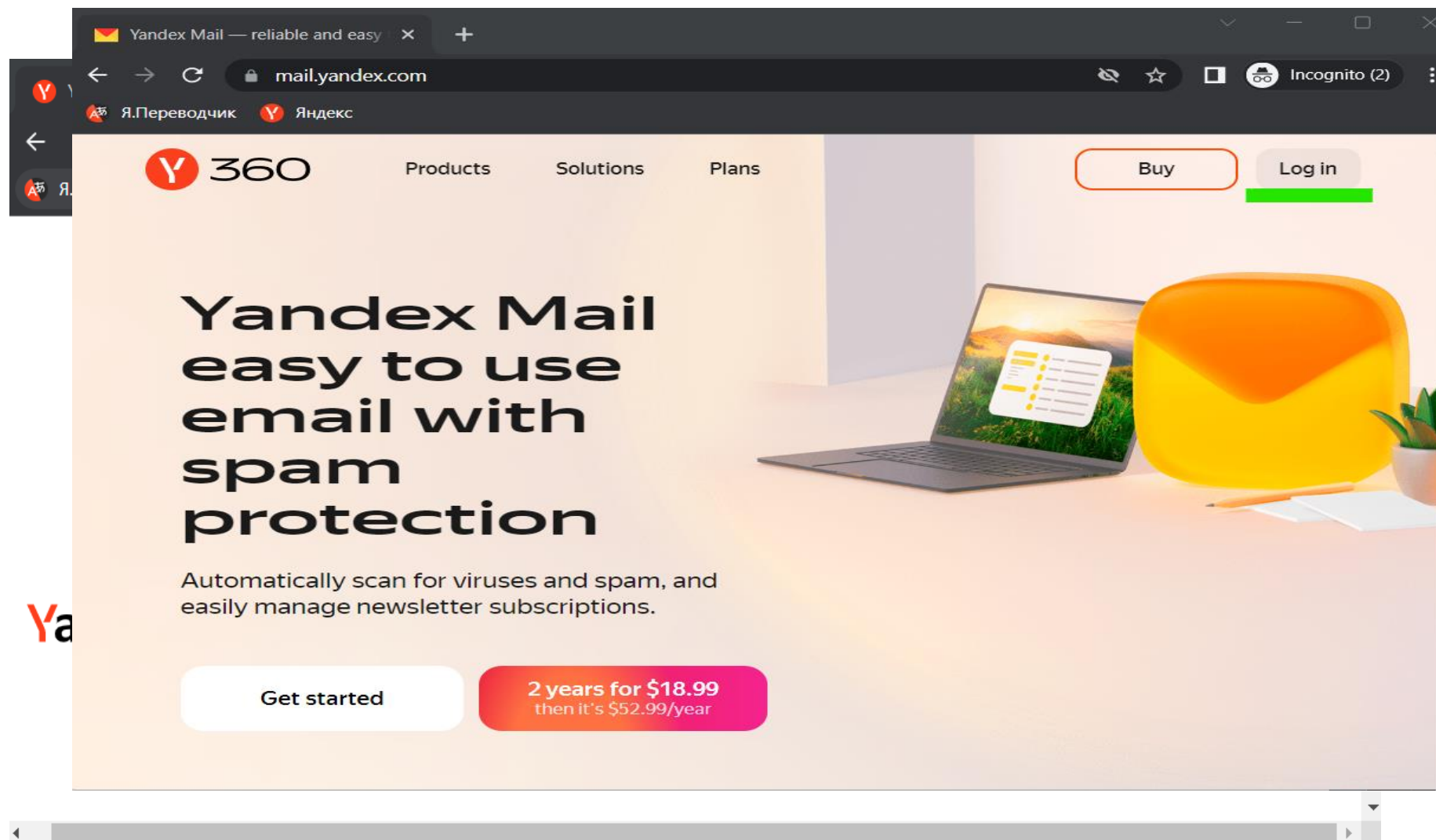
Типичный логин

- Состоит из 3 – 4 шагов
- Встречается в 95% UI тестов
- Тестируется в 2-3% тестов
- В случае поломки тормозит все тесты

