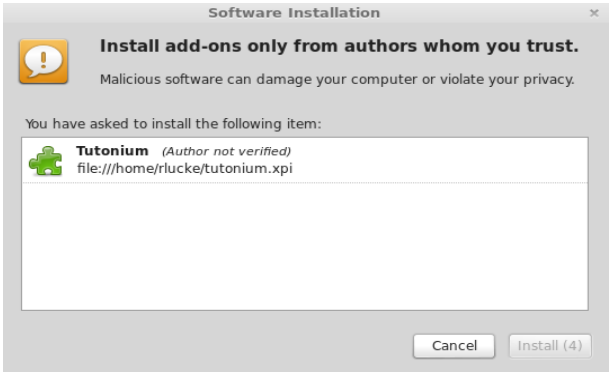
Tutonium 1.0.1

Requirements

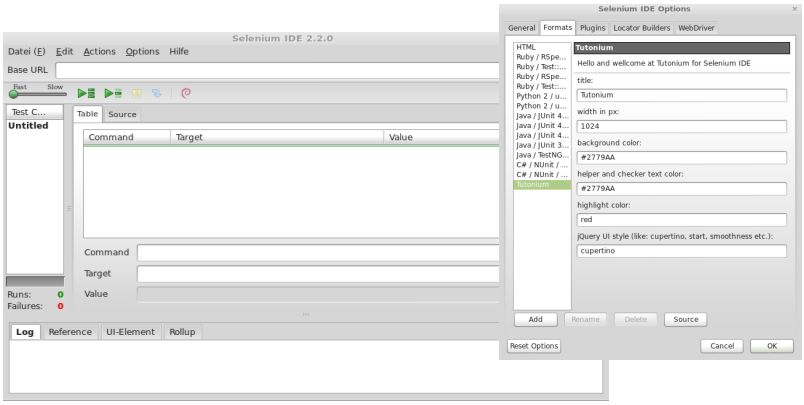
All you need is an up to date version of Firefox (http://www.mozilla.org/de/firefox/) and the Selenium IDE plug-in (http://docs.seleniumhq.org/download/).

Install

Download the Tutonium archive and unpack the tutonium.xpi file. This file is a Firefox add-on installer, you can easily open it with Firefox. Whether you drag and drop or use open with, Firefox should display something like this:



After restarting your Firefox you will find Tutonium in your Selenium IDE.



Usage

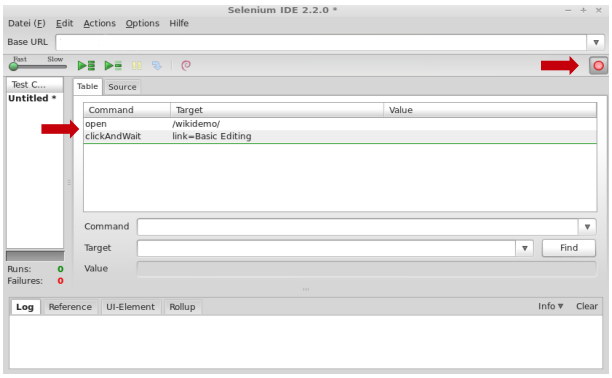
To create a tutorial you need to build test cases. Each case represents a task (tab) in the tutorial.
First you have to describe the task. Use Selenium command storeText and put some text into the value field.

Tutonium gives you the opportunity to check solutions. Commands like verifyElementPresent, verifyTextPresent, verifyText, verifyLocation and verifyPath enables Tutonium's checking function. Furthermore you have the chance to display a help.
The clickAndWait command enables Tutonium's helper function.

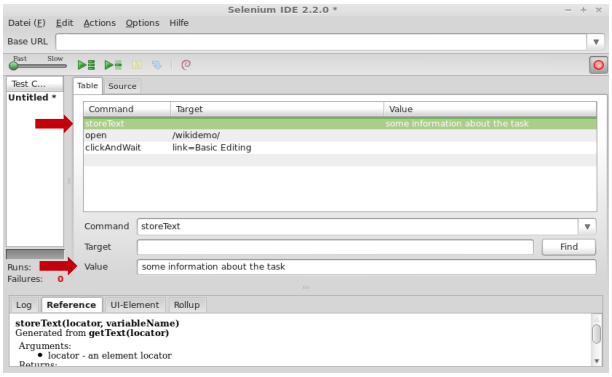
Example

We will now create a small tutorial called 'how to use PmWiki'.
When PmWiki is installed as well as Tutonium we a ready to start.

