

Лекция 10. Selenium. Selenide. Selenoid.

Тестирование. Практическое применение.

Брекелов Всеволод 02.12.2020



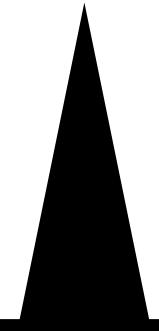
Говорят, что
надо быстро
начать
тестировать
функционал



Но у нас нет компонентных тестов
и юнит тестов

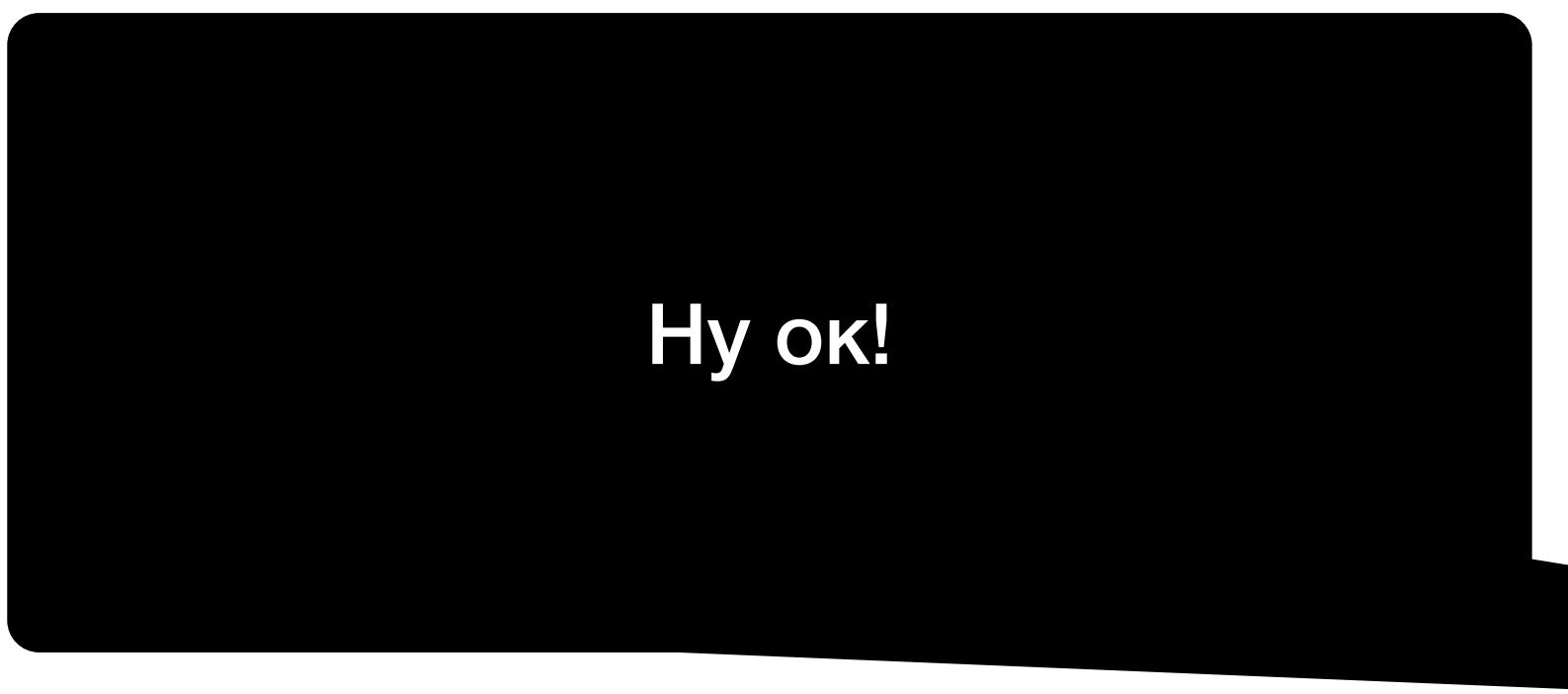


Нет времени, давай делать
быстрее!
Надо проверять
функциональность



Слушай,
Мы делаем сейчас твр и может
многое поменяться, поэтому e2e
тесты быстрее и выгоднее





А что, брать просто Selenium и все?
Что он вообще делает?



Selenium automates browsers. That's it!
What you do with that power is entirely up to you.

Primarily it is for automating web applications for testing purposes, but is certainly not limited to just that.
Boring web-based administration tasks can (and should) also be automated as well.