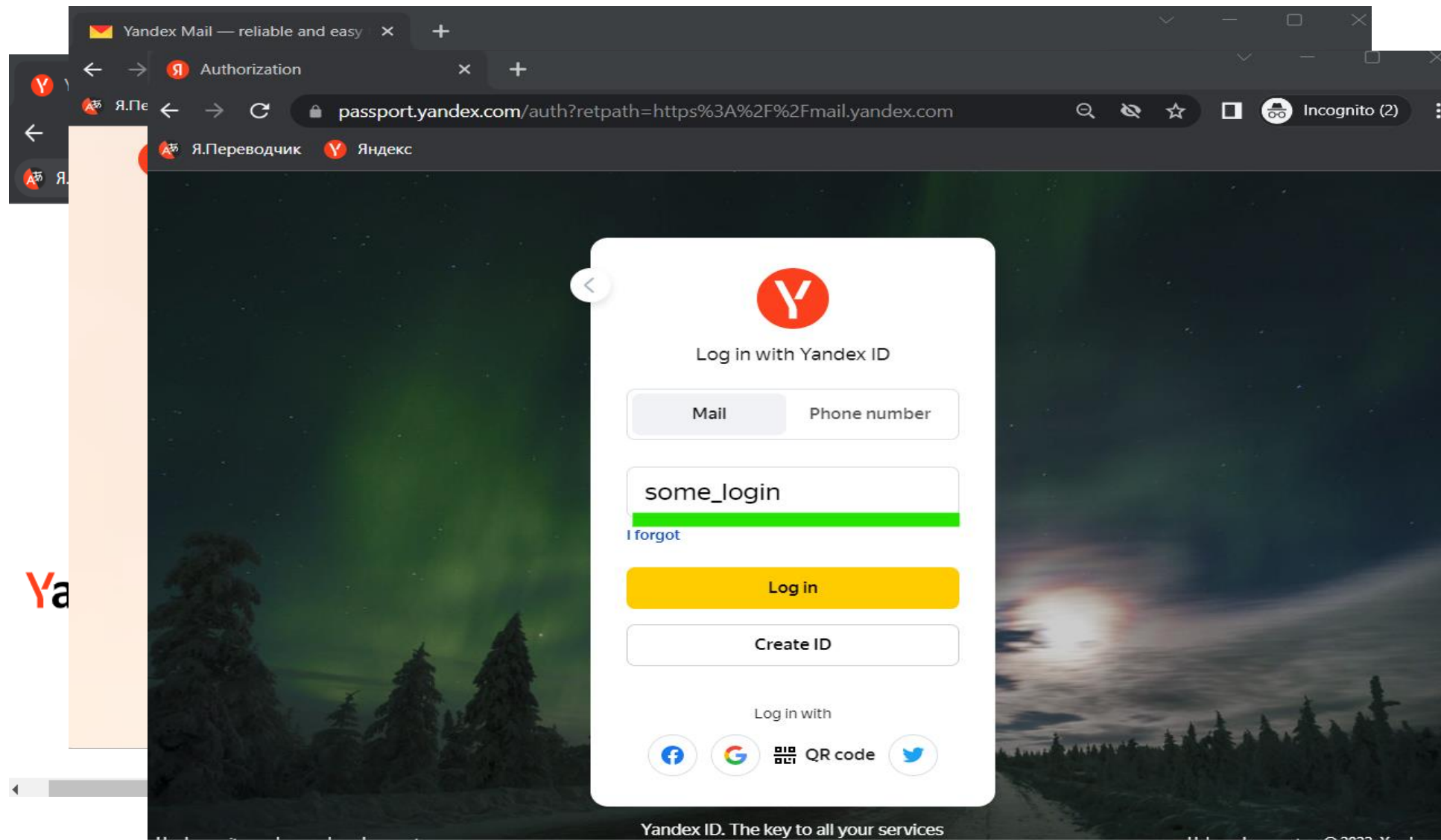
Типичный логин



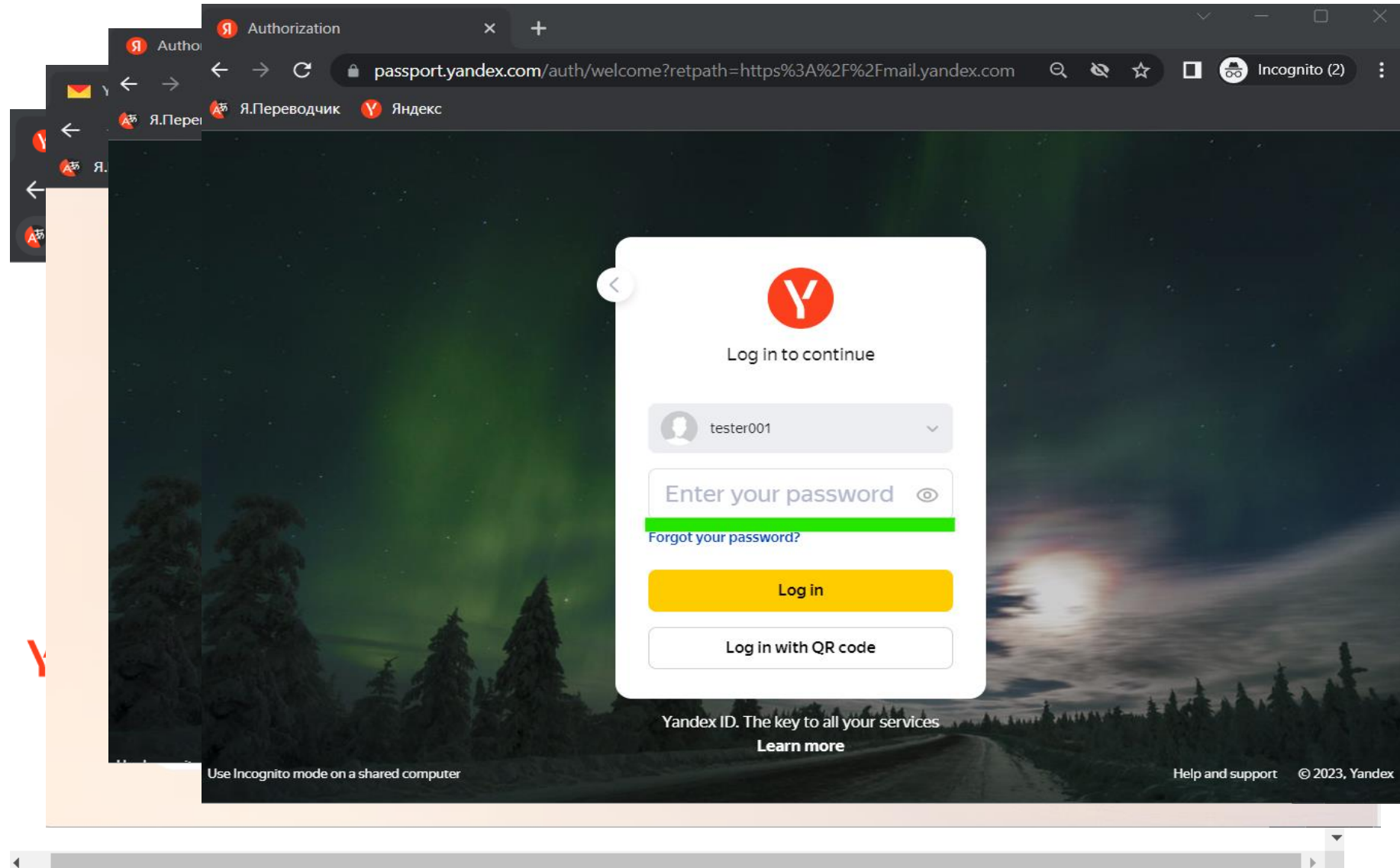
Типичный логин



Типичный логин



Типичный логин



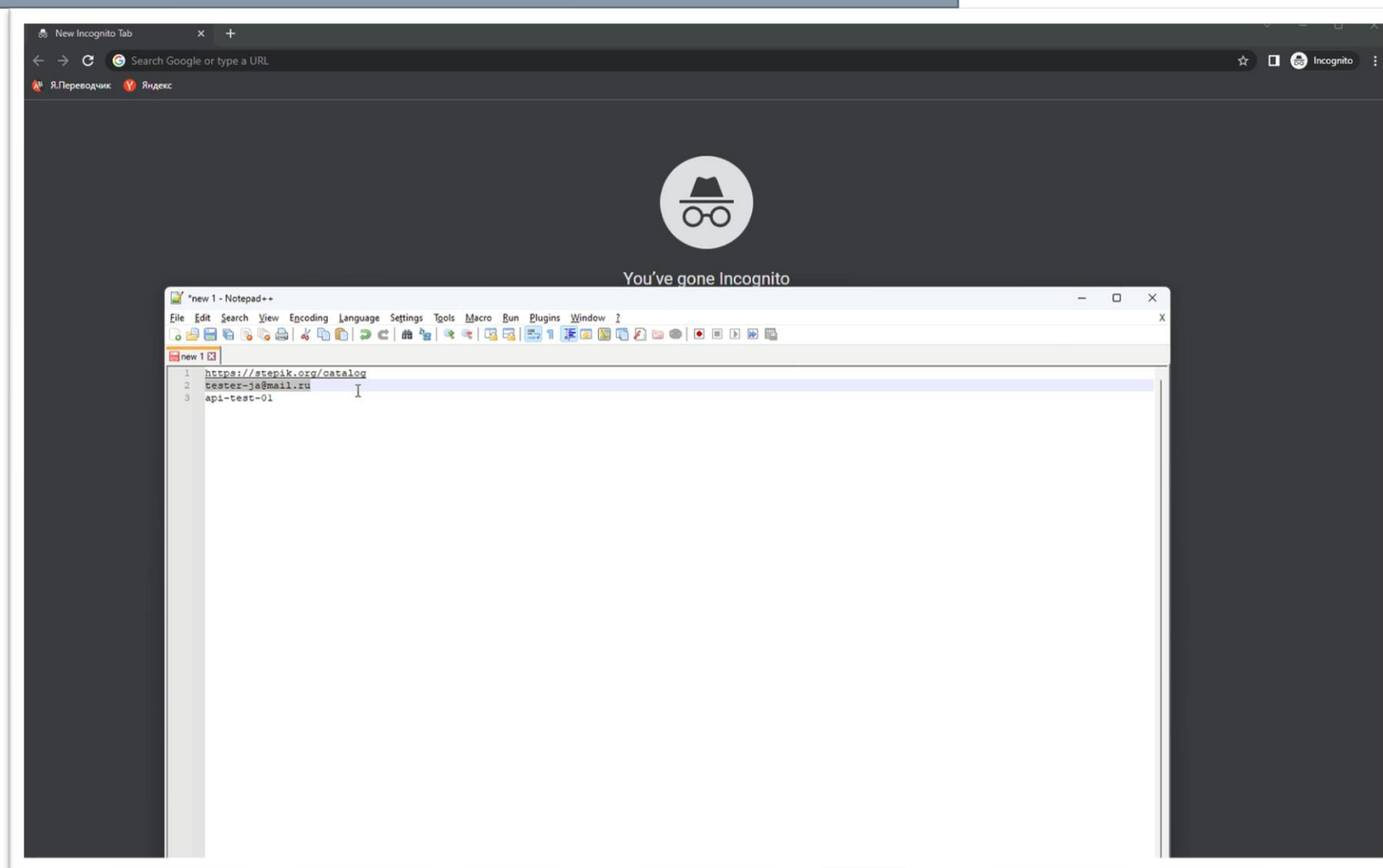
Типичный логин

Выберите все
изображения с современным
искусством





Тест для Stepik.org:



Настройки для сайта Stepik.org

The screenshot shows the PyCharm IDE interface. The top bar indicates the project is 'pythonUlandAPI' on the 'master' branch. The left sidebar shows the project structure:

- pythonProject [pythonUlandAPI] C:\Users\julia\PycharmProjects\p
 - github
 - stepic
 - api
 - resources
 - stepicSettings.py
 - steps
 - __init__.py
 - loginSteps.py
 - tests
 - __init__.py
 - apiLoginTest.py
 - uiLoginTest.py
 - trelo
 - utils
 - venv library root
 - External Libraries
 - Scratches and Consoles

The right pane shows the code in 'stepicSettings.py':

```
1 STEPIC_EMAIL = 'tester-ja@mail.ru'
2 STEPIC_PASSWORD = 'api-test-01'
3 STEPIC_WISHLIST_URL = 'https://stepik.org/learn/courses/wishlist'
4 STEPIC_FAVOURITES_URL = 'https://stepik.org/learn/courses/favorites'
5 STEPIC_HOME_URL = 'https://stepik.org/catalog'
6 STEPIC_AUTH_URL = 'https://stepik.org/api/users/login'
7
```




Тест для Stepik.org: UI login

```
pythonUlandAPI master Python tests for...UIUITest  
Project  
pythonProject [pythonUlandAPI]  
  github  
  api  
  resources  
  tests  
    invalidEmailApiTest.py  
    invalidEmailUiTest.py  
    validEmailApiTest.py  
    validEmailUiTest.py  
  stepic  
    api  
    resources  
      stepicSettings.py  
    steps  
      __init__.py  
      loginSteps.py  
    tests  
      __init__.py  
      apiLoginTest.py  
      uiLoginTest.py  
  trello  
  utils  
  venv library root  
  External Libraries  
  Scratches and Consoles  
Run Python tests for invalidEmailUiTest.ValidEmailUiTest  
pythonProject > stepic > tests > uiLoginTest.py  
16:41 CRLF UTF-8 4 spaces Python 3.11 (pythonProject)
```

```
7 from selenium.webdriver.support.wait import WebDriverWait  
8 from selenium.webdriver.support import expected_conditions as ec  
9  
10 class UiLoginTest(unittest.TestCase):  
11  
12  
13 def setUp(self) -> None:  
14     self.driver = webdriver.Chrome()  
15     self.driver.maximize_window()  
16     self.driver.get(STEPIC_HOME_URL)  
17  
18  
19 def test_check_favorites(self):  
20     stepic.steps.loginSteps.login_with_ui(self.driver)  
21     self.driver.get(STEPIC_FAVOURITES_URL)  
22     wait = WebDriverWait(self.driver, 10)  
23     wait.until(ec.visibility_of_element_located((By.CSS_SELECTOR, '.item-tile__title_with_badge >a')))  
24     favourites = self.driver.find_elements(By.CSS_SELECTOR, '.item-tile__title_with_badge >a')  
25     self.assertEqual(len(favourites), 2)  
26     self.assertEqual(favourites[0].text, 'Common English Verbs')  
27     self.assertEqual(favourites[1].text, 'TOEFL vocabulary')  
28  
29  
30 def tearDown(self):  
31     self.driver.quit()
```



Тест для Stepik.org: API login

The screenshot shows an IDE window with the following components:

