The Comparison:

|  | | | |
| --- | --- | --- | --- |
| **CATEGORIES** | **PROTRACTOR** | **WEBDRIVERIO** | **NIGHT WATCH** |
| **ARCHITECTURE** | Protractor is a wrapper around WebdriverJS (selenium-webdriver node module). | WebdriverIO is custom implementation of W3C webdriver JSON wire protocol. | Night Watch is custom implementation of W3C webdriver JSON wire protocol. |
| **SUPPORTED BROWSERS** | Edge, Chrome, Firefox, Safari, Opera, HtmlUnit, PhantomJs | Edge, Chrome, Firefox, Safari, Opera, HtmlUnit, PhantomJs | Edge, Chrome, Firefox, Safari, Opera, HtmlUnit, PhantomJs |
| **ANGULAR SPECIFIC LOCATOR STRATEGIES** | Supported | Partially supported by adding ‘webdriverjs-angular’ node module. Only ‘WaitForAngular’ has been implemented. | Not supported. But still we should be able to automate angular js application as selenium does. |
| **INBUILT TEST RUNNER** | Available | Available | Available |
| **SUPPORTED TESTING FRAMEWORKS** | Jasmine, Mocha, Cucumber | Jasmine, Mocha, Cucumber | Mocha, inbuilt framework |
| **REPORTING** | Junit Xml Reporter, Protractor Html screenshot reporter, Jasmine spec reporter | Allure reporting, Jasmine Spec reporter, Junit xml reporter, Cucumber reporting | Junit xml reporter, Mocha reporting |
| **PARALLEL EXECUTION** | Supported | Supported | Supported |
| **CLOUD EXECUTION**   * **SAUCE LABS** * **BROWSERSTACK** * **TESTING BOT** | Supported | Supported | Supported |
| **MOBILE SUPPORT (APPIUM)** | Partial support (Supports mobile browsers but does not support native apps) | Android, IOS | Android, IOS |
| **SYNCHRONOUS EXECUTION** | Supported | Supported | Supported |
| **PAGE OBJECT PATTERN** | Supported | Supported | Supported |
| **CONTINUOUS INTEGRATION** | Supported | Supported | Supported |
| **RETRYING FLAKY TESTS** | Possible | Possible | Possible |
| **TYPESCRIPT SUPPORT** | Inbuilt support | Possible when using Codecept.js | Not yet supported |
| **ADDITIONAL FEATURES** | Angular application support | CSS regression testing using ‘webdrivercss’ npm package. | – |

The Summary:

* **Protractor**
  + Advantages
    - Protractor is the only automation tool which has inbuilt support for AngularJS element identification strategies. Better to go with protractor if the application is developed using AngularJS.
    - It has support for different unit test frameworks like (Jasmine, Mocha, Cucumber etc..)
    - Good reporting mechanism (html, xml etc..)
    - Compatible with Continuous integration
    - TypeScript support is available which will help Angular 2 developers to write unit tests very easily.
  + Disadvantages
    - It does not have support to automate mobile Apps.
    - It is implemented as a wrapper to the WebdriverJs. So there is one more layer added in between Selenium server and the Protractor. If there is an issue with WebdriverJs the Protractor team should wait for the WebDriverJs team to fix that issue.
* **WebDriverIO**
  + Advantages
    - WebDriverIO is a custom implementation of W3C webdriver API. This gives it an edge to have full control over implementation rather than depending on WebDriverJS implementation.
    - It is fully extensible and is written to be as flexible and framework agnostic as possible. It can be applied in any context and serves not only the purpose of testing.
    - It has a command line interface ‘wdio’ which makes test configuration as easy and simple as possible so that a non-programmer can configure the setup.
    - It has support for most BDD and TDD test frameworks.
    - It has good support and enthusiastic developer community and end users which gives it an edge over NightwatchJS.
    - It can be used with ‘webdrivercss’ to compare css stylings of an element in the webpage.
  + Disadvantages
    - Since it is a custom implementation, it is also a disadvantage as it deviates from generic syntax which may confuse selenium developers coming from other languages.
    - It can be used for automating AngularJS apps but it is not as customized as Protractor.
* **NightwatchJS**
  + Advantages
    - Similar to WebDriverIO, it is also a custom implementation of W3C webdriver API.
    - As all other JS libraries, it is extensible easily – one can write custom commands, assertions extending it.
    - It has its own testing framework and assertions mechanism which solves the programmer’s headache of choosing a testing framework.
  + Disadvantages
    - It does not have much of choices for unit test frameworks as it has an own testing framework and also support Mocha.
    - Slightly lesser support than WebDriverIO and Protractor.