Installation

Thursday, March 22, 2018

10:31 PM

How to do environmental installation

* 1. Install Ruby

<https://rubyinstaller.org/downloads/>

Ruby: 2.3.3

Machine generated alternative text:
Downloads 
Rubylnstallers 
Archives» 
WHICH VERSION TO DOWNLOAD? 
If you don't know what version to install and you 're getting started with 
Ruby. we recomrnend you use Ruby installers. These provide a 
stable language and a extensive list of packages (gems) that are 
cornpatible and updated. 
However, not all packages (gems) are maintained. Some older packages 
may not be compatible with newer versions of Ruby and Rubylnstaller. 
Not sure what version to download? Please read the right column for 
Ruby 2.42-2 (x64) 
B Ruby 2.3S 
Ruby 

Machine generated alternative text:
Instanatön Destnatbn and Tasks 
wd Ruby 2.2.8-94-9-x64fito C:kk 
r to a dffemt 
•vod "y fouer Out cont&ns (e.g. Pies). 
C: W-by21.x€4 
@ Riby executae•s to PATH 
.rb rd .rb„•. Ruby 
Select to enable running your Ruby scripts by double clicking 
cr "rnpty typvng the script name at your shell prompt. This may 
affect existing Ruby installations. 
Recured 2 

* 1. Go back to that website and download

Development kit

For use with Ruby 2.0 to 2.3 (x64 - 64bits only)

* + [DevKit-mingw64-64-4.7.2-20130224-1432-sfx.exe](https://dl.bintray.com/oneclick/rubyinstaller/DevKit-mingw64-64-4.7.2-20130224-1432-sfx.exe)

Go to Downloads folder >and copy to the drive C:\ where is your Ruby install C:\Ruby24-x64

After completed cd to C:\>cd Ruby24-x64 >ruby dk.rb init

C:\>cd Ruby24-x64

C:\Ruby24-x64>ruby dk.rb init

[INFO] found RubyInstaller v2.3.3 at C:/Ruby23-x64

[INFO] found RubyInstaller v2.4.3 at C:/Ruby24-x64

Initialization complete!

C:\Ruby24-x64>ruby dk.rb install

Confirm that you are have that install correctly

C:\Users\Teacher>ruby -v

ruby 2.3.3p222 (2016-11-21 revision 56859) [x64-mingw32]

* + Install cucumber

C:\Ruby24-x64>gem install cucumber

Successfully installed cucumber-3.1.0

Parsing documentation for cucumber-3.1.0

Done installing documentation for cucumber after 2 seconds

1 gem installed

C:\Ruby24-x64>

* 1. Install RubyMine for your machine

<https://www.jetbrains.com/ruby/>

* 1. You will need ChromeDriver (be sure to take the latest chrome driver otherwise it's not gone work!

**For Windows And Mac**

------------------------------

<https://sites.google.com/a/chromium.org/chromedriver/downloads>

Machine generated alternative text:
Automation Pyramid 
PIT 
Integration 
Component Tests 
Unit Tests 
* Developed by Mike Cohn, described in his book Succeeding with Agile 

Cucumber Ruby Selenium API

Cucumber is BDD (all about collaboration)

Machine generated alternative text:
Cucumber Testing Stack 
Features 
-Scenarios 
-Steps 
-Gherkin 
Glue code 
-Step definitions 
Application 
-Ruby 
-Java 
-C# & others 

C:\Users\Teacher\Desktop\repos\practical-sqa>gem install bundler

C:\Users\Teacher\Desktop\repos\practical-sqa>bundle install

* 1. Create first file features

create a first scenario

**Feature:**Contactme

**Scenario:**Leaving a Comment

**Given**InavigatetoContactMepage

**When**IverifytheContactMepageloaded

**Then**Iwritemycomment"**Helloworld!ThisisSeleniumtest.**"

**And**Iinputmyname"**JohnSmith**"

**Then**Iinputmyemailaddress"**my.email@gmail.com**"

**And**Iinputmywebsiteurl"**www.my\_url.com**"

**Then**Ichecksubscribecheckbox

**And**IclickonPostCommentbutton

What Is a Regular Expression, Regexp, or Regex?

* 1. We nee to

From <<https://www.regexbuddy.com/regex.html>>

It is a reference to the current file name. In the file foo.rb, \_\_FILE\_\_ would be interpreted as "foo.rb".