- Project Explorer (Left):** Displays the project structure for 'pythonProject [pythonUlandAPI]'. The 'tests' directory is expanded, showing files like 'apiLoginTest.py' and 'uiLoginTest.py'.
- Editor (Center):** Shows the code for 'apiLoginTest.py'. The code includes imports for Selenium WebDriver and WebDriverWait, a class definition 'ApiLoginTest' inheriting from 'unittest.TestCase', and three methods: 'setUp', 'test_check_wishlist', and 'tearDown'. The 'test_check_wishlist' method performs an API login and asserts the wishlist contents.
- Run Console (Bottom):** Shows the command 'Run Python tests for uiLoginTest.UiLoginTest'.
- Status Bar (Bottom):** Displays the file path 'pythonProject > stepic > tests > apiLoginTest.py' and the Python version 'Python 3.11 (pythonProject)'.

```
7 from selenium.webdriver.support.wait import WebDriverWait
8 from selenium.webdriver.support import expected_conditions as ec
9
10 class ApiLoginTest(unittest.TestCase):
11
12
13 def setUp(self) -> None:
14     self.driver = webdriver.Chrome()
15     self.driver.maximize_window()
16     self.driver.get(stepicSettings.STEPIC_HOME_URL)
17
18
19 def test_check_wishlist(self):
20     stepic.steps.loginSteps.login_with_api(self.driver)
21     self.driver.get(STEPIC_WISHLIST_URL)
22     wait = WebDriverWait(self.driver, 10)
23     wait.until(ec.visibility_of_element_located((By.CSS_SELECTOR, '.item-tile__title_with_badge >a')))
24     wishlist = self.driver.find_elements(By.CSS_SELECTOR, '.item-tile__title_with_badge >a')
25     self.assertEqual(len(wishlist), 2)
26     self.assertEqual(wishlist[0].text, 'Algorithms and Data Structures')
27     self.assertEqual(wishlist[1].text, 'Data Structures')
28
29
30 def tearDown(self):
31     self.driver.quit()
```


Результаты тестов

Run Python tests for uiLoginTest.UiLoginTest x

✓ Test Results 18 sec 817 ms

- ✓ uiLoginTest 18 sec 817 ms
 - ✓ UiLoginTest 18 sec 817 ms
 - ✓ test_check_favorites 18 sec 817 ms

Run Python tests for apiLoginTest.ApiLoginTest x

✓ Test Results 13 sec 442 ms

- ✓ apiLoginTest 13 sec 442 ms
 - ✓ ApiLoginTest 13 sec 442 ms
 - ✓ test_check_wishlist 13 sec 442 ms

Cookies Stepik.org

The screenshot shows the Stepik.org website in an Incognito browser window. The page displays a 'Wishlist' section with two courses: 'Algorithms and Data Structures' by Adrian Spătaru and 'Data Structures' by Niema Moshiri, Liz Izhikevich. The browser's developer tools are open to the 'Application' tab, specifically the 'Cookies' section for the URL 'https://stepik.org'. A table of cookies is displayed, with the 'csrftoken' cookie highlighted. Below the table, the 'Cookie Value' is shown as 'HqecKkDG2qI967PZY0J7EtIb7m4vOnenyxysLVVH4gj2NDw1RQh2IZikb3boB8rI'.

Name	Value	Domain	P...	Expir...	Size	Http...	Secure	Sam...	Parti...	Pri...
sessionid	nu0admrf2p04j0i...	stepik.org	/	2023...	41	✓	✓	None	Medi...	
csrftoken	HqecKkDG2qI96...	stepik.org	/	2024...	73		✓	None	Medi...	
_ga	GA1.2.163458218...	.stepik.org	/	2024...	30				Medi...	
_ym_isad	2	.stepik.org	/	2023...	9		✓	None	Medi...	
_ym_visorc	w	.stepik.org	/	2023...	11		✓	None	Medi...	
_ym_d	1681412651	.stepik.org	/	2024...	15		✓	None	Medi...	
_ym_uid	16814126511964...	.stepik.org	/	2024...	26		✓	None	Medi...	
_ga_3VHTYDHHZK	GS1.1.168141265...	.stepik.org	/	2024...	52				Medi...	
_gid	GA1.2.677114042...	.stepik.org	/	2023...	30				Medi...	

Cookie Value ☐ Show URL decoded
HqecKkDG2qI967PZY0J7EtIb7m4vOnenyxysLVVH4gj2NDw1RQh2IZikb3boB8rI

Cookies Stepik.org

Application					
Filter					
Only show cookies v					
Name	Value	Domain	P...	Expi...	
sessionid	xqwu01xu2lxezx...	stepik.org	/	202...	
csrftoken	NoGdr4JoiWjQ2...	stepik.org	/	202...	
_gali	login_form	.stepik.org	/	202...	
_gat	1	.stepik.org	/	202...	
_ga	GA1.2.30482934...	.stepik.org	/	202...	
_ym_isad	2	.stepik.org	/	202...	
_ym_visorc	w	.stepik.org	/	202...	
_ym_d	1681573410	.stepik.org	/	202...	
_ym_uid	1681573410555...	.stepik.org	/	202...	
_ga_3YHTYDHHZK	GS1.1.16815734...	.stepik.org	/	202...	
_gid	GA1.2.69069865...	.stepik.org	/	202...	

Код теста

Project ▾

pythonProject [pythonUIandAPI] C:\

> github

▼ stepic

> api

▼ resources

stepicSettings.py

▼ steps

__init__.py

loginSteps.py

▼ tests

__init__.py

apiLoginTest.py

uiLoginTest.py

> trello

> utils

> venv library root

> External Libraries

Scratches and Consoles

stepicSettings.py ×

uiLoginTest.py

loginSteps.py

apiLoginTest.py ×

```
7 from selenium.webdriver.support.wait import WebDriverWait
8 from selenium.webdriver.support import expected_conditions as ec
9
10 class ApiLoginTest(unittest.TestCase):
11
12
13 def setUp(self) -> None:
14     self.driver = webdriver.Chrome()
15     self.driver.maximize_window()
16     self.driver.get(stepicSettings.STEPIC_HOME_URL)
17
18
19 def test_check_wishlist(self):
20     stepic.steps.loginSteps.login_with_api(self.driver)
21     self.driver.get(STEPIC_WISHLIST_URL)
22     wait = WebDriverWait(self.driver, 10)
23     wait.until(ec.visibility_of_element_located((By.CSS_SELECTOR, '.item-tile__title_with_badge >a')))
24     wishlist = self.driver.find_elements(By.CSS_SELECTOR, '.item-tile__title_with_badge >a')
25     self.assertEqual(len(wishlist), 2)
26     self.assertEqual(wishlist[0].text, 'Algorithms and Data Structures')
27     self.assertEqual(wishlist[1].text, 'Data Structures')
28
29
30 def tearDown(self):
31     self.driver.quit()
```

Код теста

The image shows a screenshot of an IDE with a project structure on the left and a Python script on the right.

Project Structure (Left Panel):

- pythonProject [pythonUlandAPI] C:\...
 - github
 - stepic
 - api
 - resources
 - stepicSettings.py
 - steps
 - __init__.py
 - loginSteps.py
 - tests
 - __init__.py
 - apiLoginTest.py
 - uiLoginTest.py
 - trello
 - utils
 - venv library root
 - External Libraries
 - Scratches and Consoles

Code Editor (Right Panel):

Files: stepicSettings.py, uiLoginTest.py, loginSteps.py, apiLoginTest.py

```
22 time.sleep(3)
23 wait.until(ec.visibility_of_element_located((By.CSS_SELECTOR, '.navbar__profile-img')))
24
25
26 1 usage Julia Atlasova
27 def login_with_api(driver: webdriver):
28     wait = WebDriverWait(driver, 10)
29     time.sleep(3)
30     wait.until(ec.visibility_of_element_located((By.CLASS_NAME, 'navbar__auth_login')))
31     all_cookies = driver.get_cookies()
32     new_cookies = loginApi.get_auth_cookies(all_cookies)
33     session_cookie = {
34         'name': 'sessionid',
35         'value': new_cookies.get('sessionid'),
36         'domain': 'stepik.org',
37         'path': '/',
38         'secure': True
39     }
40     csrftoken_cookie = {
41         'name': 'csrftoken',
42         'value': new_cookies.get('csrftoken'),
43         'domain': 'stepik.org',
44         'path': '/',
45         'secure': True
46     }
47     driver.add_cookie(session_cookie)
48     driver.add_cookie(csrftoken_cookie)
```

Код теста

Project ▾

pythonProject [pythonUlandAPI] C:\

> github

▾ stepic

> api

▾ resources

stepicSettings.py

▾ steps

__init__.py

loginSteps.py

▾ tests

__init__.py

apiLoginTest.py

uiLoginTest.py

> trello

> utils

> venv library root

> External Libraries

Scratches and Consoles

stepicSettings.py

uiLoginTest.py

loginSteps.py

loginApi.py ×

apiLoginTest.py

```
1 import requests
2 from stepic.resources.stepicSettings import STEPIC_EMAIL
3 from stepic.resources.stepicSettings import STEPIC_PASSWORD
4 from stepic.resources.stepicSettings import STEPIC_AUTH_URL
5
6 1 usage  Julia Atlasova
7 def get_auth_cookies(ui_cookies: list) -> dict:
8     current_session = requests.Session()
9     headers_dict: dict[str, str] = {}
10    for cookie in ui_cookies:
11        current_session.cookies.set(cookie['name'], cookie['value'], domain='stepik.org', path='/')
12        if cookie['name'] in 'sessionid, csrftoken':
13            headers_dict[cookie['name']] = cookie['value']
14            if cookie['name'] == 'csrftoken':
15                headers_dict['x-csrftoken'] = cookie['value']
16    credentials = {"email": STEPIC_EMAIL, "password": STEPIC_PASSWORD}
17    headers_dict['referer'] = 'https://stepik.org/catalog?auth=login'
18    headers_dict['authority'] = 'stepik.org'
19    headers_dict['origin'] = 'https://stepik.org'
20    req = requests.Request('POST', STEPIC_AUTH_URL, json=credentials, headers=headers_dict)
21    prepped = current_session.prepare_request(req)
22    resp = current_session.send(prepped)
23    result_cookies = {
24        'sessionid': resp.cookies.get('sessionid'),
25        'csrftoken': resp.cookies.get('csrftoken')
26    }
27    current_session.close()
28    return result_cookies
```