<https://www.selenium.dev/>

Selenium

WebDriver

IDE

Selenium Grid

Selenium

WebDriver

IDE

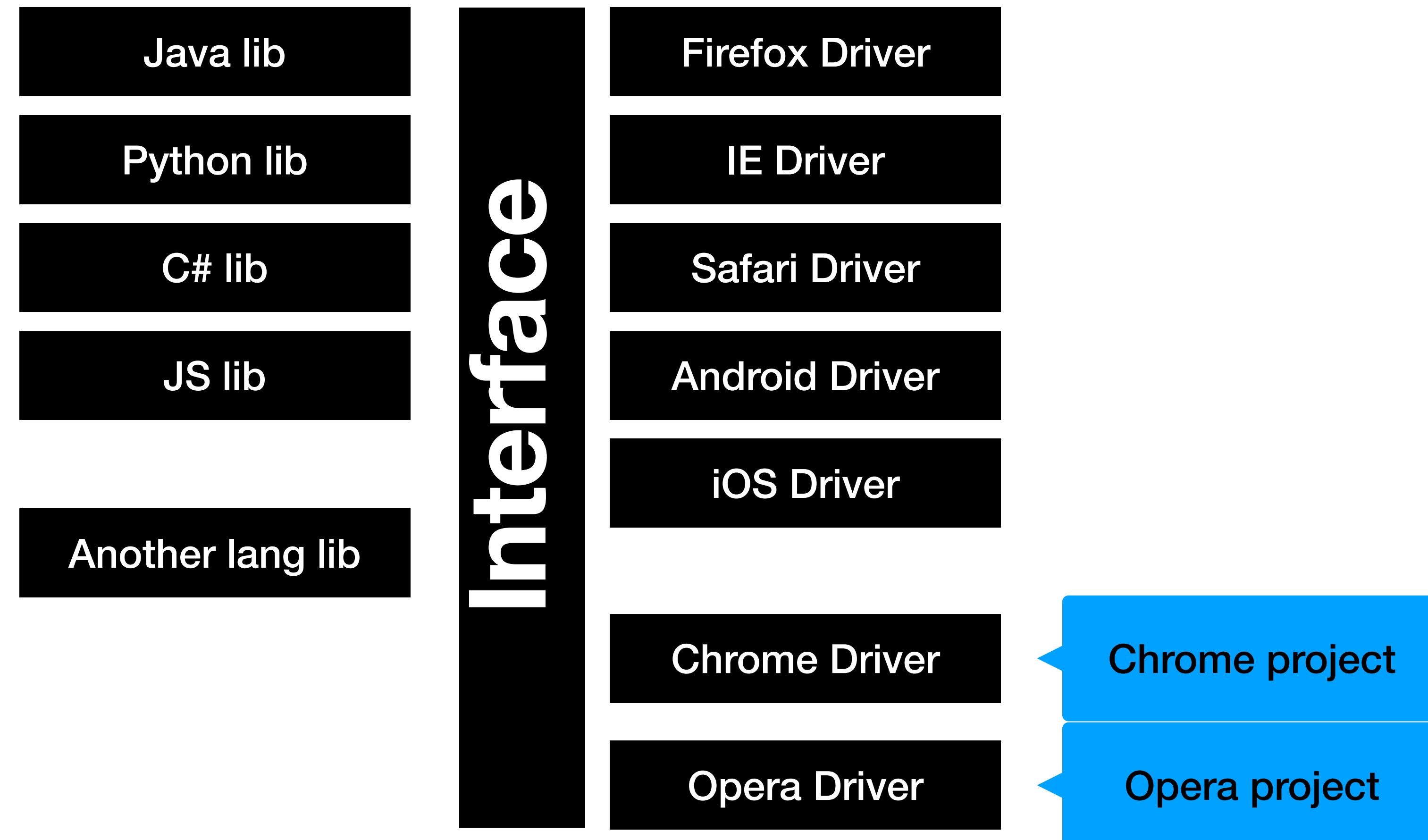
Selenium Grid

Control Browser

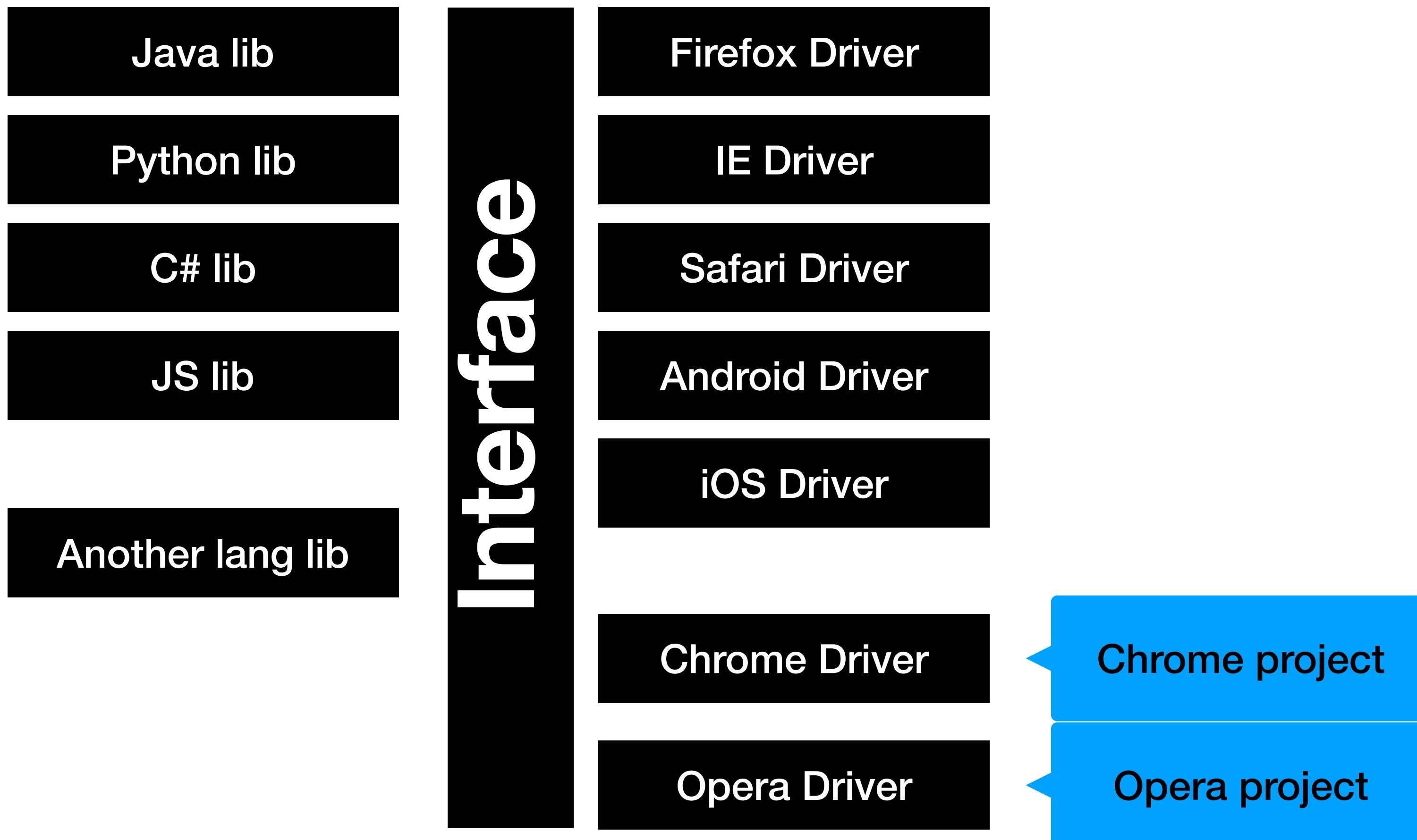
<https://w3c.github.io/webdriver/>

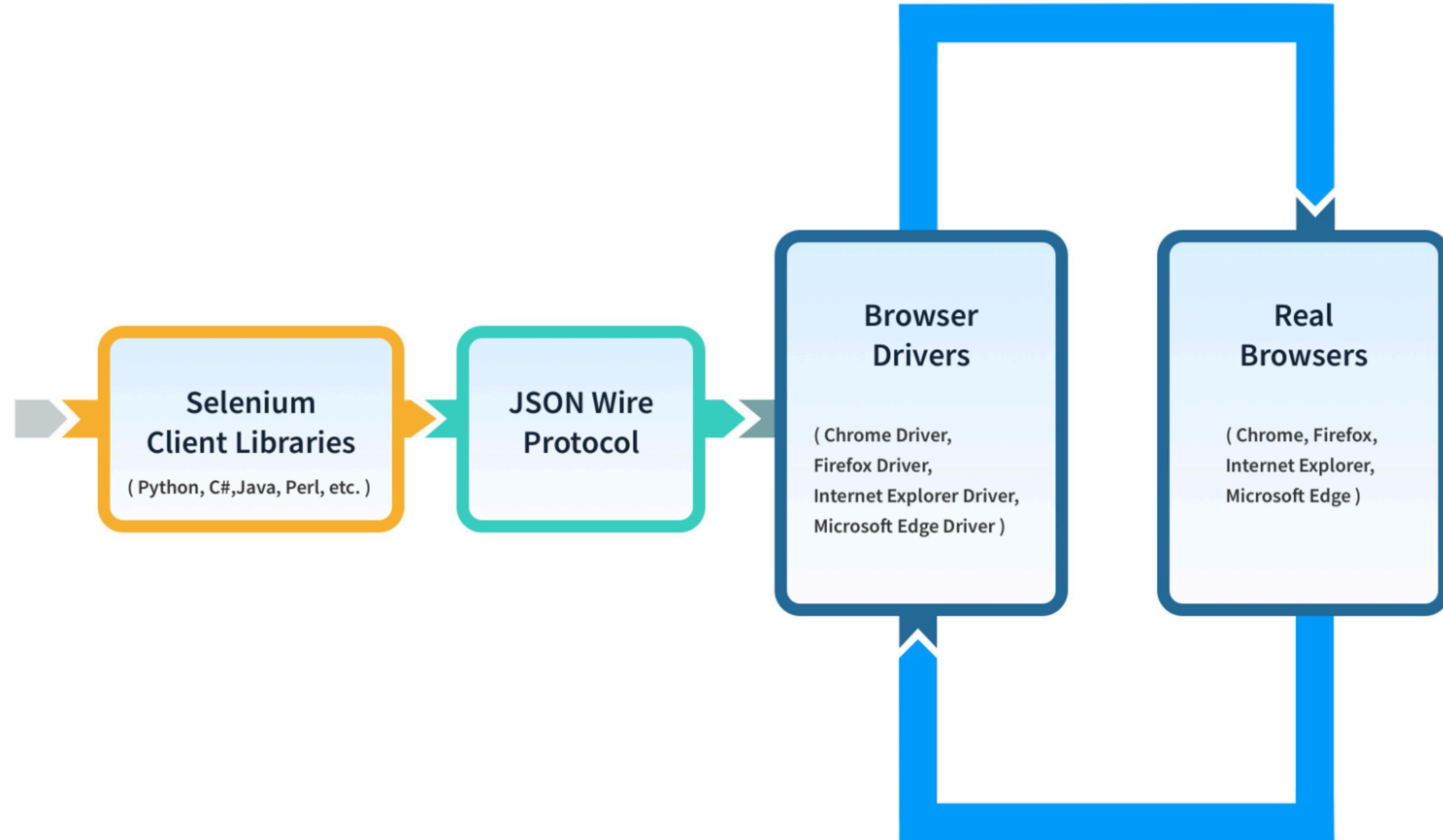
Parallel tests

Selenium



WebDrivers





Selenium

The screenshot shows the Stack Overflow Tags page for the 'selenium' tag. The left sidebar includes links for Home, PUBLIC, Stack Overflow, Tags (which is selected), Users, FIND A JOB, Jobs, Companies, TEAMS, What's this?, and a Free 30 Day Trial offer. The main content area has a search bar with 'selenium' typed in, a 'Show all tag synonyms' link, and sorting options for Popular, Name, and New. The 'selenium' card shows it's a popular open-source tool for automating web browsers, with 81194 questions asked today, 49 asked today, 290 this week, and a red box highlighting the weekly stats. Other cards include 'selenium.webdriver', 'selenium-chromedriver', 'selenium-ide', 'selenium-grid', 'selenium-rc', 'selenium-firefoxdriver', and 'rselenium'.

A tag is a keyword or label that categorizes your question with other, similar questions. Using the right tags makes it easier for others to find and answer your question.

[Show all tag synonyms](#)

Popular Name New

selenium

a popular open-source tool for automating web browsers. When using this tag, also include other tags for the specific components you are using,...

81194 49 asked today, 290 this week

selenium.webdriver

Selenium WebDriver provides the WebDriver API for controlling browsers in different programming languages ("language bindings"). When using th...

41633 23 asked today, 121 this week

selenium-chromedriver

Developed in collaboration with the Chromium team, ChromeDriver is a standalone server which implements WebDriver's wire protocol.

9707 6 asked today, 54 this week

selenium-ide

