For WedDrive r protocol

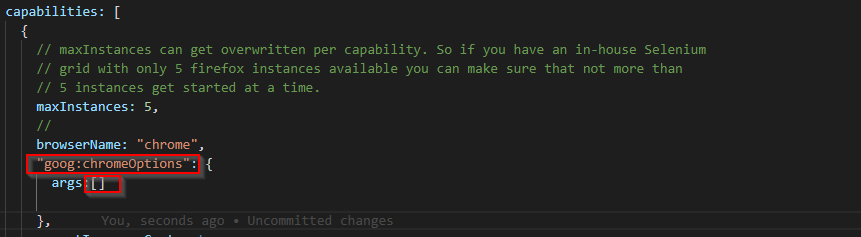
<https://www.w3.org/TR/webdriver2/>

[https://www.w3.org/TR/webdriver2/#capabilities](https://www.w3.org/TR/webdriver2/" \l "capabilities)

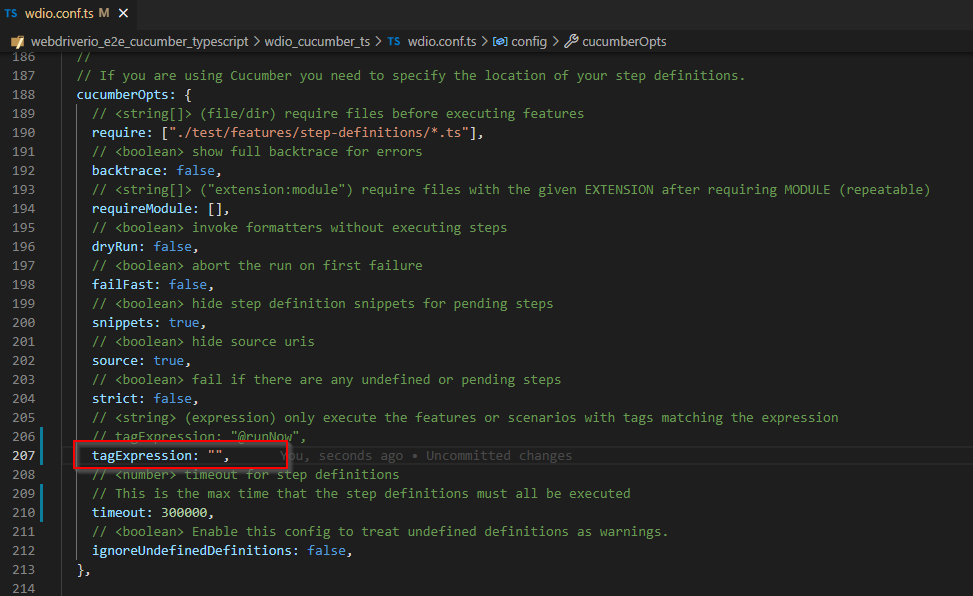
Chrome options:

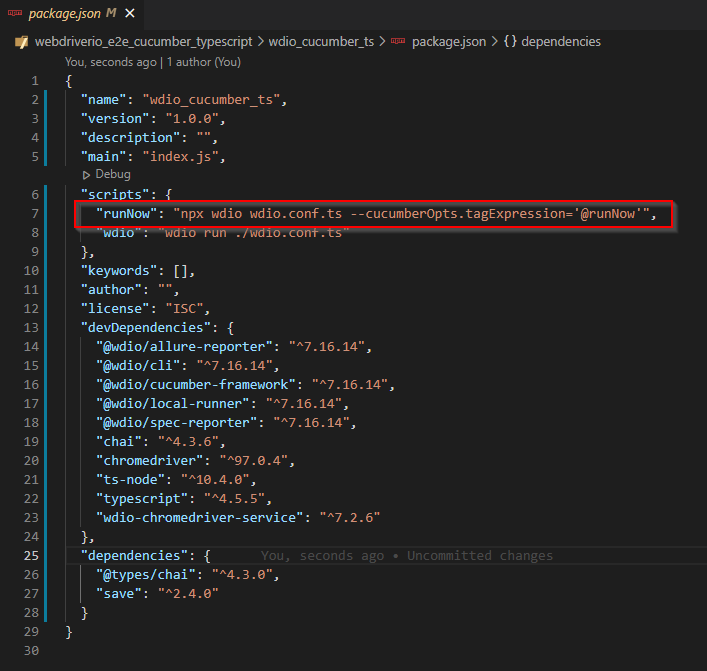
https://www.chromium.org/developers/how-tos/run-chromium-with-flags/