Проблема 1:

лишние UI шаги для авторизации /
верификации пользователя

Решение:

выполняйте шаги авторизации
пользователя вызовом API, в браузер
добавляйте полученные от API cookies

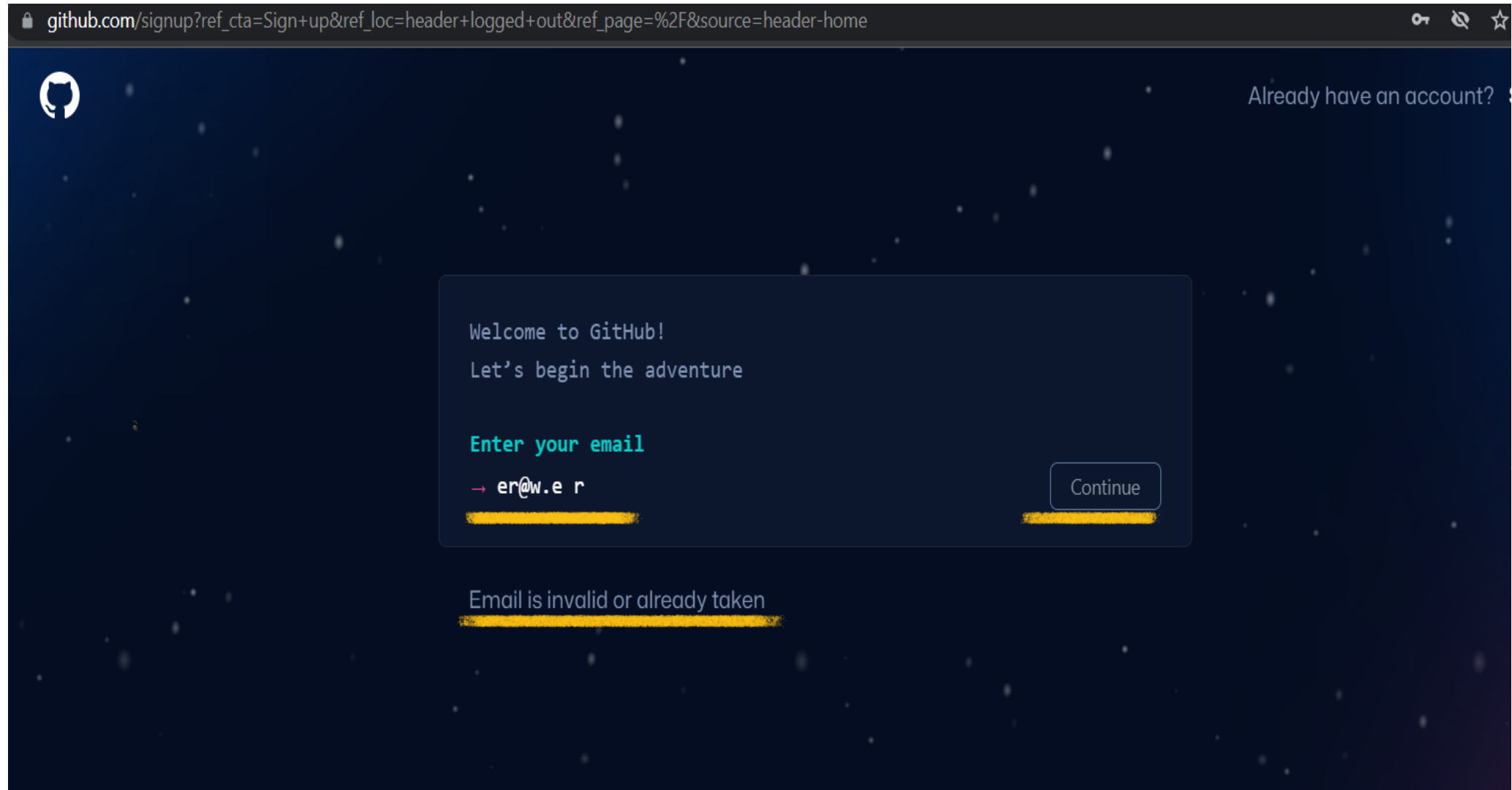
Проблема 2:

лишние тесты для валидации пользовательского ввода

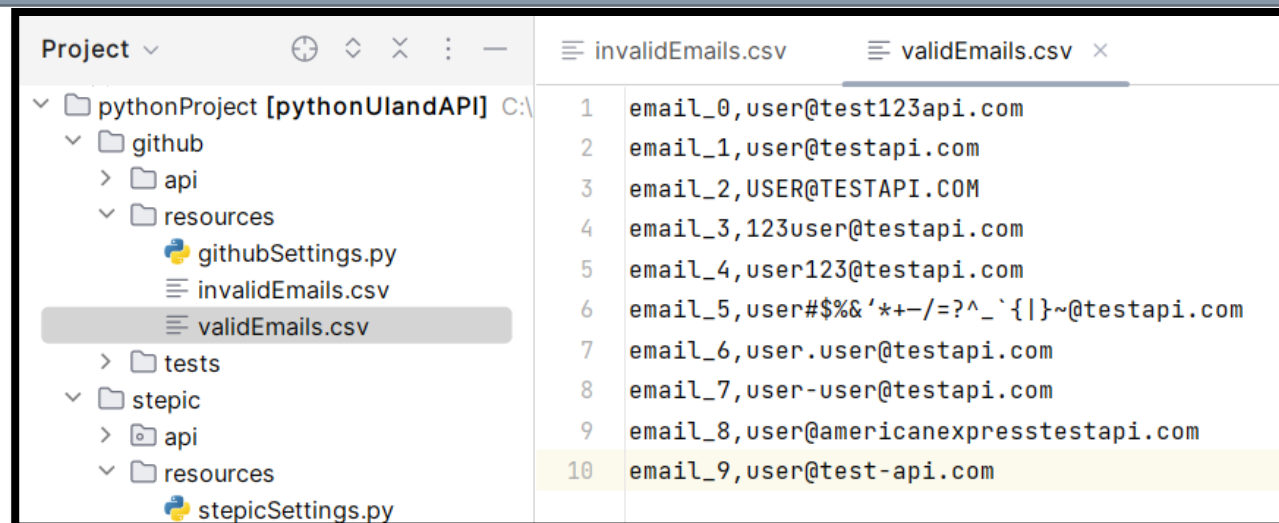
- Логин
- Email
- Номер телефона
- Даты
- ...

Проблема: лишние UI тесты.

Валидация полей



Тестовые данные

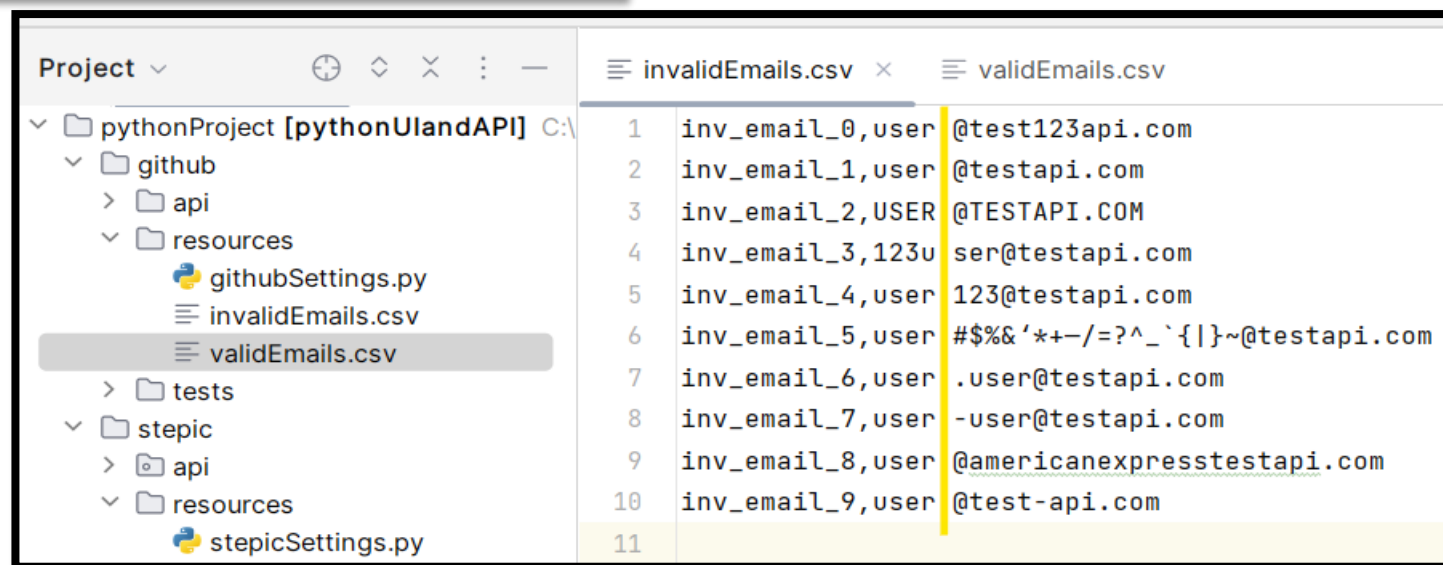


Project ▾

- pythonProject [pythonUlandAPI] C:\
 - github
 - api
 - resources
 - githubSettings.py
 - invalidEmails.csv
 - validEmails.csv
 - tests
 - stepic
 - api
 - resources
 - stepicSettings.py

invalidEmails.csv

```
1 email_0,user@test123api.com
2 email_1,user@testapi.com
3 email_2,USER@TESTAPI.COM
4 email_3,123user@testapi.com
5 email_4,user123@testapi.com
6 email_5,user#$$% '*+~/=?^_`{|}~@testapi.com
7 email_6,user.user@testapi.com
8 email_7,user-user@testapi.com
9 email_8,user@americanexpresstestapi.com
10 email_9,user@test-api.com
```