With these files:

# test.rb  
puts \_\_FILE\_\_  
require './dir2/test.rb'

# dir2/test.rb  
puts \_\_FILE\_\_

Running ruby test.rb will output

test.rb  
/full/path/to/dir2/test.rb

From <<https://stackoverflow.com/questions/224379/what-does-file-mean-in-ruby>>

*driver\_path*=***File***.join(***File***.dirname(**\_\_FILE\_\_**),**".."**,**".."**,**"drivers"**,**"chromedriver.exe"**)

puts*driver\_path*

*Given*(**/**^InavigatetoContactMepage$**/**)**do**

**@browser**=***Selenium***::***WebDriver***.for**:chrome**,**driver\_path**:*driver\_path*

**@browser**.manage.timeouts.implicit\_wait=10

**@browser**.manage.timeouts.page\_load=10

What is implicit wait ?

Selenium Web driver provides two types of waits

* + Implicit waits
  + Explicit waits

**Implicit Waits**

Selenium Web Driver has borrowed the idea of implicit waits from Watir. An implicit wait is to tell Web Driver to poll the DOM for a certain amount of time when trying to find an element or elements if they are not immediately available.The default setting is 0. Once set, the implicit wait is set for the life of the Web Driver object instance until it changed again. This means that we can tell Selenium that we would like it to wait for a certain amount of time before throwing an exception that if it cannot find the element on the page.

(Web Driver object instance).implicitly\_wait(seconds) ###Syntax

>>> from selenium import web driver

>>> driver = webdriver.Firefox()

>>> driver.implicitly\_wait(30)

>>> driver.get("<https://www.google.co.in/>")

>>> driver.find\_element\_by\_id(“lst-ib”)

In the above example the implicit waits value is given as 30sec i.e, if web driver able to find the element with in the given span 30sec it immediately returns true else it raises an exception after 30 seconds.

From <<http://www.aptuz.com/blog/selenium-implicit-vs-explicit-waits/>>

Difference Between Implicit, Explicit and Fluent Wait

**Implicit Wait:** During Implicit wait if the Web Driver cannot find it immediately because of its availability, it will keep polling (around 250 milli seconds) the DOM to get the element. If the element is not available within the specified Time an NoSuchElementException will be raised. The default setting is zero. Once we set a time, the Web Driver waits for the period of the WebDriver object instance.

**Explicit Wait:** There can be instance when a particular element takes more than a minute to load. In that case you definitely not like to set a huge time to Implicit wait, as if you do this your browser will going to wait for the same time for every element.

To avoid that situation you can simply put a separate time on the required element only. By following this your browser implicit wait time would be short for every element and it would be large for specific element.

**Fluent Wait:**Let’s say you have an element which sometime appears in just 1 second and some time it takes minutes to appear. In that case it is better to use fluent wait, as this will try to find element again and again until it find it or until the final timer runs out.

**Solutions:**We always get confuse when it comes to using Wait commands, to better understand it we need to remember that there is a difference between several scenarios:

An element not being present at all in the DOM.

An element being present in the DOM but not visible.

An element being present in the DOM but not enabled. (i.e. clickable)

There are pages which get displayed with the **JavaScript**, the elements are already present in the browser **DOM**, but are not visible. The implicit wait only waits for an element to appear in the DOM, so it returns immediately, but when you try to interact with the element you get a **NoSuchElementException**. You could test this hypothesis by writing a helper method that explicit wait for an element to be visible or clickable.

*driver\_path*=***File***.join(***File***.dirname(**\_\_FILE\_\_**),**".."**,**".."**,**"drivers"**,**"chromedriver.exe"**)

puts*driver\_path*

*Given*(**/**^InavigatetoContactMepage$**/**)**do**

**@browser**=***Selenium***::***WebDriver***.for**:chrome**,**driver\_path**:*driver\_path*

**@browser**.manage.timeouts.implicit\_wait=20

**@browser**.manage.timeouts.page\_load=10

*#url=****"http://www.practicalsqa.net/contact-me/"***

*#****@browser****.navigate.tourl*

>

Let’s try to describe what we see in HTML. We see that our element has tag ***a*** class with identifier ***rt*** and also text ***Today***. The only thing we are missing by now is the knowledge of how to write it so that Selenium will understand us.