The Selenium-IDE (Integrated Development Environment) is the tool you use to develop your Selenium test cases. It's an easy-to-use Firefox plu...

2081 9 asked this month, 146 this year

selenium-grid

Selenium-Grid allows you run your tests on different machines against different browsers in parallel. That is, running multiple tests at the same ti...

1663 15 asked this month, 193 this year

selenium-rc

Selenium Remote Control - Answer to UI level functional test Web automation

1382 questions 7 asked this year

selenium-firefoxdriver

a WebDriver implementation that controls the Firefox browser using a Firefox plugin.

858 14 asked this month, 130 this year

rselenium

to make it easy to connect to a Selenium Server/ Remote Selenium Server from within R. RSelenium provides R bindings for the Selenium...

640 18 asked this month, 150 this year

Cypress

The screenshot shows the Stack Overflow 'Tags' page for the 'cypress' tag. The left sidebar is visible, showing links for Home, PUBLIC, Stack Overflow, Tags (which is selected), Users, FIND A JOB, Jobs, Companies, TEAMS, What's this?, and a Free 30 Day Trial button. The main content area has a title 'Tags' and a descriptive paragraph: 'A tag is a keyword or label that categorizes your question with other, similar questions. Using the right tags makes it easier for others to find and answer your question.' Below this is a search bar containing 'cypress'. Three cards are displayed below the search bar:

- cypress**: Description: 'Cypress, an all-in-one testing framework and assertion library, complete with automatic waits, request mocking, and request stubbing'. Metrics: 3407 questions, 6 asked today, 56 this week.
- cypressio**: Description: ' '. Metrics: 22 questions, 9 asked this week, 20 this month.
- cypress-cucumber-preprocessor**: Description: ' '. Metrics: 6 questions.

A red box highlights the '6 asked today, 56 this week' text under the first card.