Project ▾

- pythonProject [pythonUlandAPI] C:\
 - github
 - api
 - resources
 - githubSettings.py
 - invalidEmails.csv
 - validEmails.csv
 - tests
 - stepic
 - api
 - resources
 - stepicSettings.py

invalidEmails.csv

```
1 inv_email_0,user @test123api.com
2 inv_email_1,user @testapi.com
3 inv_email_2,USER @TESTAPI.COM
4 inv_email_3,123u ser@testapi.com
5 inv_email_4,user 123@testapi.com
6 inv_email_5,user #$$% '*+~/=?^_`{|}~@testapi.com
7 inv_email_6,user .user@testapi.com
8 inv_email_7,user -user@testapi.com
9 inv_email_8,user @americanexpresstestapi.com
10 inv_email_9,user @test-api.com
11
```



Тест для валидных значений UI:

The screenshot displays the PyCharm IDE interface. The top toolbar shows the 'Run' button (a green play icon). The left sidebar shows the project structure for 'pythonUlandAPI', with the 'tests' directory expanded. The main editor window shows the code for 'validEmailUiTest.py'. The code includes imports for 'START_URL', 'debug_print', and 'load_csv_test_cases'. It defines a 'ValidEmailUiTest' class inheriting from 'unittest.TestCase'. The 'setUpClass' method is decorated with '@classmethod' and initializes a 'driver' using 'webdriver.Chrome()'. The bottom panel shows the 'Run' output, indicating that 1 of 10 tests failed. The test results list shows 'validEmailUiTest' failed, with 'ValidEmailUiTest' and 'setUpClass' as the failing tests. The console output shows the command used to run the tests and the execution time of 19.901s.

```
7
8 from github.resources.githubSettings import START_URL
9 from utils.debug import debug_print
10 from utils.file_loader import load_csv_test_cases
11
12
13 class ValidEmailUiTest(unittest.TestCase):
14     driver = None
15
16     @classmethod
17     def setUpClass(cls):
18         cls.driver = webdriver.Chrome()
19         cls.driver.maximize_window()
20         cls.driver.get(START_URL)
```

Run Python tests for validEmailUiTest.ValidEmailUiTest

Tests failed: 1 of 10 tests

Test Results

- validEmailUiTest
- ValidEmailUiTest
- setUpClass

C:\Users\julia\PycharmProjects\pythonProject\venv\Scripts\python.exe "C:/Program Files/JetBrains/PyCharm Community Edition 2023.1/plugins/python-ce/helpers/pycharm/_jb_unittest_runner.py" --target validEmailUiTest.ValidEmailUiTest

Testing started at 1:03 PM ...

Launching unittests with arguments python -m unittest validEmailUiTest.ValidEmailUiTest in C:\Users\julia\PycharmProjects\pythonProject\github\tests

Ran 0 tests in 19.901s

pythonProject > github > tests > validEmailUiTest.py CRLF UTF-8 4 spaces Python 3.11 (pythonProject)



Тест для невалидных значений UI:

The screenshot displays the PyCharm IDE interface. The top toolbar shows the 'Run' button (a green play icon). The left sidebar shows the project structure for 'pythonUlandAPI', with the 'tests' folder expanded, highlighting 'invalidEmailUiTest.py'. The main editor window shows the code for 'InvalidEmailUiTest' in 'invalidEmailUiTest.py'. The code includes imports for Selenium WebDriver and custom utility functions, followed by a class definition and a setup method.

```
4 from selenium.webdriver.common.by import By
5 from selenium.webdriver.support import expected_conditions as ec
6 from selenium.webdriver.support.wait import WebDriverWait
7
8 from github.resources.githubSettings import START_URL
9 from utils.debug import debug_print
10 from utils.file_loader import load_csv_test_cases
11
12
13 class InvalidEmailUiTest(unittest.TestCase):
14     driver = None
15
16     @classmethod
17     def setUpClass(cls):
```

Below the code editor, the 'Run' tool window is open, showing the test results. The status bar at the bottom indicates 'Tests passed: 10 of 10 tests - 9 sec 110 ms'. The test results table shows the following details:

Test Name	Duration
invalidEmailUiTest	9 sec 110 ms
InvalidEmailUiTest	9 sec 110 ms
test_invalid_email_0_inv_email_0	881 ms
test_invalid_email_1_inv_email_1	863 ms
test_invalid_email_2_inv_email_2	829 ms
test_invalid_email_3_inv_email_3	862 ms
test_invalid_email_4_inv_email_4	927 ms
test_invalid_email_5_inv_email_5	933 ms
test_invalid_email_6_inv_email_6	817 ms
test_invalid_email_7_inv_email_7	803 ms

The console output shows the command used to run the tests and the resulting log messages:

```
C:\Users\julia\PycharmProjects\pythonProject\venv\Scripts\python.exe "C:/Program Files/JetBrains/PyCharm Community Edition
2023.1/plugins/python-ce/helpers/pycharm/_jb_unittest_runner.py" --target invalidEmailUiTest.InvalidEmailUiTest
Testing started at 1:09 PM ...
Launching unittests with arguments python -m unittest invalidEmailUiTest.InvalidEmailUiTest in
C:\Users\julia\PycharmProjects\pythonProject\github\tests
----- driver connection open -----
----- driver connection closed -----
```



2 теста для всех значений API:

The screenshot displays the PyCharm IDE interface. The top toolbar shows the 'Run' button (a green play icon). The left sidebar shows the project structure for 'pythonUlandAPI', with the 'tests' directory expanded. The main editor window shows the code for 'validEmailApiTest.py' and 'invalidEmailApiTest.py'. The code includes imports for 'debug_print', 'load_csv_test_cases', 'get_request_token', and 'LOGIN_VALIDATION_URL'. It defines a 'ValidEmailApiTest' class inheriting from 'unittest.TestCase'. The 'setUpClass' method is decorated with '@classmethod' and sets up a 'requests.Session()' and a 'token' using 'get_request_token'. The 'Run' console at the bottom shows the execution of 'Python tests for invalidEmailUiTest.InvalidEmailUiTest'. The console output indicates that all 10 tests passed in 8 seconds and 429 milliseconds. The test results are listed in a table:

Test Name	Duration
invalidEmailUiTest	8 sec 429 ms
InvalidEmailUiTest	8 sec 429 ms
test_invalid_email_0_inv_email_0	895 ms
test_invalid_email_1_inv_email_1	806 ms
test_invalid_email_2_inv_email_2	778 ms
test_invalid_email_3_inv_email_3	937 ms
test_invalid_email_4_inv_email_4	819 ms
test_invalid_email_5_inv_email_5	852 ms











The Run console also shows the command used to launch the tests: 'C:\Users\julia\PycharmProjects\pythonProject\venv\Scripts\python.exe "C:/Program Files/JetBrains/PyCharm Community Edition/2023.1/plugins/python-ce/helpers/pycharm/_jb_unittest_runner.py" --target invalidEmailUiTest.InvalidEmailUiTest'. The output includes the message 'Testing started at 1:14 PM ...' and 'Launching unittests with arguments python -m unittest invalidEmailUiTest.InvalidEmailUiTest in C:\Users\julia\PycharmProjects\pythonProject\github\tests'. The console also shows the output of the 'setUpClass' method: '----- driver connection open -----'.











Результаты UI тестов

Run Python tests for invalidEmailUiTest.InvalidEmailUiTest	
✓ Test Results	9 sec 244 ms
✓ invalidEmailUiTest	9 sec 244 ms
✓ InvalidEmailUiTest	9 sec 244 ms
✓ test_invalid_email_0_inv_email_0	1 sec 12 ms
✓ test_invalid_email_1_inv_email_1	871 ms
✓ test_invalid_email_2_inv_email_2	780 ms
✓ test_invalid_email_3_inv_email_3	877 ms
✓ test_invalid_email_4_inv_email_4	1 sec 38 ms
✓ test_invalid_email_5_inv_email_5	916 ms
✓ test_invalid_email_6_inv_email_6	870 ms
✓ test_invalid_email_7_inv_email_7	985 ms
✓ test_invalid_email_8_inv_email_8	972 ms
✓ test_invalid_email_9_inv_email_9	923 ms

Run Python tests for validEmailUiTest.ValidEmailUiTest	
✓ Test Results	8 sec 502 ms
✓ validEmailUiTest	8 sec 502 ms
✓ ValidEmailUiTest	8 sec 502 ms
✓ test_valid_email_0_email_0	834 ms
✓ test_valid_email_1_email_1	814 ms
✓ test_valid_email_2_email_2	788 ms
✓ test_valid_email_3_email_3	818 ms
✓ test_valid_email_4_email_4	778 ms
✓ test_valid_email_5_email_5	930 ms
✓ test_valid_email_6_email_6	846 ms
✓ test_valid_email_7_email_7	947 ms
✓ test_valid_email_8_email_8	917 ms
✓ test_valid_email_9_email_9	830 ms

Результаты API тестов

Run Python tests for invalidEmailApiTest.InvalidEmailApiTest	
   	
     	
✓ Test Results	2 sec 190 ms
✓ invalidEmailApiTest	2 sec 190 ms
✓ InvalidEmailApiTest	2 sec 190 ms
✓ test_valid_email_0_inv_email_0	265 ms
✓ test_valid_email_1_inv_email_1	202 ms
✓ test_valid_email_2_inv_email_2	168 ms
✓ test_valid_email_3_inv_email_3	178 ms
✓ test_valid_email_4_inv_email_4	258 ms
✓ test_valid_email_5_inv_email_5	306 ms

Run Python tests for validEmailApiTest.ValidEmailApiTest	
   	
     	
✓ Test Results	1 sec 968 ms
✓ validEmailApiTest	1 sec 968 ms
✓ ValidEmailApiTest	1 sec 968 ms
✓ test_valid_email_0_email_0	233 ms
✓ test_valid_email_1_email_1	205 ms
✓ test_valid_email_2_email_2	204 ms
✓ test_valid_email_3_email_3	207 ms
✓ test_valid_email_4_email_4	202 ms
✓ test_valid_email_5_email_5	202 ms



Механика теста:

The screenshot shows a web browser window with the GitHub sign-up page. The network developer tools are open, showing a single request named 'stats' with a status of 200, type of ping, and a time of 155 ms. The page content includes a welcome message, a sign-in link, and a sign-up form with a 'Continue' button.