<https://peter.sh/experiments/chromium-command-line-switches/>

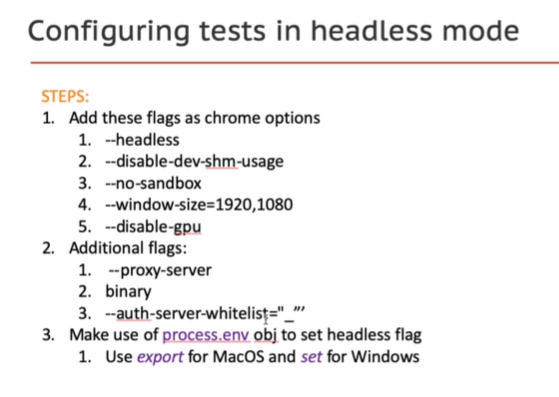


Set tagExpression=”” and add script to pass equivalent to via cmdline

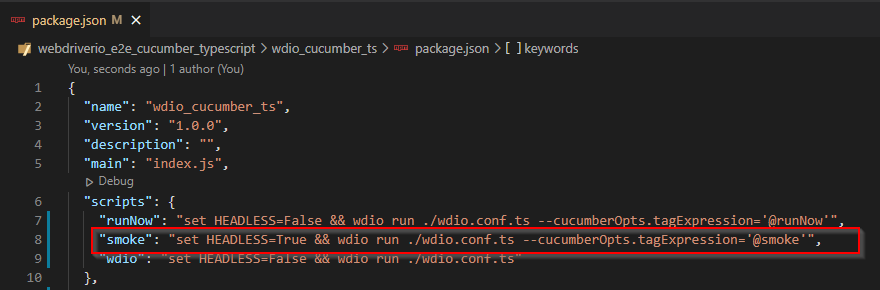




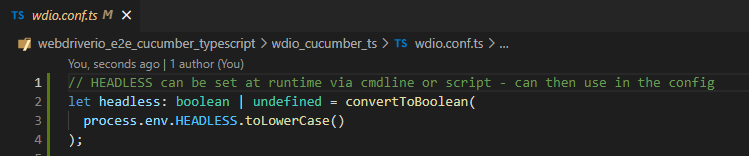
Headless Mode



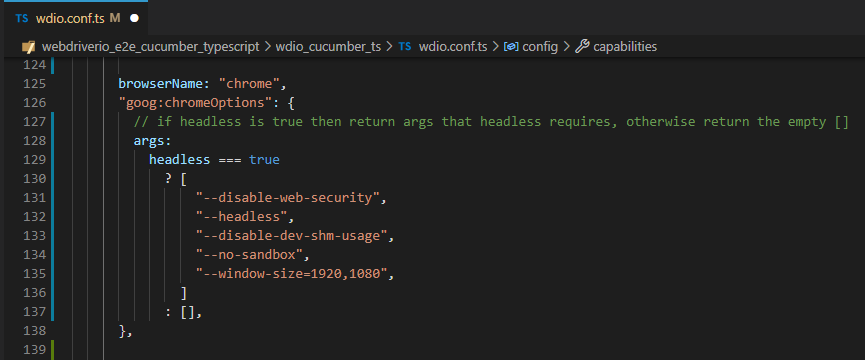
Set a ‘HEADLESS’ key in the process.env that can use during runtime and do what you need