Playwright

The screenshot shows the Stack Overflow Tags page. The left sidebar has links for Home, PUBLIC, Stack Overflow, Tags (which is selected), Users, FIND A JOB, Jobs, Companies, TEAMS, What's this?, and a Free 30 Day Trial offer. The main content area has a search bar with "Search..." and a magnifying glass icon, followed by the title "Tags". Below the title is a definition: "A tag is a keyword or label that categorizes your question with other, similar questions. Using the right tags makes it easier for others to find and answer your question." A blue link "Show all tag synonyms" is present. A search bar contains the query "playwright". Three cards are displayed:

- playwright**: "a Node library to automate the Chromium, WebKit and Firefox browsers. Playwright is similar to Puppeteer, but with cross-browser...". It shows statistics: 109 questions and 5 asked this week, 19 this month.
- playwright-python**: "a web automation library comparable to Selenium. This tag refers to the Python port of the library.". It shows 9 questions.
- playwright-sharp**: "1 question".

Puppeteer

The screenshot shows the Stack Overflow Tags page for the 'puppeteer' tag. The left sidebar is visible with links like Home, PUBLIC, Stack Overflow, Tags (which is selected), Users, FIND A JOB, Jobs, Companies, TEAMS, What's this?, and Free 30 Day Trial. The main content area has a search bar with 'Search...' and a results section for 'puppeteer'. The results are as follows:

Tag	Description	Questions	Asked Today	Asked This Week
puppeteer	a Node.js library which provides a high-level API to control headless Chrome or Chromium over the DevTools Protocol. It can also be configured to...	4726	5	asked today, 48 this week
puppeteer-sharp	A .NET port of the official Node.JS Puppeteer API.	118	5	asked this month, 65 this year
jest-puppeteer		81		questions
puppeteer-cluster	puppeteer-cluster manages a pool of headless browsers via puppeteer. This is useful to crawl multiple pages in parallel or to keep a pool of open...	39		22 asked this year
aws-lambda-puppeteer		6		5 asked this year

Webdriver io

The screenshot shows the Stack Overflow website's 'Tags' page for the tag 'webdriverio'. The left sidebar has a 'Tags' section highlighted. The main content area shows a search bar with 'webdriverio' and two cards: one for 'webdriver-io' and one for 'webdriver-io-v4'. A red box highlights the 'webdriverio' search term and the 'webdriver-io' card's statistics.

stack overflow Products Search... Search...

Home PUBLIC Stack Overflow Tags Show all tag synonyms

Users FIND A JOB Jobs Companies

TEAMS What's this? Free 30 Day Trial

Tags

A tag is a keyword or label that categorizes your question with other, similar questions. Using the right tags makes it easier for others to find and answer your question.

[Show all tag synonyms](#)

webdriver-io
a JavaScript library for controlling browsers with Selenium server via the WebDriver API.
1167 questions 5 asked this week, 25 this month

webdriver-io-v4
Use tag only for questions particular to problems with version 4 of WebdriverIO. Always add the general [webdriver-io] tag to your question.
23 questions 12 asked this year

Selenium

И производные

Selenium setup

Set deps

```
<dependencies>
    <!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java -->
    <dependency>
        <groupId>org.seleniumhq.selenium</groupId>
        <artifactId>selenium-java</artifactId>
        <version>3.14.0</version>
    </dependency>

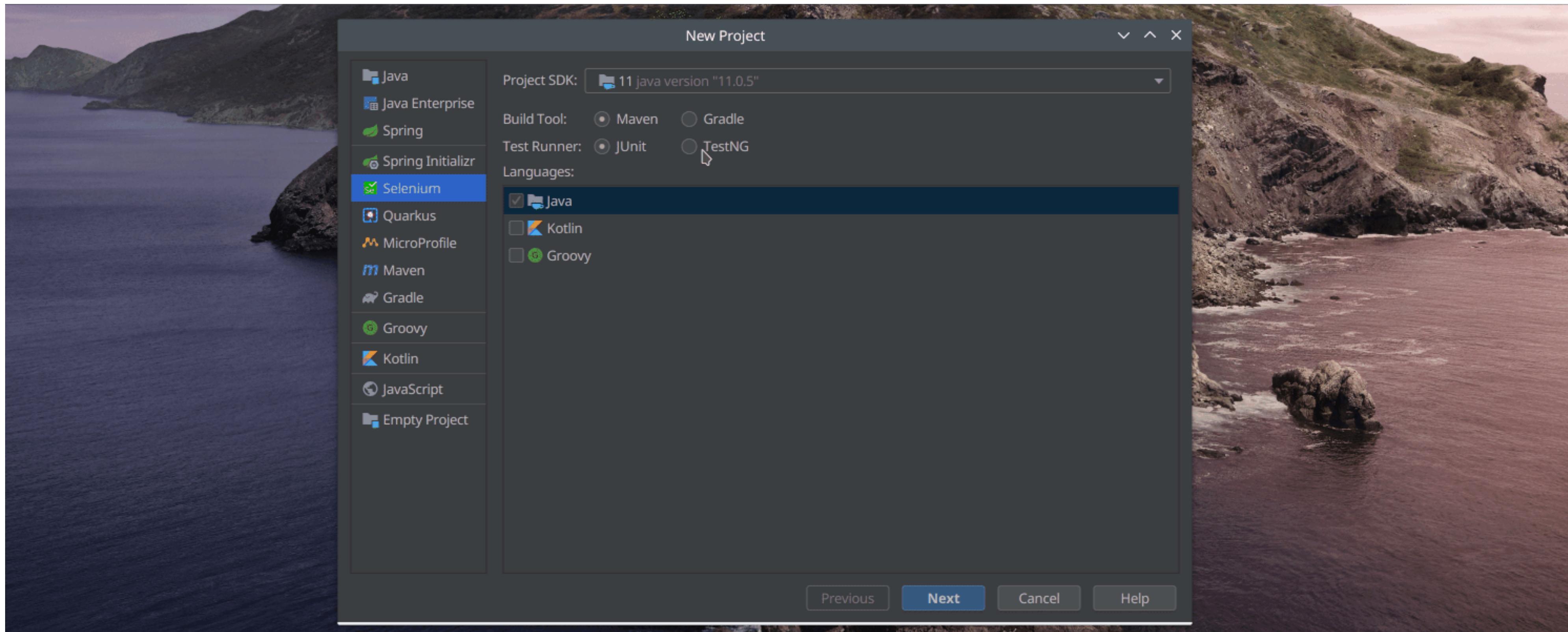
    <!-- https://mvnrepository.com/artifact/junit/junit -->
    <dependency>
        <groupId>junit</groupId>
        <artifactId>junit</artifactId>
        <version>4.12</version>
        <scope>test</scope>
    </dependency>
</dependencies>
```

Download driver

Use driver

```
WebDriver driver = new ChromeDriver();
System.setProperty("webdriver.chrome.driver", "/usr/bin/chromedriver");
```

