|  |  |
| --- | --- |
| **Задание** | **Task 2. Selenium Webdriver (yandex market)** |
| Время на выполнение задания | 14 ч |
| Описание задания | Необходимо разработать автотест для следующего тест-кейса:   |  |  |  | | --- | --- | --- | | **#** | **Step** | **Expected Result** | | 1 | Зайти на страницу [https://market.yandex.ru](https://market.yandex.ru/) | Главная страница Yandex.Market открылась | | 2 | Выполнить процесс авторизации | Вход успешно выполнен. Пользователь авторизирован | | 3 | Получить список популярных категорий |  | | 4 | Перейти на случайную категорию из списка | Открыта страница выбранной категории товаров | | 5 | Перейти на главную страницу сайта |  | | 6 | Используя driver.getPageSource(),при помощи регулярных выражений найти названия всех популярных товаров на главной странице.    Все найденные товары записывать в csv файл, который должен создаваться вашим тестом автоматически. | csv файл со списком всех популярных товаров | | 7 | Выполнить log out | Пользователь успешно вылогинился | |
| Требования к выполнению задания | 1. Реализовать PageObjects подход в своём фреймворке. 2. Реализовать получение браузера через Singleton. 3. Проект должен быть организован и запускаться через TestNG Framework 4. Созданный проект должен автоматически собираться без дополнительного вмешательства человека (всё необходимое должно быть в проекте или выкачиваться автоматически).   Простыми словами - чтобы запустить тесты нужно - скачать проект, открыть в IDE и запустить   1. Вынести настройки: имя браузера, url сайта, путь к директории с файлом из шага 6 в отдельный конфигурационный файл (xml|json|ym and etc.) и разработать класс, который будет читать оттуда данные. 2. Тест должен работать в 3х браузерах (IE, Firefox, Chrome) 3. Тест должен поддерживать запуск в браузерах на OS Windows (Win 10) и Linux (Ubuntu) 4. В тесте должен использоваться хотябы 1 раз локатор/селектор каждого из типов:   By.Xpath  By.CSS   1. В тесте должны быть использованы методы click, sendKeys, getText/Text, getAttribute.   Местами это может быть неуместно, но важно продемонстрировать работу с ними.   1. В тесте должен быть по крайней мере 1 клик через [Actions.Click](https://seleniumhq.github.io/selenium/docs/api/java/org/openqa/selenium/interactions/Actions.html#click--). 2. В задании должна быть выполнена работа с ожиданиями: fluentwait, implicitywait, explicity wait 3. Код теста должен соответствовать принятым в компании [Coding Guidelines](https://wiki.a1qa.com/display/CCA/Coding+Guidelines) 4. Код задания необходимо поместить в отдельную ветку (Branch) репозитория студента. Branch name = t2\_selenium\_webdriver. |
| Ожидаемый результат выполнения задания | 1. Создан Pull Request на мердж из ветки с заданием в master 2. Письмо куратору ([v.voytenkov@a1qa.com](mailto:v.voytenkov@a1qa.com)) и техническому эксперту ([a.mishchenko@a1qa.com)](mailto:a.mishchenko@a1qa.com)), содержащее:    * Ссылку на Pull request в Git на мердж кода задания в master    * Время, затраченное на выполнение задания    * На что было затрачено время сверх оценки времени на задание, если было превышение    * Что вызвало наибольшие сложности при выполнении задания    * Какие остались нерешенные проблемы, если остались    * Тема письма = TC428. <studentName (e.g. v.voytenkov)>. Task 2: Selenium Webdriver Results |