**DOCUMENTATION FOR TODAY’S DISCOUNT**

* **Configurations**
* CasperJs
* Nodejs
* Azure Storage

**CASPERJS**

**Installation Steps for CasperJs:**

1. Download that link

<https://github.com/n1k0/casperjs/zipball/1.1-beta3>

1. Download this one also

<https://bitbucket.org/ariya/phantomjs/downloads/phantomjs-1.9.8-windows.zip>

1. Create a folder structure **c:\casperjs** and **c:\phantomjs**
2. Extract all the contents of the 1st download into **c:\casperjs**
3. Extract the phantojs.exe file of the 2nd download into **c:\phantomjs** (other files not required)
4. Please ensure you should have a path like this, **c:\phantomjs\phantomjs.exe**
5. Also you must have **c:\casperjs\n1k0-casperjs-4f105a9**
6. Go to system variables, Open the PATH variable under system variables.
7. Add ; at the end if not already there.
8. And add this at the end: **c:\phantomjs;c:\casperjs\n1k0-casperjs-4f105a9\bin;**
9. Save and close.
10. Open Command Prompt, run casperjs

**Running Casperjs Program:**

1. Write the casperjs program in VS / sublime and save that file with **.js** extension

(eg: “program.js”).

1. Go to command prompt and go to the folder where that casperjs program has been saved, and give command like this

**f:\XAVICA\GitHub\gitworkflow\casperjs> casperjs “program.js”**

**Installing Lodash in Casperjs:**

1. Go to the folder where casperjs programs are written, and give the command like

**f:\XAVICA\GitHub\gitworkflow\casperjs>** npm install lodash

1. You will get the below message back.

lodash@3.9.3 node\_modules\lodash

1. Include the below line in casperjs program.

var \_ = require('lodash');

**“Evaluate()” Method:**

1. Think of the evaluate() method as a gate between the CasperJS environment and the one of the page you have opened. Every time you pass a closure to evaluate(), you’re entering the page and execute code as if you were using the browser console.
2. For More information refer

<http://casperjs.readthedocs.org/en/latest/modules/casper.html>

1. It doesn’t accept “forEach()” Method, it will iterate the loop but will not execute statements inside.
2. It doesn’t work for the string function "startsWith()".
3. For console output then only method that works in this function is \_\_utils\_\_.echo();

Eg: **\_\_utils\_\_.echo(“hello”);**

**Logging & Mailing:**

1. casperjs and phantomjs have own fs (FileStream object), So it doesn’t work if you install fs module of nodejs. Note: When you run casperjs programs make sure node\_modules shouldn’t have fs module.
2. Casperjs doesn’t have mailer module. So we have to use nodejs modules for sending emails.

**Points to Remember:**

1. Casper is asynchronous in its execution, so keep each process in a separate casper.then() method. With the then() method, casper waits for the current method statements to execute before jumping into next then() function.
2. After entering into casper.then() function, for console output

Eg: **this.echo(“hello”);**

1. Better to keep loading of images on web page off to crawl the page fast, as the default timeout of casper is 5000ms, to exit a process.

**NODEJS**

**Running Nodejs Program:**

1. Write the casperjs program in VS / sublime and save that file with **.js** extension

(eg: “program.js”).

1. Go to command prompt and go to the folder where that nodejs program has been saved, and give command like this

**f:\XAVICA\GitHub\gitworkflow\nodejs> node “program.js”**

**Installing Lodash in Nodejs:**

1. Go to the folder where casperjs programs are written, and give the command like

**f:\XAVICA\GitHub\gitworkflow\nodejs>** **npm install --save lodash-node**

1. Then you get a folder name called ‘lodash-node’ in the above directory structure, under ‘node modules’ folder.
2. Include the below line in nodejs program.

**var \_ = require('lodash-node');**

**Installing Request in Nodejs (For running http methods):**

1. Go to the folder where nodejs programs are written, and give the command like

**f:\XAVICA\GitHub\gitworkflow\nodejs>** **npm install request**

1. Then you get a folder name called ‘request’ in the above directory structure, under ‘node modules’ folder.
2. Include the below line in nodejs program.

**var request = require('request');**

**Installing Request in NodeMailer (For Sending mail):**

1. While installing nodemailer, have to use Specifically this version (the newer versions has nodejs dependency)

**f:\XAVICA\GitHub\gitworkflow\nodejs>npm install** [**nodemailer@0.7.0**](mailto:nodemailer@0.7.0)

**AZURE STORAGE**

**Setting Path for uploading Image in Azure:**

1. Go to Environment variable for your account.
2. In user variable, click new and create two variables with the below details
   1. Variable Name: **AZURE\_STORAGE\_ACCOUNT**

Variable Value: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Variable Name: **AZURE\_STORAGE\_ACCESS\_KEY**

Variable Value: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Create a folder named ‘Images’ in the folder where your nodejs azure image upload / download program is there.

**Installing Azure Storage in Nodejs (For Download / Uploading Images):**

1. Go to the folder where noderjs programs are written, and give the command like

**f:\XAVICA\GitHub\gitworkflow\nodejs>** **npm install azure-storage**

1. Then you get a folder name called ‘azure-storage’ in the above directory structure, under ‘node modules’ folder.
2. Include the below line in nodejs program.

**var storage = require('azure-storage');**