Selenium setup



<https://blog.jetbrains.com/idea/2020/03/intellij-idea-2020-1-selenium-support/>

Ok! Засетапил, написал много
тестов.
А как правильно писать? =)





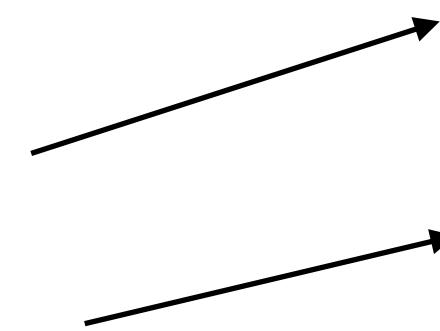
Ну все так же ООП, и прочее как и с
обычным кодом.
Ну есть еще PageObject



PageObject

```
public class ProfilePage {  
    /**  
     * Конструктор класса, занимающийся инициализацией полей класса  
     */  
    public WebDriver driver;  
    public ProfilePage(WebDriver driver) {  
        PageFactory.initElements(driver, this);  
        this.driver = driver; }  
    /**  
     * определение локатора меню пользователя  
     */  
    @FindBy(xpath = "//*[contains(@class, 'account__name_hasAccentLetter')])")  
    private WebElement userMenu;  
    /**  
     * определение локатора кнопки выхода из аккаунта  
     */  
    @FindBy(xpath = "//*[contains(@class, 'menu-item_action_exit menu__item menu__item_type_link')])")  
    private WebElement logoutBtn;  
    /**  
     * метод для получения имени пользователя из меню пользователя  
     */  
    public String getUserName() {  
        String userName = userMenu.getText();  
        return userName; }  
    /**  
     * метод для нажатия кнопки меню пользователя  
     */  
    public void entryMenu() {  
        userMenu.click(); }  
    /**  
     * метод для нажатия кнопки выхода из аккаунта  
     */  
    public void userLogout() {  
        logoutBtn.click(); } }
```

Selectors



Tests

```
@Test  
public void loginTest() {  
    //значение login/password берутся из файла настроек по аналогии с chromedriver  
    //и loginpage  
    //вводим логин  
    loginPage.inputLogin(ConfProperties.getProperty("login"));  
    //нажимаем кнопку входа  
    loginPage.clickLoginBtn();  
    //вводим пароль  
    loginPage.inputPasswd(ConfProperties.getProperty("password"));  
    //нажимаем кнопку входа  
    loginPage.clickLoginBtn();  
    //получаем отображаемый логин  
    String user = profilePage.getUserName();  
    //и сравниваем его с логином из файла настроек  
    Assert.assertEquals(ConfProperties.getProperty("login"), user); }
```

А можно скриншоты?
И тесты часто падают, надо писать
везде ожидания...

Ну и драйвера скачивать это такое
себе





WHAT IS Selenide?

Selenide is a framework for test automation powered by Selenium WebDriver that brings the following advantages:

- ✓ Concise fluent API for tests
- ✓ Ajax support for stable tests
- ✓ Powerful selectors
- ✓ Simple configuration

You don't need to think how to shutdown browser, handle timeouts and StaleElement Exceptions or search for relevant log lines, debugging your tests.

Just focus on your business logic and let Selenide do the rest!

Selenide setup

Set deps

```
<dependency>
  <groupId>com.codeborne</groupId>
  <artifactId>selenide</artifactId>
  <version>5.16.2</version>
  <scope>test</scope>
</dependency>
```

Write test

```
@Test
public void userCanLoginByUsername() {
    open("/login");
    $(By.name("user.name")).setValue("johny");
    $("#submit").click();
    $(".loading_progress").should(disappear); // Waits until element disappears
    $("#username").shouldHave(text("Hello, Johny!")); // Waits until element gets text
}
```

build/reports/tests

ООО! А когда тесты фейлятся —
сам сохраняет скриншот!



Или в ручную:

```
screenshot("my_file_name");
```

<https://ru.selenide.org/documentation.html>

Ядро библиотеки Selenide. Основные методы - это `open`, `$` и `$$`:

- [open\(URL\)](#)
- [\\$\(String cssSelector\)](#) - возвращает объект типа `SelenideElement`, который представляет первый найденный по CSS селектору элемент на странице
- [\\$\(By\)](#) - возвращает "первый `SelenideElement`" по локатору типа `By`
- [\\$\\$\(String cssSelector\)](#) - возвращает объект типа `ElementsCollection`, который представляет коллекцию всех элементов найденных по CSS селектру
- [\\$\\$\(By\)](#) - возвращает "коллекцию элементов" по локатору типа `By`

com.codeborne.selenide.SelenideElement [[src](#)] [[javadoc](#)]

Класс `SelenideElement` - описывает элемент найденный на странице. Его объект можно например получить с помощью команды `$`. В классе описаны следующие полезные методы.

Методы поиска внутренних элементов

- `find(String cssSelector) / $(String cssSelector)`
- `find(By) / $(By)`
- `findAll(String cssSelector) / $$(String cssSelector)`
- `findAll(By) / $$(By)`

Здесь `$` и `$$` просто лаконичные "алиасы" (синонимы) для соответствующих команд.

Таким образом, можно пошагово уточнять - какой внутренний элемент необходимо получить внутри внешнего элемента, строя цепочку последовательных вызовов, например:

```
$("#header").find("#menu").findAll(".item")
```

Методы-проверки состояния элемента - assertions

- `should(Condition) / shouldBe(Condition) / shouldHave(Condition)`
- `shouldNot(Condition) / shouldNotBe(Condition) / shouldNotHave(Condition)`