Here are some rules:

* + XPath sholid be in single or double quotes (We will always use **“”**)
  + We use a relevant XPath (I will explain later what it means) and it is always starts from **//**
  + Before we use any attribute we use **@**
  + We put any expression (description) in **[]**
  + If we use name of the attribute (string) we put it in single quotes **”**

From <<http://www.practicalsqa.net/second-part-of-the-first-lesson/>>

{**xpath**:**"//h1[@class='site-title']/a[text()='PracticalSQA']"**}

**def***page\_loaded*

wait\_for(20){**@browser**.find\_element(***PAGE\_LOADED***)}

**end**

**def***visit*

*path*=**"/contact-me/"**

*url*=***URI***(**"http://www.**#{**$base\_url**}#{*path*}**"**)

**@browser**.navigate.to*url*

**end**

**def***page\_loaded*

wait\_for(20){**@browser**.find\_element(***PAGE\_LOADED***)}

**end**

**def***write\_comment*(*comment*)

*comment\_field*=**@browser**.find\_element(***COMMENT\_FIELD***)

*comment\_field*.send\_keys*comment*

**End**

*And*(**/**^Iinputmyname"([^"]\*)"$**/**)**do**|*name*|

**@contactme\_page**.input\_name(*name*)

**end**

**def***input\_name*(*name*)

*name\_field*=**@browser**.find\_element(***NAME\_FIELD***)

*name\_field*.send\_keys*name*

**end**

*Then*(**/**^Iinputmyemailaddress"([^"]\*)"$**/**)**do**|*email*|

**@contactme\_page**.input\_email(*email*)

**end**

*And*(**/**^Iinputmywebsiteurl"([^"]\*)"$**/**)**do**|*url*|

**@contactme\_page**.input\_url(*url*)

**End**

**def***post\_comm\_button*

*button*=**@browser**.find\_element(***POST\_COMM\_BUTTON***)

*#button.click*

sleep1

**end**

***PAGE\_LOADED***={**xpath**:**"//h1[@class='entry-title'andtext()='ContactMe']"**}

***COMMENT\_FIELD***={**id**:**"comment"**}

***NAME\_FIELD***={**id**:**"author"**}

***EMAIL\_FIELD***={**id**:**"email"**}

***URL\_FIELD***={**id**:**"url"**}

***SUBSCRIBE\_CHECKBOX***={**id**:**"subscribe\_blog"**}

***POST\_COMM\_BUTTON***={**id**:**"submit"**}

*driver\_path* = ***File***.join(***File***.dirname(**\_\_FILE\_\_**), **".."**, **".."**, **"drivers"**,**"chromedriver.exe"**)  
 puts *driver\_path*

*Given*(**/**^I navigate to Contact Me page$**/**) **do  
 @browser** = ***Selenium***::***WebDriver***.for **:chrome**, **driver\_path**: *driver\_path* **@browser**.manage.timeouts.implicit\_wait = 10  
 **@browser**.manage.timeouts.page\_load =10  
 *url* = **"http://www.practicalsqa.net/contact-me/"  
 @browser**.navigate.to *url* puts **"WebDriver has been created"  
end**

*When*(**/**^IverifytheContactMepageloaded$**/**)**do**

**@browser**.find\_element({**xpath**:**"//h1[@class='entry-title'andtext()='ContactMe']"**})

**end**

*Then*(**/**^Iwritemycomment"([^"]\*)"$**/**)**do**|*comment*|

**@browser**.find\_element(**id**:**"comment"**).send\_keys*comment*

**end**

*And*(**/**^Iinputmyname"([^"]\*)"$**/**)**do**|*name*|

**@browser**.find\_element(**id**:**"author"**).send\_keys(*name*)

**end**

*Then*(**/**^Iinputmyemailaddress"([^"]\*)"$**/**)**do**|*emaile*|

**@browser**.find\_element(**id**:**"email"**).send\_keys(*emaile*)

**end**

*And*(**/**^Iinputmywebsiteurl"([^"]\*)"$**/**)**do**|*my\_url*|

**@browser**.find\_element(**id**:**"url"**).send\_keys(*my\_url*)

**end**

*Then*(**/**^Ichecksubscribecheckbox$**/**)**do**

**@browser**.find\_element(**id**:**"subscribe\_blog"**).click

**end**

*And*(**/**^IclickonPostCommentbutton$**/**)**do**

*button*=**@browser**.find\_element(**id**:**"submit"**)

sleep1

**end**