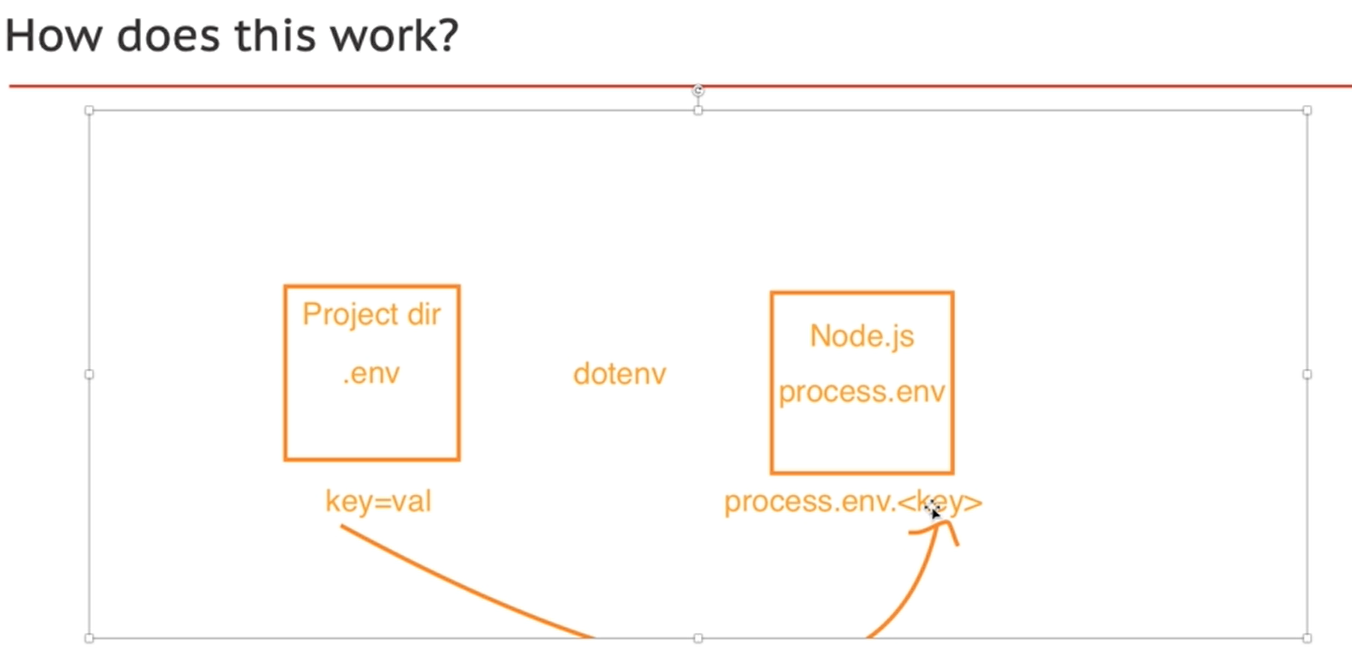
1st file called will be the wdio.conf.ts so can grab it there



And then use if(headless)

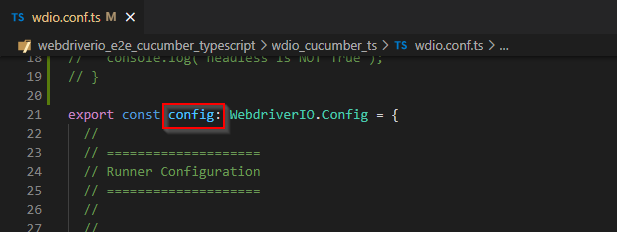


dotEnv loads env variables from .env file (same as if set via cmd line)