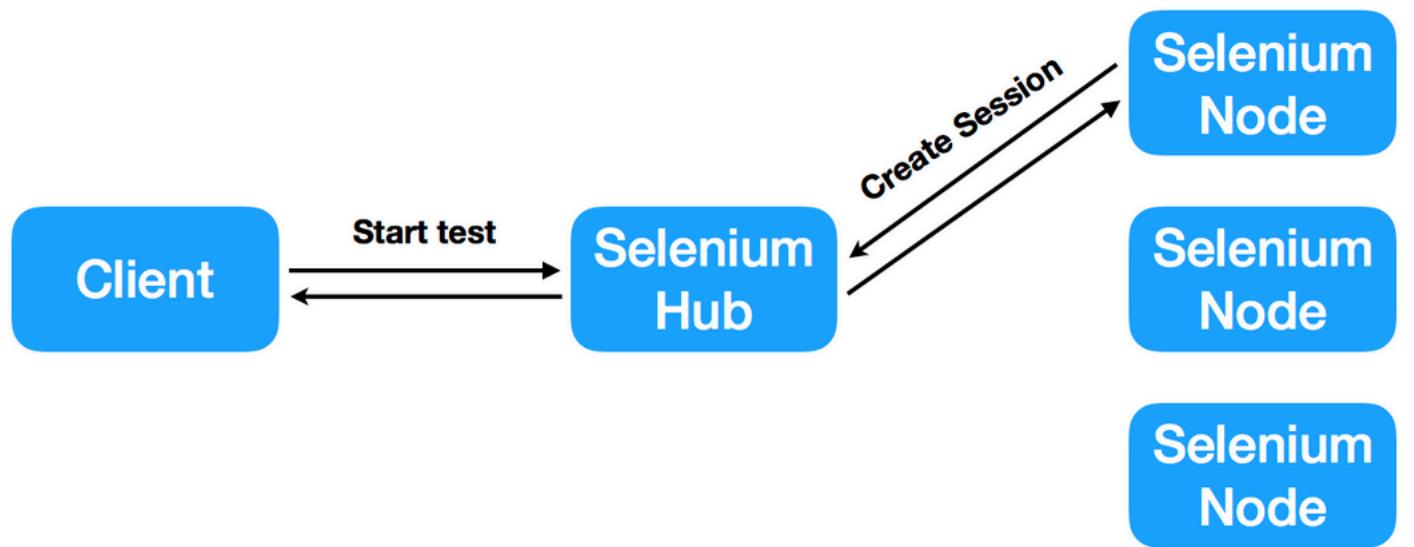
Рекомендуется выбирать такой метод, чтобы строка кода легко воспринималась, как обычная фраза, например:

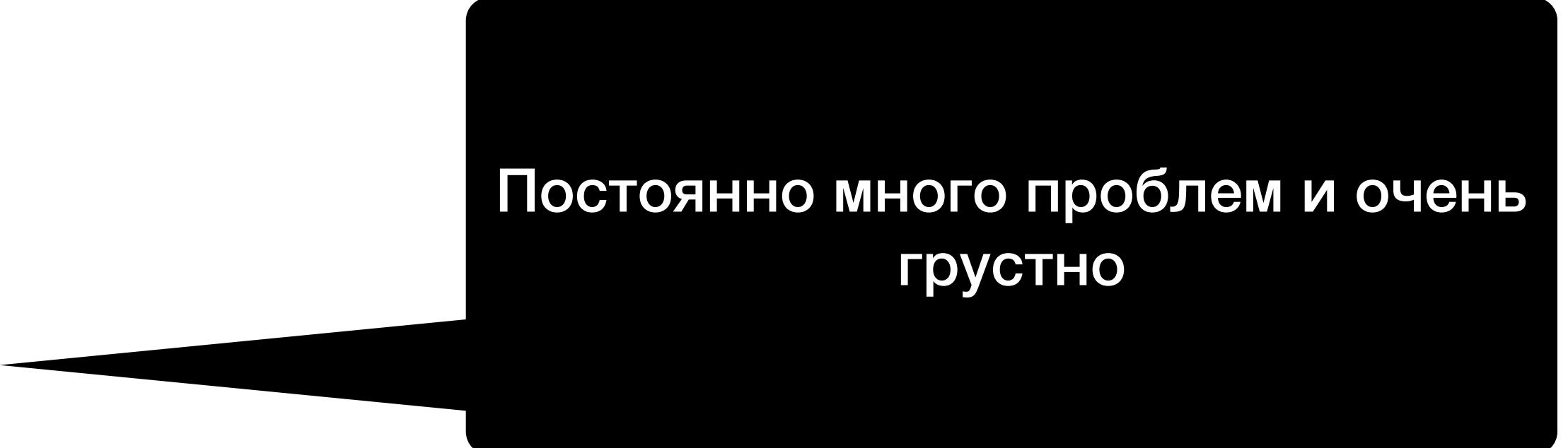
```
$("input").should(exist);
$("input").shouldBe(visible);
$("input").shouldHave(exactText("Some text"));
```

Класс!
А как запускать на разные
браузеры и параллельно?



Selenium Grid





Постоянно много проблем и очень
грустно

Issues

- Непредсказуемое поведение
- Отсутствие поддержки большого количества нод
- Масштабируемость
- Отказоустойчивость
- Невозможно обновить/перезапустить грид без даунтайма

