**Инструменты для автоматизации тестирования**

**1. Rest-Assured** - DSL для тестирования REST-сервисов, который встраивается в тесты на Java. Это решение появилось более девяти лет назад и стало популярным из-за своей простоты и удобного функционала.

Rest-Assured позволяет десериализовать ответ в требуемый класс. Полученный ответ раскладывается в заранее известную структуру с помощью Jackson. Классы для десериализации выглядят максимально просто — описываются все ожидаемые поля с их типами.

Также стоит заметить, что одно из преимуществ этого решения заключается в его масштабируемости и адаптируемости под сложную логику. А значит, оно наилучшим образом подходит для серьезных проектов, где действительно есть смысл в описании больших объектов, чистом коде и блочной архитектуре, когда есть отдельные куски, отвечающие за подготовку, передачу и получение данных в читаемом виде, а также последующую проверку данных на соответствие каким-то условиям. В маленькие проекты все это тянуть бессмысленно.

2. **Selenium IDE** - расширение к браузеру FireFox, которое позволяет записывать, редактировать, отлаживать и выполнять тесты. Представляет собой простейший инструмент записи/воспроизведения авто тестов и отлично подходит для внедрения “легкой? автоматизации. [Selenium](https://automated-testing.info/tags/webdriver) IDE входит в семейство инструментов автоматизации Selenium. В основе Selenium лежит среда Selenium Core, разработанная на языке [JavaScript](https://automated-testing.info/tags/javascript). Данная среда дает возможность производить манипуляции над браузером и вызывать различные (поддерживаемые браузером) команды для выполнения определенных действий и проверок. Иными словами, можно сказать, что команды, полученные от сценариев теста, выполняются непосредственно средствами браузера. Данный инструмент распространяется бесплатно.

3. **NUnit** — открытая среда юнит-тестирования приложений для [.NET](https://ru.wikipedia.org/wiki/.NET_Framework). Она была портирована с языка Java (библиотека [JUnit](https://ru.wikipedia.org/wiki/JUnit)). Первые версии NUnit были написаны на J#, но затем весь код был переписан на [C#](https://ru.wikipedia.org/wiki/C_Sharp) с использованием таких новшеств .NET, как атрибуты.

Существуют также известные расширения оригинального пакета NUnit, большая часть из них также с открытым исходным кодом. **NUnit.Forms** дополняет NUnit средствами тестирования элементов пользовательского интерфейса Windows Forms. **NUnit.ASP** выполняет ту же задачу для элементов интерфейса в ASP.NET.

4. **Sikuli** — это API позволяющая писать на Jython сценарии автоматизации опираясь на визуальную составляющую любой программы/сайта и т.д. Особенно приятна для автоматизации Flash. **Sikuli** — открытая кроссплатформенная визуальная среда создания сценариев-скриптов, которая ориентирована на программирование графического интерфейса при помощи изображений (скриншотов).В качестве скриптового языка в Sikuli используется Jython, то есть в скрипте при желании можно использовать конструкции из языка [Python](https://ru.wikipedia.org/wiki/Python). В SikuliX появилась возможность использовать для написания скриптов язык Ruby в реализации jRuby. Sikuli доступна для работы в Windows, Mac OS X и Linux.

5. **HP QuickTest Professional** (QTP) — один из ведущих инструментов автоматизации функционального тестирования, является флагманским продуктом компании [HP](https://ru.wikipedia.org/wiki/Hewlett-Packard) в своей линейке. Для разработки автоматизированных тестов QTP использует язык VBScript, и поддерживает следующие технологии: Windows® Presentation Foundation, Web services, Macromedia Flex, Ajax, Delphi, [.NET](https://ru.wikipedia.org/wiki/.NET_Framework), J2EEWeb, Visual Basic, ActiveX, Java, Oracle, SAP Solution, TE, PowerBuilder, Siebel, PeopleSoft, VisualAge, Stingray.

Компания HP рекомендует использование QTP в интеграции с HP Quality Center для установления связи тестов с требованиями, хранения тестов, управления их запуском, формирования отчётов.

В отличие от ряда других продуктов для автоматизации функционального тестирования QTP позволяет контролировать генерируемый текст скрипта в процессе записи действий пользователя, за счёт чего снижается время, необходимое для разработки теста.

В QTP информация обо всех объектах экранного интерфейса сохраняется в специальный репозиторий (Object Repository), что новому пользователю может показаться непрозрачным. Умолчания по выбору существенных свойств каждого типа объектов экранного интерфейса могут быть настроены отдельно, например, окно может определяться заголовком, а столбец таблицы — шириной и порядковым номером в таблице.