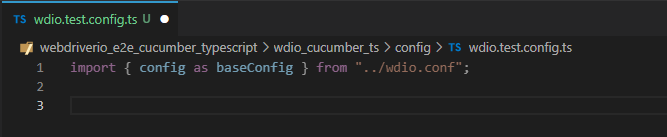


npm install dotenv --save

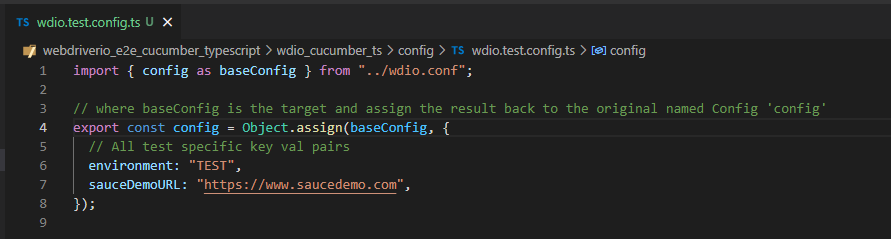
Merge config files for different environments

The general wdio.conf.ts is being exported as a named Config == ‘config’ 

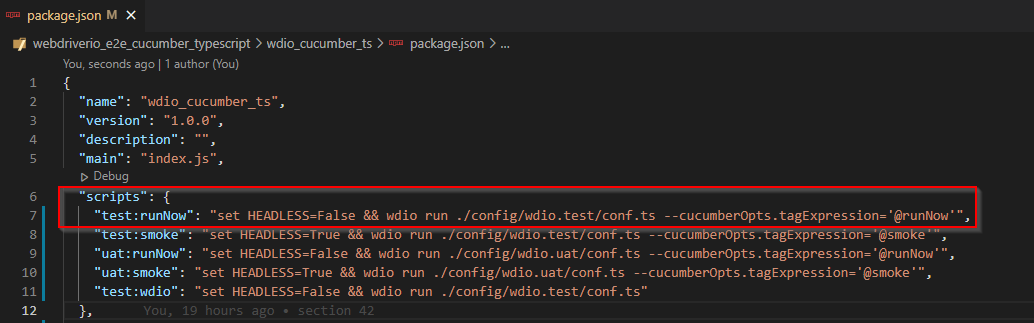
cannot use the same name otherwise will get a name conflict. Therefor use an alias ‘baseConfig’

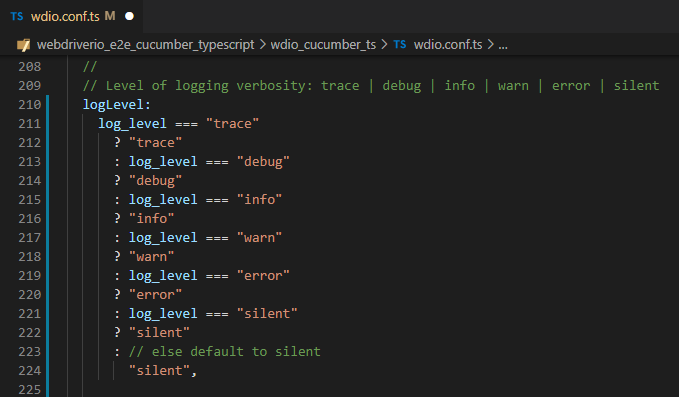


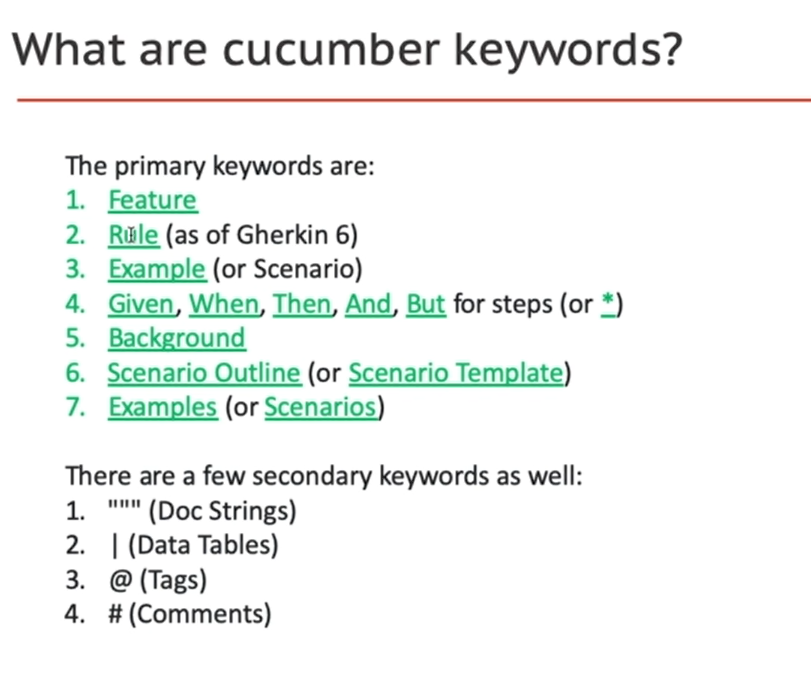
Then use assign to ‘merge’ any values into the config, otherwise will use base values