Solutions

Go Grid Router

<https://github.com/aerokube/ggr>

Selenoid

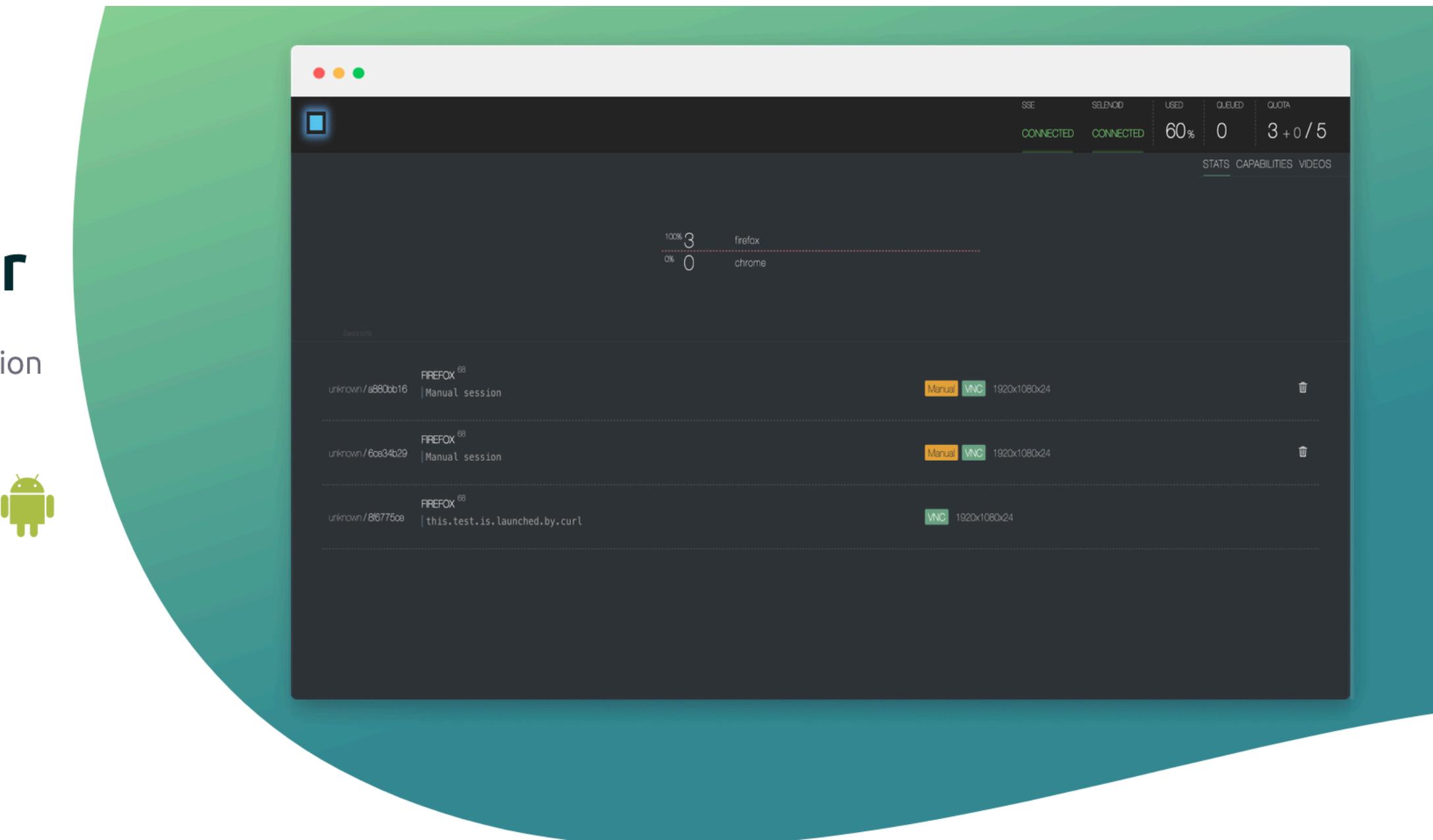
<https://github.com/aerokube/selenoid>



Selenoid

Selenium in Docker

An open-source automated testing solution



Lightning fast

Times faster than
Selenium Grid



Easy installation

One binary does all
the job in minutes



Modern UI

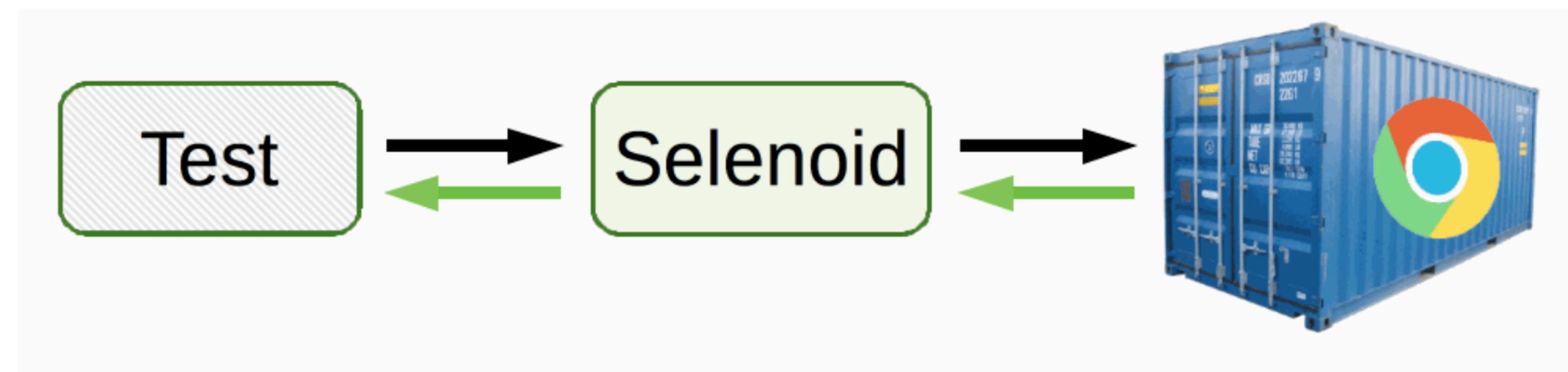
Shows running browser
screen and logs



Cluster ready

Works like a charm
in big clusters

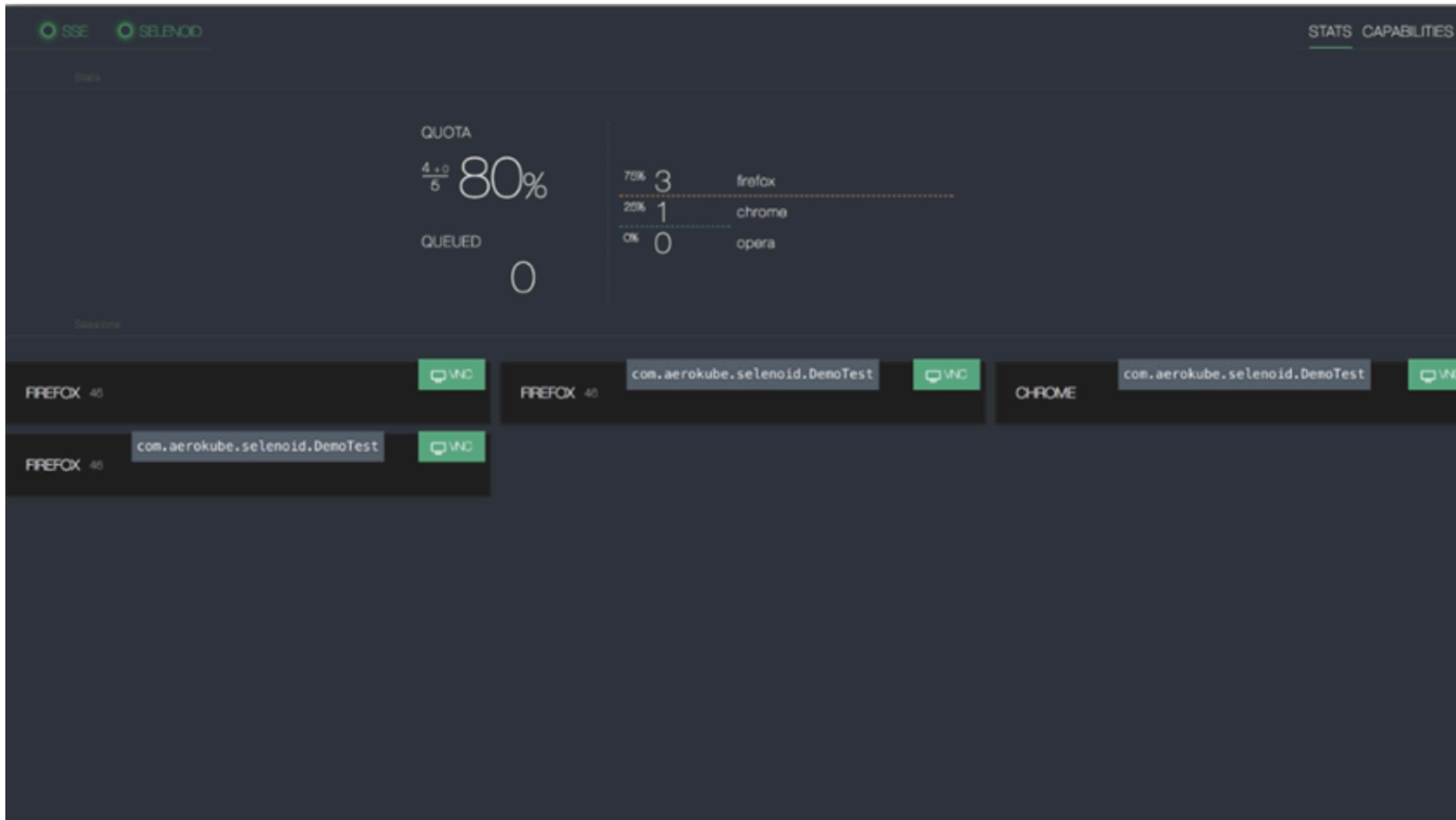
How does it work



Features

- Единая среда для параллельного запуска автотестов
- Изолированное окружение: каждый браузер запускается в отдельном контейнере, что позволяет полностью изолировать окружение нашего браузера
- Масштабируемость: окружение никак не влияет на качественное и непрерывное проведение тестов
- Потребление и утилизация ресурсов: Selenoid позволяет поддерживать высокую нагрузку без дополнительных ресурсозатрат; кроме того, он утилизирует все неактивные контейнеры после завершения самой его сессии, тем самым постоянно поддерживая нужное количество свободной памяти
- Установка: не занимает много времени и осуществляется, по сути, при помощи одной команды
- Одновременная поддержка нескольких версий одного браузера: данная опция доступна только у Selenoid, для этого необходимо создать несколько контейнеров с необходимыми браузерами
- Фокус: операционная система работает таким образом, что в фокусе может быть только одно окно. При запуске нескольких браузеров на одной машине, окна могут начать конкурировать за фокус. У Selenoid такой проблемы нет, поскольку каждый тест запускается в отдельном контейнере
- Пользовательский интерфейс и логи: Selenoid позволяет быстро получить доступ к имеющимся журналам. Помимо этого есть возможность интеграции с [ELK](#) стэком для более быстрого сбора и анализа текущих файлов.

Selenoid UI



Setup

```
@BeforeClass
public static void setUp() {
    //Url удалённого веб драйвера
    Configuration.remote = "http://10.0.75.1:4444/wd/hub";
    //Определяем какой браузер будем использовать
    Configuration.browser = "chrome";
    //Размер окна браузера
    Configuration.browserSize = "1920x1080";
    //Создаём объект класса DesiredCapabilities, используется как настройка вашей конфигурации с помощью пары ключ-значение