Editing text and using markups is the key to use PmWiki, but newbies surely do not know anything about markups. Let us show them where to look up. At the left navigation bar you will find 'Basic Editing', click on it while Selenium is active. Selenium records your actions, so it will look like this:



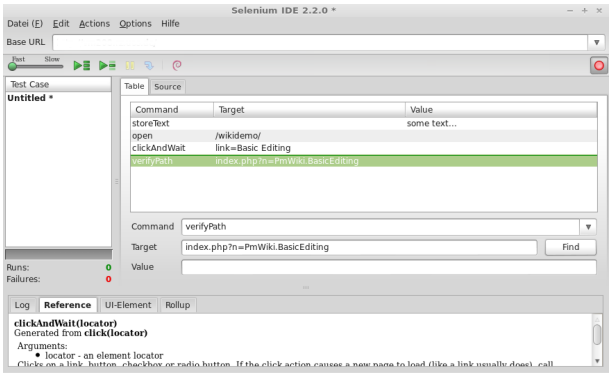
Two commands has been set, open and clickAndWait. Open will tell Tutonium at which page the task starts. clickAndWait will create a helper which will point at our target 'Basic Editing'. Now we have to describe our Task, therefor we have to create a new command called 'storeText'. Its value will be the tasks description.



Maybe we want to edit this task later or use it in another tutorial, anyway it is always smart to store you work. First we just want to store the Test Case. Use 'File -> Save Test Case' or ctrl+s. Choose the file name wise, it will be the title for our task.

Showing our newbie where to look at is fine, but he needs although to check if he did right. Tutonium offers different ways to verify. verifyPath and verifyLocation will use the URL to verify. verifyPath looks for a part of the URL and verifyLocation compares the whole URL. verifyElementPresent and verifyTextPresent search the page for an element or a text.
If you are looking for a text in a special element use verifyText. Whether you insert the command using the context menu or insert it manually all verify commands, but verifyText, need only a target.

In our case it seems smart to use verifyPath, because PmWiki pages have their name in their URL. Insert a new command 'verifyPath' and set it's target to 'index.php?n=PmWiki.BasicEditing'.



Our first task is now ready for testing. By exporting the test suite with the Tutonium formatter, Selenium creates a Tutonium file which contains everything needed for our tutorial. All we need to do now is to upload this file to our server. Same-Origin-Policy forbids cross script actions, therefore the Tutonium tutorial has to be on the same server as our web application.
Adding new tasks is as easy as the first one. Tryout the following commands and explore Tutonium's possibilities.

Commands

command	description	helper	checker	id	name	css	link
clickAndWait	point at an element	x		x	x	x	x
click	point at an element	x		x	x	x	x
verifyLocation	target URL		x	x	x	x	x
verifyPath	target path		x	x	x	x	x
verifyElementPresent	Element that has to be present		x	x	x	x	x
verifyTextPresent	Text that has to be present		x	x	x	x	x
verifyText	Text at a certain position that has to be present		x	x	x	x	x
select	point at a drop down element / selected value in a drop down element	x	x	x	x	x	

Markups

Your task description, set with storeText, escapes any HTML type stings. For instance you like to have a line break and use '
', Tutonium will display those four characters, but no line break. Use the line break markup '\\' instead.

markup	example	description
\\	first line. \\ next line … → first line next line	sets a tag, which causes a line break
''text''	''hello'' → *hello*	sets tag, so the text between will be italic
'''text'''	'''hello''' → **hello**	sets tag, so the text between will be bold
[[http://www.link.org|Link]]	[[http://rlucke.github.io/tutonium/|Tutonium]] → Tutonium	creates an anchor tag <a>, which will open in a new browser tab.
%color%text%%	%red%colorful text%% normal text → colorful text normal text	sets a tag with css color, color names work as well as hexadecimal code
->	-> creates →	create an arrow without using [rarr]
<-	<- creates ←	create an arrow without using [larr]
[character]	[alpha] → α	creates additional characters by using the HTML entity names

Thanks

We thank the PmWiki team (http://www.pmwiki.org/) and especially the Selenium contributors (http://www.seleniumhq.org/) for their superb software.