



# Никита Александрович Зенкин.pdf

## Специализация

Тестирование ПО (Software QA Engineer)

## Контакты

Email: fianitnz@gmail.com WhatsApp\Telegram: +7 926 292-39-26 Skype: fianitnz

Репозиторий Git: <https://github.com/fianitnz>

## Навыки

Linux, Windows, Bash, Python, Java (basic), Gradle (basic), Git, Issue tracker, Wireshark, Selenium, WebDriver, KVM, QEMU, Docker (basic), CSV, JSON, API, REST, SOAP, SoapUI (basic), HTML, CSS, DevTools, XML, SQL (soloLearn, sql-ex.ru=6), ORM, English (A1 Elementary)

## Опыт

Создал учебный проект автоматизации тестирования веб панели администратора сервера [Ajenti2](#)

Доступен по адресу: <https://github.com/fianitnz/ajenti2-automation-of-web-interface-testing>

Содержит **Тест план, Тест кейсы, Найденные проблемы**

В проекте использовал такие инструменты как:

### основные:

- selenium.webdriver , pytest , requests , pixelmatch-py , multiprocessing

### вспомогательные:

- PIL.Image , pathlib.Path , io.BytesIO , pytest\_check.check

## Текстовое описание проекта:

Разделил тестируемую страницу на логические секции, заголовок, контент, меню. В которые включил элементы, такие как поля, кнопки, ссылки. Привязав их через локаторы скомпонованные так же в логические секции страницы.

В качестве локаторов использовал в основном XPath. Имена тегов. И CSS локаторы в JavaScript обертках.

Из библиотеки Selenium использовал By, Keys, expected\_conditions, WebDriverWait, Color, ActionChains, разные exceptions.

В связи с отсутствием доступа к псевдоклассам CSS при помощи Selenium, использовал JavaScript.

Написал несколько своих Expected conditions для обработки цвета, трансформации, и выполнения JS в браузере что бы получить значения атрибута псевдокласса CSS.

Использовал pixelmatch-ру для сравнения изображений с эталоном при скриншот тестах. В помощь к скриншот тестам через JS отключаю анимацию путем добавления CSS правил.

Реализовал тестирование логина через API REST при помощи библиотеки requests, в процессе познакомился в работе с ПО SoapUI и протоколом SOAP, попробовал Postman. Для контроля передаваемых\получаемых данных использовал Wireshark.

**В качестве тестового задания написал на Java несколько запросов к API Yandex Speller:**

Доступен по адресу: <https://github.com/fianitnz/test-automation-on-java>

**Использовал библиотеки:**

- JUnit , Unirest , RestAssured , Gson , json

В процессе познакомился с системой сборки Gradle

## Обо мне

За компьютером около 20 лет. На свой первый компьютер заработал в 2002г. Первой ОС была Windows 98, после XP с 2007г использую Linux как основную операционную систему. Windows на втором компьютере держу для САПР. Повторял проекты на Arduino с небольшими правками в коде на C, мини-принтер с Bluetooth 'телеграф', USB ключ симулирующий ввод с клавиатуры для ввода\хранения паролей основанный на проекте Digispark 'attiny85'. С C++ познакомился в 2004 по книге 'для чайников'. По фану изучал Python для себя. Писал всякие вспомогательные скрипты, bash обертки над библиотеками, переписывал их на Python. Например была обертка над библиотекой гугл транслейта, для перевода выделенного текста в любом месте системы по горячим клавишам, как в одну так и в другую сторону в том числе с озвучиванием средствами библиотеки и кешированием озвучивания своими силами. Конвертер тегов для mp3 файлов.

Двое друзей работают в тестировании, давно предлагали сменить профессию. Помогают освоиться, рассказывают о внутренней кухне. Первый работает в тестировании больше одиннадцати лет, занимается ручным тестированием. Второй работает два года, пишет юнит тесты на C\C++ и Python, в 'Код Безопасности'