    DesiredCapabilities capabilities = new DesiredCapabilities();
    //Включить поддержку отображения экрана браузера во время выполнения теста
    capabilities.setCapability(capabilityName: "enableVNC", value: true);
    //Включение записи видео в процессе выполнения тестов
    capabilities.setCapability(capabilityName: "enableVideo", value: true);
    //Переопределяем Browser capabilities
    Configuration.browserCapabilities = capabilities;
```

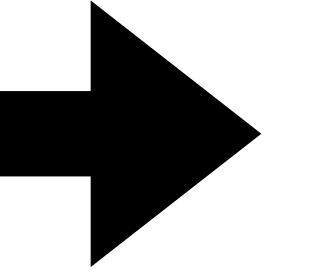


Bax!

Selenide + Selenoid

Альтернативы Selenide?

HtmlElements



Atlas

<https://github.com/qameta/atlas>



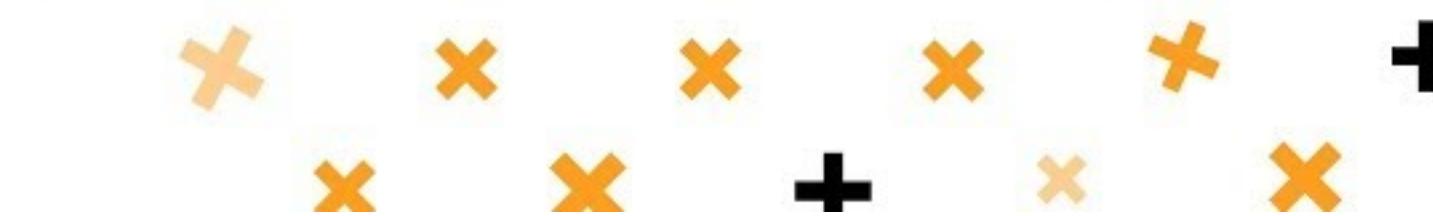
HEISENBUG

// 2018 Piter

Юрий Калинин

Яндекс

Atlas – ваш новый
путеводитель
по PageObject



```

public class WebSiteTest {

    private Atlas atlas;
    private WebDriver driver;

    @Before
    public void startDriver() {
        WebDriverManager.chromedriver().setup();
        driver = new ChromeDriver();
        atlas = new Atlas(new WebDriverConfiguration(driver, "https://github.com"));
    }

    @Ignore
    @Test
    public void simpleWebSiteTest() {
        onSite().onMainPage().open();
        onSite().onMainPage().header().searchInput().sendKeys("Atlas");
        onSite().onMainPage().header().searchInput().submit();
        onSite().onSearchPage("Junit 5").repositories().waitUntil(hasSize(10));
    }

    @Ignore
    @Test
    public void baseUriWebSiteTest() {
        onSite().onSearchPage("Junit 5").repositories().waitUntil(hasSize(10))
            .should(everyItem(text(containsString("junit"))));
    }
}

/**
 * Point of testing WebSite.
 */
public interface GitHubSite extends WebSite {

    @Page
    MainPage onMainPage();

    @Page(url = "search")
    SearchPage onSearchPage(@Query("q") String value);

    @Page(url = "{profile}/{project}/tree/master/")
    ProjectPage onProjectPage(@Path("profile") String profile, @Path("project") String project);

    @Page
    ContributorsPage onContributorsPage();
}

```

Selenide / Atlas
JUnit 5
Selenoid
Allure



Links

<https://habr.com/ru/post/473454/>

<https://habr.com/ru/post/502292/>

<https://habr.com/ru/company/avito/blog/352208/>

<https://medium.com/@eugenetriuts/how-to-install-selenoid-on-macos-751ee2955c70>