Join GitHub · GitHub

github.com/signup?ref_cta=Sign+up&ref_loc=header+logged+out&ref_page=%2F&source=header-home

Incognito

Elements Console Sources Network Performance Memory Application Security

Filter ☐ Invert ☐ Hide data URLs ☒ All Fetch/XHR JS CSS Img Media Font Doc WS Wasm Manifest Other

☐ Has blocked cookies ☐ Blocked Requests ☐ 3rd-party requests

20 ms 40 ms 60 ms 80 ms 100 ms 120 ms 140 ms 160 ms

Name	Status	Type	Initiator	Size	Time	Waterfall
<input type="checkbox"/> stats	200	ping	stats.ts:90	0 B	155 ms	

1 requests | 0 B transferred | 0 B resources

Already have an account? [Sign in](#) →

Welcome to GitHub!
Let's begin the adventure

Enter your email
→ ert@mai.tu

Continue

By creating an account, you agree to the [Terms of Service](#).
For more information about GitHub's privacy practices,
see the [GitHub Privacy Statement](#). We'll occasionally send
you account-related emails.

Механика теста:

The screenshot shows the Chrome DevTools Network tab. The top toolbar includes tabs for Elements, Console, Sources, Network (selected), Performance, Memory, Application, Security, and Lighthouse. The Network tab shows a list of requests, all named 'email_validity_checks'. The first request is selected, and its details are shown in the right pane. The details pane has tabs for Headers, Payload (selected), Preview, Response, Initiator, Timing, and Cookies. The Payload tab shows the form data, including an 'authenticity_token' and a 'value'.

Network tab toolbar: Elements, Console, Sources, Network, Performance, Memory, Application, Security, Lighthouse. 7 errors, 2 messages.

Filter: ☐ Invert ☐ Hide data URLs ☒ All Fetch/XHR JS CSS Img Media Font Doc WS Wasm Manifest Other ☐ Has blocked cookies

☐ Blocked Requests ☐ 3rd-party requests

Timeline: 500 ms, 1000 ms, 1500 ms, 2000 ms, 2500 ms, 3000 ms, 3500 ms, 4000 ms, 4500 ms, 5000 ms, 5500 ms, 6000 ms, 6500 ms, 7000 ms, 7500 ms

Name	Headers	Payload	Preview	Response	Initiator	Timing	Cookies
<input type="checkbox"/> email_validity_checks		▼ Form Data view source view decoded					
<input type="checkbox"/> email_validity_checks		authenticity_token: gWb09c7GFUtVwhiRdrRnIB3is5TnnTDdHYyW0e//pNH2ncyUqHH4oeLMMyhTt3Pqw9bzq2Rkuu3mWBGyo/10MA==					
<input type="checkbox"/> email_validity_checks		value: sdfsd@yan.ro					
<input type="checkbox"/> email_validity_checks							
<input type="checkbox"/> email_validity_checks							
<input type="checkbox"/> email_validity_checks							
<input type="checkbox"/> email_validity_checks							

Код теста

The screenshot shows an IDE with a project structure on the left and a Python test file on the right. The project structure is as follows:

- pythonProject [pythonUlandAPI] C:\Users\julia\...
 - github
 - api
 - resources
 - githubSettings.py
 - invalidEmails.csv
 - validEmails.csv
 - tests
 - invalidEmailApiTest.py
 - invalidEmailUiTest.py
 - validEmailApiTest.py
 - validEmailUiTest.py
 - stepic
 - trello
 - utils
 - venv library root
 - External Libraries
 - Scratches and Consoles

The right pane shows the code for `validEmailApiTest.py`. The code is as follows:

```
7 from github.resources.githubSettings import LOGIN_VALIDATION_URL
8
9 class ValidEmailApiTest(unittest.TestCase):
10     connection = None
11
12     @classmethod
13     def setUpClass(cls):
14         cls.connection = requests.Session()
15         cls.token = get_request_token(open_session=cls.connection)
16         debug_print('connection open', '')
17
18     @classmethod
19     def tearDownClass(cls):
20         cls.connection.close()
21         debug_print('connection closed', '')
22
23     @parameterized.expand(load_csv_test_cases('../resources/validEmails.csv'))
24     def test_valid_email(self, name, valid_email):
25         debug_print('test name = ', name)
26         payload = {"authenticity_token": self.token, "value": valid_email}
27         resp = self.connection.post(LOGIN_VALIDATION_URL, data=payload)
28         assert resp.status_code == requests.codes.ok
29
```

Код теста

Project ▾

⊕ ⊖ ✕ ⋮ —

pythonProject [pythonUlandAPI] C:\Users\julia\Documents\pythonProject

github

api

resources

githubSettings.py

invalidEmails.csv

validEmails.csv

tests

invalidEmailApiTest.py

invalidEmailUiTest.py

validEmailApiTest.py

validEmailUiTest.py

stepic

trello

utils

venv library root

External Libraries

Scratches and Consoles

validEmailApiTest.py

getUnauthToken.py ✕

1 import requests

2 from github.resources.githubSettings import START_URL

3 from github.resources.githubSettings import REQUEST_TOKEN_URL

4

5

4 usages Julia Atlasova

6 def get_request_token(open_session: requests.Session) -> str:

7 open_session.get(START_URL)

8 resp = open_session.get(REQUEST_TOKEN_URL)

9 page_with_token = resp.text

10 before_token1 = 'type="email" name="user[email]" />'

11 found_token_string = page_with_token.find(before_token1)

12 assert not found_token_string == -1

13 first_split_res = page_with_token.split(before_token1)[1]

14 before_token2 = 'value="'

15 second_split_res = first_split_res.split(before_token2)[1]

16 token = second_split_res.split('"')[0]

17 print(token)

18 return token

19

Проблема 2:

лишние тесты для валидации
пользовательского ввода

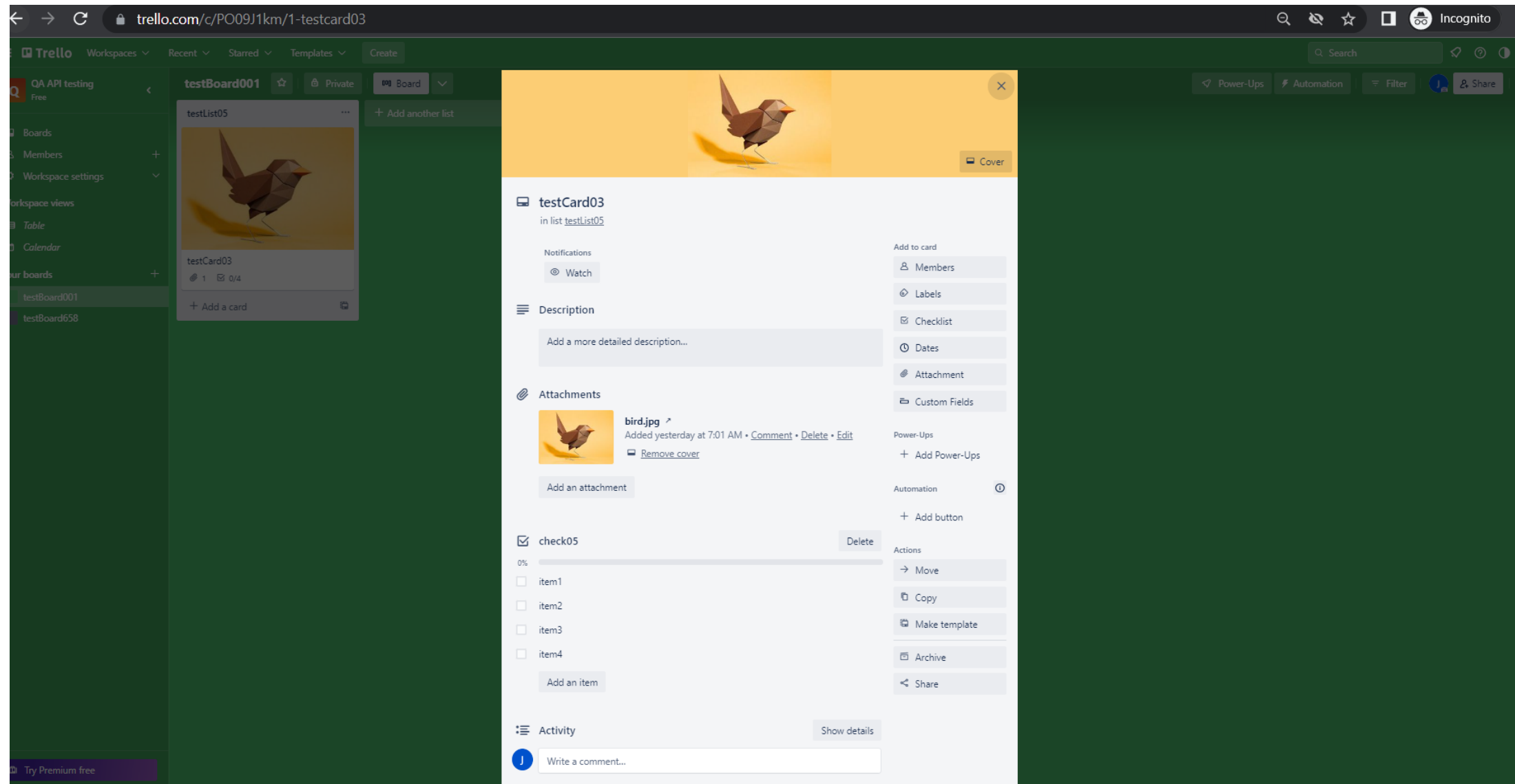
Решение:

пишите минимум UI тестов для
проверки сообщений на UI, остальные
проверки выполняйте вызовом API

Проблема 3:

создание тестовых данных
через UI

Проблема: подготовка тестовых данных в UI.