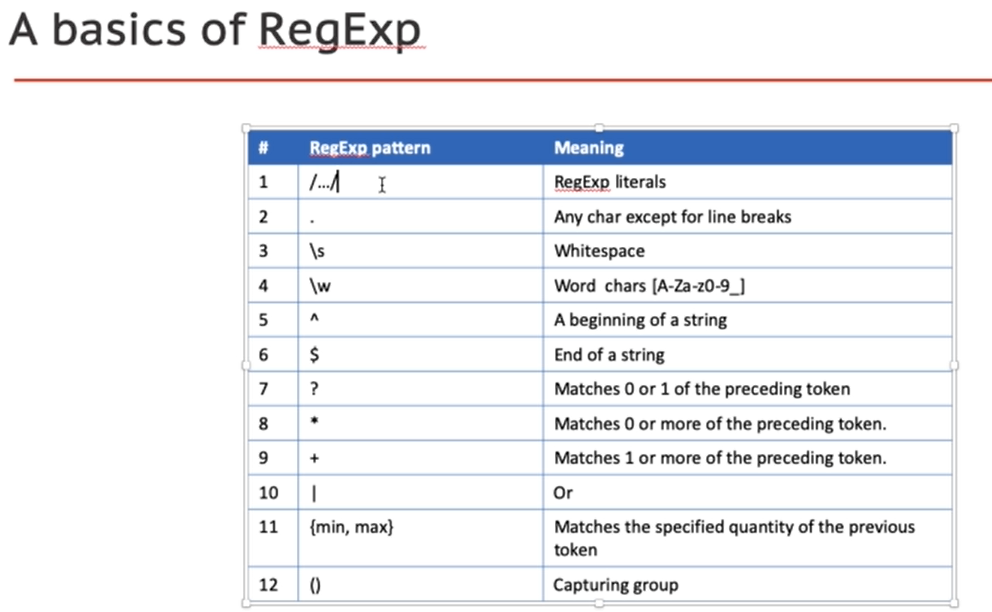


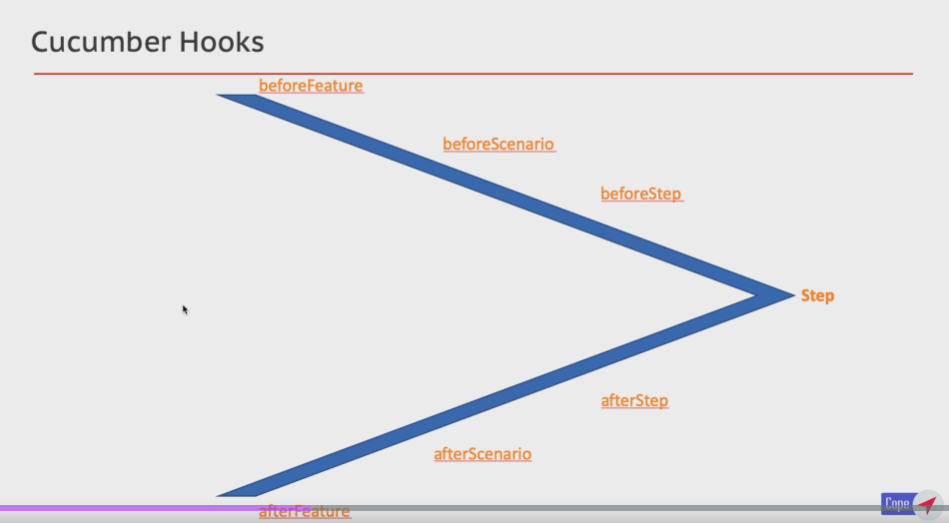
Update scripts to point to new config files

