sakshisingla / **Protractor-Non-Angular-Tests**

Creating test scripts using Protractor for non angular application

陳彥澄 edited this page on 20 Nov 2016 · 6 revisions

Protractor is majorly used for creating tests for angular application. However, support is available for non-angular pages as well.

Protractor requires the following :

1. Jasmine
2. Javascript
3. Selenium
4. Node.js server

Using protractor for non-angular pages

1. Use browser.driver instead of driver
2. Use browser.driver.ignoreSynchronization = true
   Reason: Protractor waits for angular components to load completely on a web-page befor it begins any execution. However, since our pages are non-angular, Protractor keeps waiting for 'angular' to load till the test fails with timeout. So, we need to explicitly tell the Protractor to not to wait for 'angular'

Methods to find elements on web page

- Find elements by Id: `browser.driver.findElement(by.id('elem_id'));`
- Find elements by name: `browser.driver.findElement(by.name('elem_name'));`
- Find elements by xpath: `browser.driver.findElement(by.xpath('elem_xpath'));`
- Find elements by css Selector: `browser.driver.findElement(by.css('cssSelector'));`
- Find elements by tag name: `browser.driver.findElement(by.tagName('tagName'));`
- Find elements by class name: `browser.driver.findElement(by.className('class_name'));`

There is no direct menthod to find an element by text. However, there are methods to find elements by link text:

- Find elements by link text: `browser.driver.findElement(by.linkText('link_text'));`
- Find elements by partial link text: `browser.driver.findElement(by.partialLinkText('partial_link_text'));`

Introducing Waits in execution

- Wait for an element to appear on the page and then perform a click:

```
browser.driver.wait(function() {
     return browser.driver.findElement(by.id('id'))
              .then(function(elem) {
                elem.click();
                return true;
              });
  }, timeout_interval);
```

Separating data from logic

Create separate data files and use them in your code to separate data from logic. This will help to deal with continously changing data in future. Steps to do the same:

- Create a json file(test_data.json) for data and add data to it

▾ Pages 2

Home

Creating test scripts using Protractor for non angular application

Clone this wiki locally