The screenshot shows a Trello board interface in a web browser. The URL bar indicates the board is 'testcard03' in a workspace 'c/PO09J1km/1-testcard03'. The board itself is titled 'testBoard001' and is set to 'Private'. A list named 'testList05' contains a card titled 'testCard03'. The card features a bird image as a cover and a checklist titled 'check05' with four items: 'item1', 'item2', 'item3', and 'item4'. The card also has an attachment named 'bird.jpg' and a description field. The right sidebar offers various actions for the card, including adding members, labels, checklists, dates, attachments, custom fields, power-ups, automation, and actions like move, copy, make template, archive, and share. The bottom of the card shows an activity feed with a comment input field.

Trello Workspaces Recent Starred Templates Create

QA API testing Free

Boards

Members

Workspace settings

Workspace views

Table

Calendar

My boards

testBoard001

testBoard658

testBoard001

testList05

+ Add another list

testCard03

1 0/4

+ Add a card

testCard03

in list testList05

Notifications

Watch

Description

Add a more detailed description...

Attachments

bird.jpg

Added yesterday at 7:01 AM • Comment • Delete • Edit

Remove cover

Add an attachment

check05

0%

item1

item2

item3

item4

Add an item

Activity

Show details

Write a comment...

Power-Ups

Automation

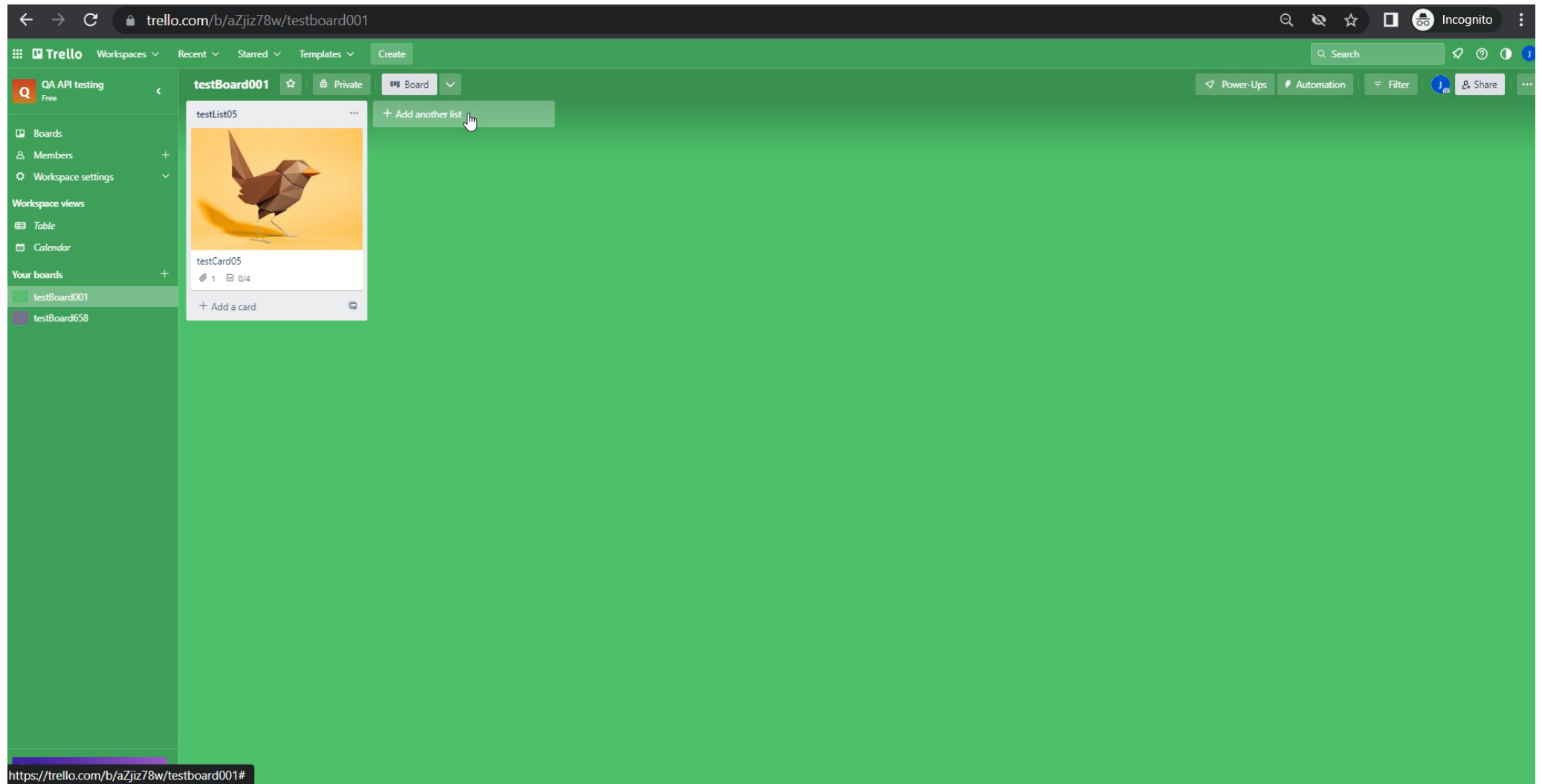
Filter

Share

Incognito

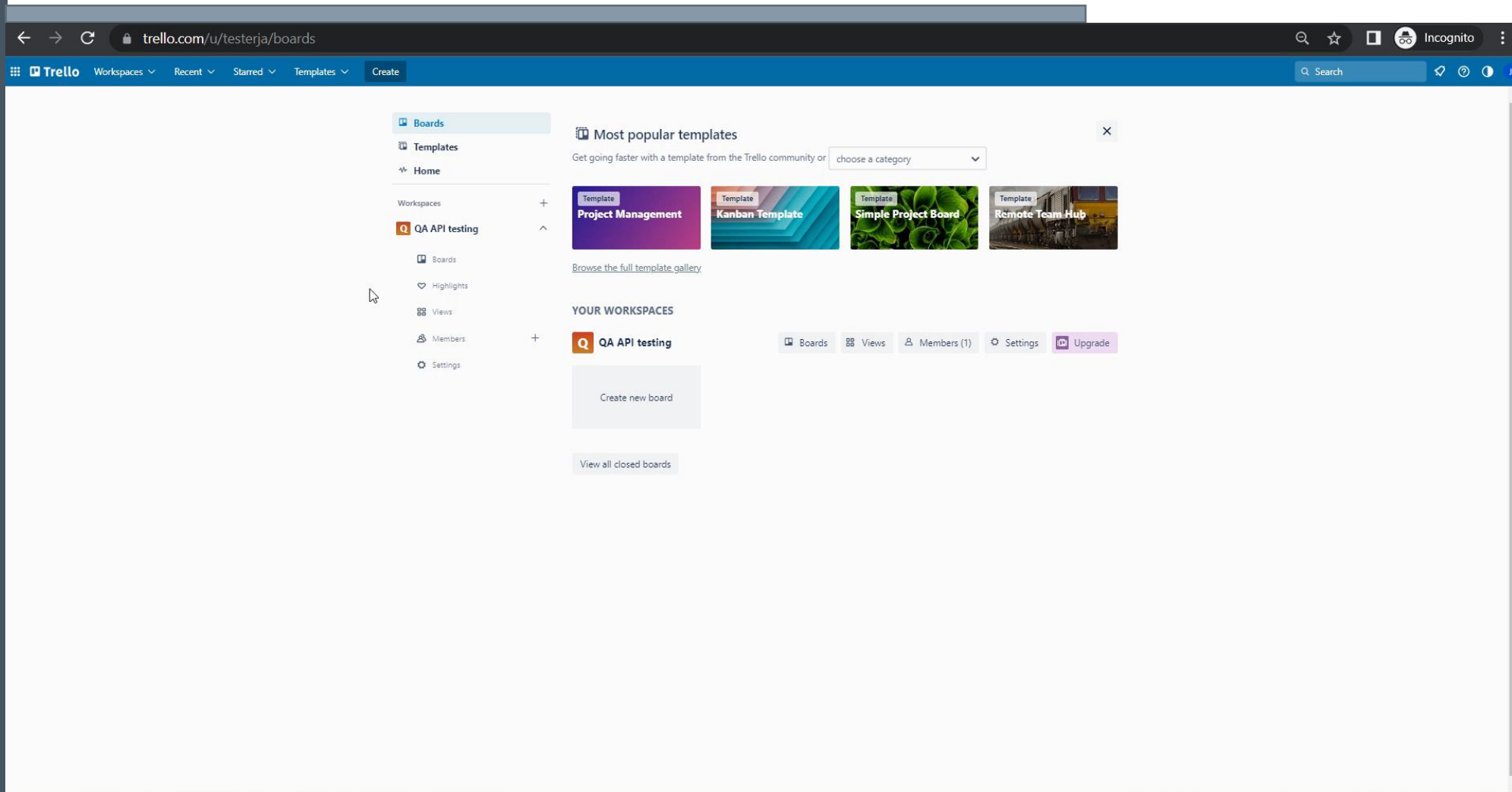


Тест для Trello.com:





API подготовка данных (поэтапно) для Trello.com:





API подготовка данных для Trello.com:

The screenshot displays the Trello web application interface. The browser's address bar shows the URL `trello.com/u/testerja/boards`. The page features a sidebar on the left with navigation links: **Boards**, **Templates**, **Home**, and a section for **Workspaces** containing **QA API testing**. The main content area is divided into several sections:
1. **Most popular templates**: A section with a search bar and four template cards: **Project Management**, **Kanban Template**, **Simple Project Board**, and **Remote Team Hub**.
2. **Recently viewed**: A section showing three board cards labeled **testBoard997**, **testBoard001**, and **testBoard658**.
3. **YOUR WORKSPACES**: A section for the **QA API testing** workspace, showing tabs for **Boards**, **Views**, **Members (1)**, **Settings**, and **Upgrade**. Below these are three board cards: **testBoard001**, **testBoard658**, and **testBoard997**, followed by a **Create new board** button indicating 7 remaining.
4. A **View all closed boards** link is located at the bottom of the workspace section.



API удаление данных Trello.com:

The screenshot shows the Trello web application interface. The browser address bar displays `trello.com/u/testerja/boards`. The sidebar on the left contains navigation links: Boards, Templates, Home, Workspaces, QA API testing (selected), Boards, Highlights, Views, Members, and Settings. The main content area is titled 'boards' and features several sections:

- Most popular templates:** A section with a search bar and four template cards: Project Management, Kanban Template, Simple Project Board, and Remote Team Hub.
- Recently viewed:** A section showing four boards: testBoard572, testBoard997, testBoard001, and testBoard658.
- YOUR WORKSPACES:** A section for the 'QA API testing' workspace, showing four boards: testBoard001, testBoard572, testBoard658, and testBoard997.
- Create new board:** A button labeled 'Create new board 6 remaining' with a plus icon.
- View all closed boards:** A button at the bottom.

Код теста

Project ▾

- pythonProject [pythonUlandAPI] C:\Users\juli
 - github
 - stepic
 - trelo
 - resources
 - bird.jpg
 - donkey.jpg
 - moose.jpg
 - treloSettings.py
 - tests
 - createAllEntitiesTest.py
 - createEntitiesTest.py
 - deleteEntitiesTest.py
 - utils
 - venv library root
 - External Libraries
 - Scratches and Consoles

tailApiTest.py getUnauthToken.py treloSettings.py createAllEntitiesTest.py × createEntitiesTest.py deleteEnti

```
9 class CreateAllEntitiesTest(unittest.TestCase):
10
11     _num = random.randint(100, 999)
12     _color = random.choice(COLORS)
13     print('New object will have number ', _num, 'and color', _color)
14     # Julia Atlasova
15     def test_create_detailed_list(self):
16         url_create_board = f'https://api.trello.com/1/boards/?defaultLists=false' \
17             f'&prefs_background={self._color}&name=testBoard{self._num}{TOKEN}'
18         resp0 = requests.post(url_create_board)
19         self.assertEqual(resp0.status_code, 200)
20         board_id = resp0.json()['id']
21
22         url_create_list = f'https://api.trello.com/1/boards/{board_id}/lists?name=testList{self._num}{TOKEN}'
23         resp1 = requests.post(url_create_list)
24         self.assertEqual(resp1.status_code, 200)
25         list_id = resp1.json()['id']
26
27         url_create_card = f'https://api.trello.com/1/cards/?idList={list_id}&name=testCard{self._num}{TOKEN}'
28         resp2 = requests.post(url_create_card)
29         card_id = resp2.json()['id']
30
31         url_create_attachment = f'https://api.trello.com/1/cards/{card_id}/attachments?{TOKEN}'
32         picture = open('../resources/donkey.jpg', 'rb')
33         file = {'file': picture}
34         resp3 = requests.post(url_create_attachment, files=file)
35         picture.close()
36         self.assertEqual(resp3.status_code, 200)
```

Документация на API Trello.com

← → ↺

developer.atlassian.com/cloud/trello/rest/api-group-boards/

ATLASSIAN Developer

Trello

Guides

Reference

Changelog

REST API

Power-Ups

SCIM API

FAQ

Q

Filter by keyword...

/

▸ Actions

▸ Applications

▸ Batch

▾ Boards

Get Memberships of a Board

Get a Board

Update a Board

Delete a Board

Get a field on a Board

Get Actions of a Board

Get a Card on a Board

Get boardStars on a Board

Get Checklists on a Board

Get Cards on a Board

Get filtered Cards on a Board

Get Custom Fields for Board

Get Labels on a Board

Boards

Get Memberships of a Board

GET /1/boards/{id}/memberships

Get information about the memberships users have to the board.

Request

PATH PARAMETERS

id **REQUIRED**

string

The ID of the board

Pattern: `^[0-9a-fA-F]{24}$`

QUERY PARAMETERS

filter

string

One of `admins`, `all`, `none`, `normal`

Default: `all`

Valid values: `admins`, `all`, `none`, `normal`

activity

boolean

Проблема 3:

создание тестовых данных через UI

Решение:

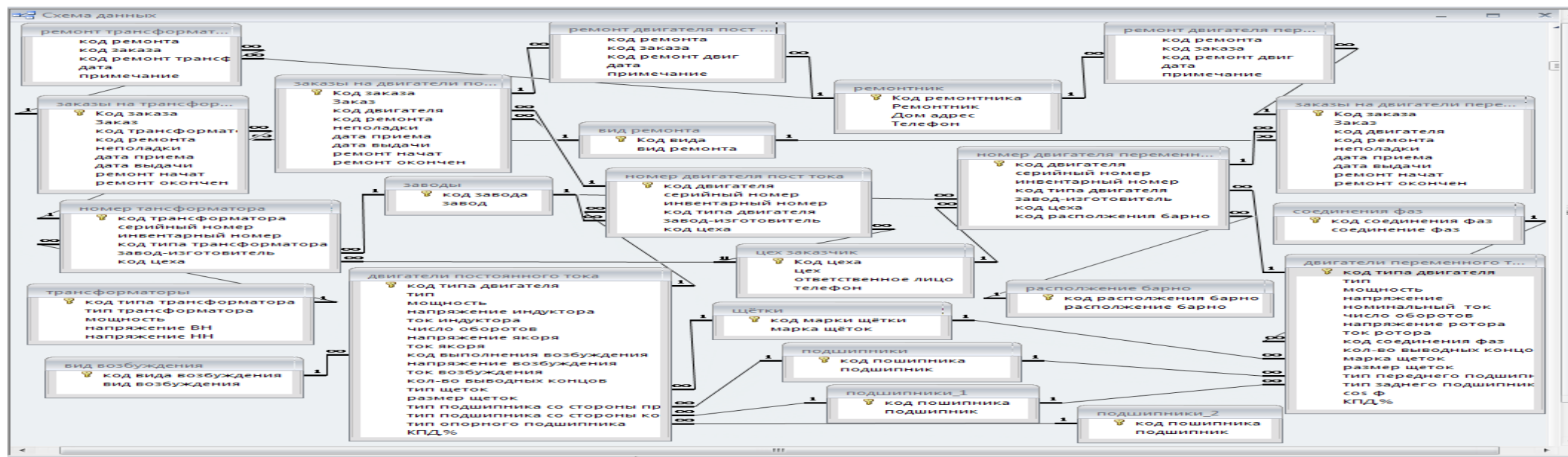
выполняйте всю подготовку,
модификацию, удаление тестовых
данных вызовом API

Советы по организации кода

- Организуйте структуру сервисного кода согласно паттерну PageObject: для сервиса создавайте ServiceObject, который будет аккумулировать в себе все функции, относящиеся к этому API
- Для создания кастомизированного объекта, используйте паттерн Builder – он позволяет задавать только требуемые атрибуты в произвольном порядке.

Альтернатива: работа с базой данных

- ВСТАВКА / ИЗМЕНЕНИЕ ДАННЫХ В ТЕСТЕ
 - ПОИСК ПОДХОДЯЩИХ ДАННЫХ НА ДАМПЕ
 - ОТДЕЛЬНЫЕ СКРИПТЫ ДЛЯ НАКАТЫВАНИЯ ТЕСТОВЫХ ДАННЫХ
- СЛОЖНАЯ СТРУКТУРА БД
 - СЛОЖНО ПОДДЕРЖИВАЕМЫЙ КОД
 - НАРУШЕНИЕ ИНКАПСУЛЯЦИИ



Возможные проблемы при работе с API

- Отсутствие документации на апи
- Атомарность апи
- SOAP сервис (сложнее писать код для вызова)

Бонусы и профит API

- быстрые
- надежные
- легко поддерживать
- Неизменные (API контракт, версионная совместимость)
- дают доступ к функциям, не доступным через UI
- Поддерживаются разработчиками (как backend так и frontend)
- Хорошо задокументированы
- Расширяют возможности инженера автоматизации

ВАШИ ВОПРОСЫ

github:

<https://github.com/YuliaAtlasova/pythonUlandAPI>

Юлия
Атласова



@Julia_Atlasova



JuliaAtlasova@yandex.ru

HEISENBUG