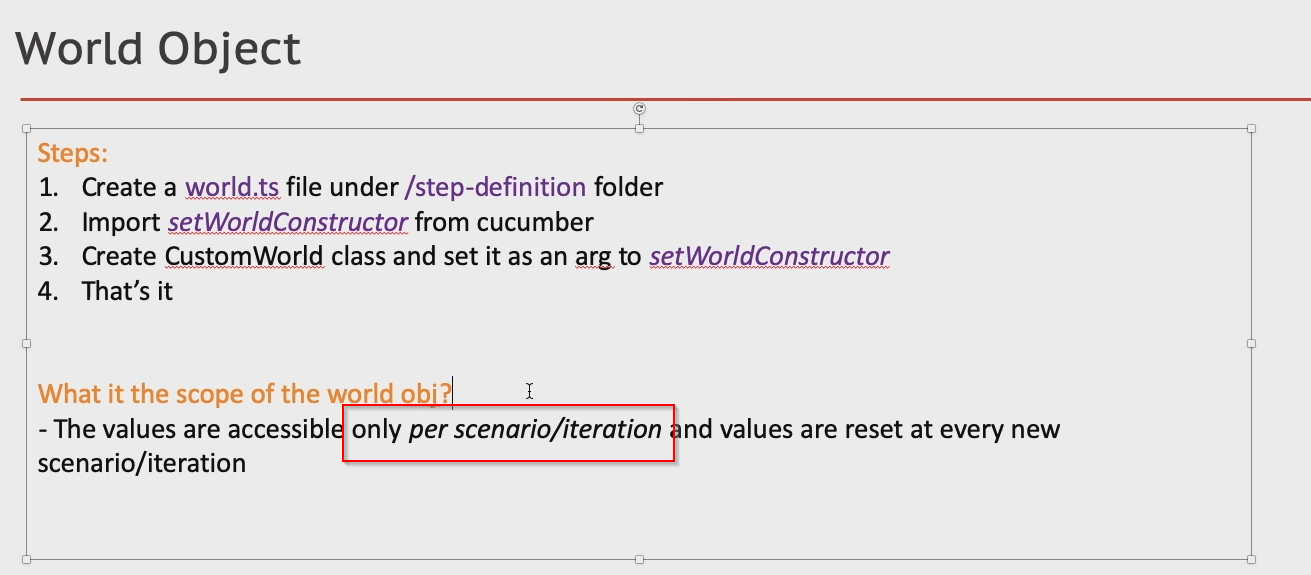


Can include other env parameters for example ‘log\_level’. Retrieve in same way and then set in config with nested ternary operator: 









Logger:

npm i --save-dev winston

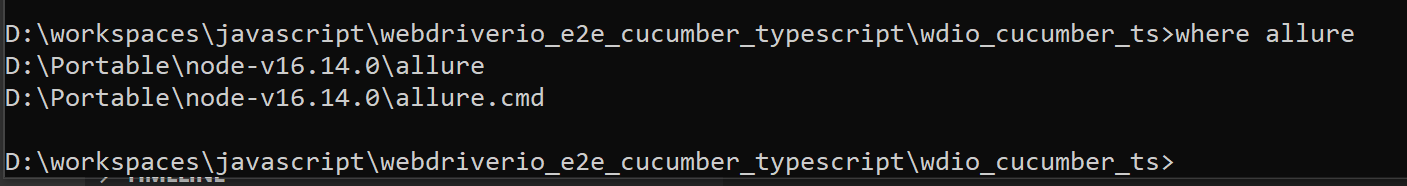
Allure reporting command line

the -g will make it available via the commandline (global install)

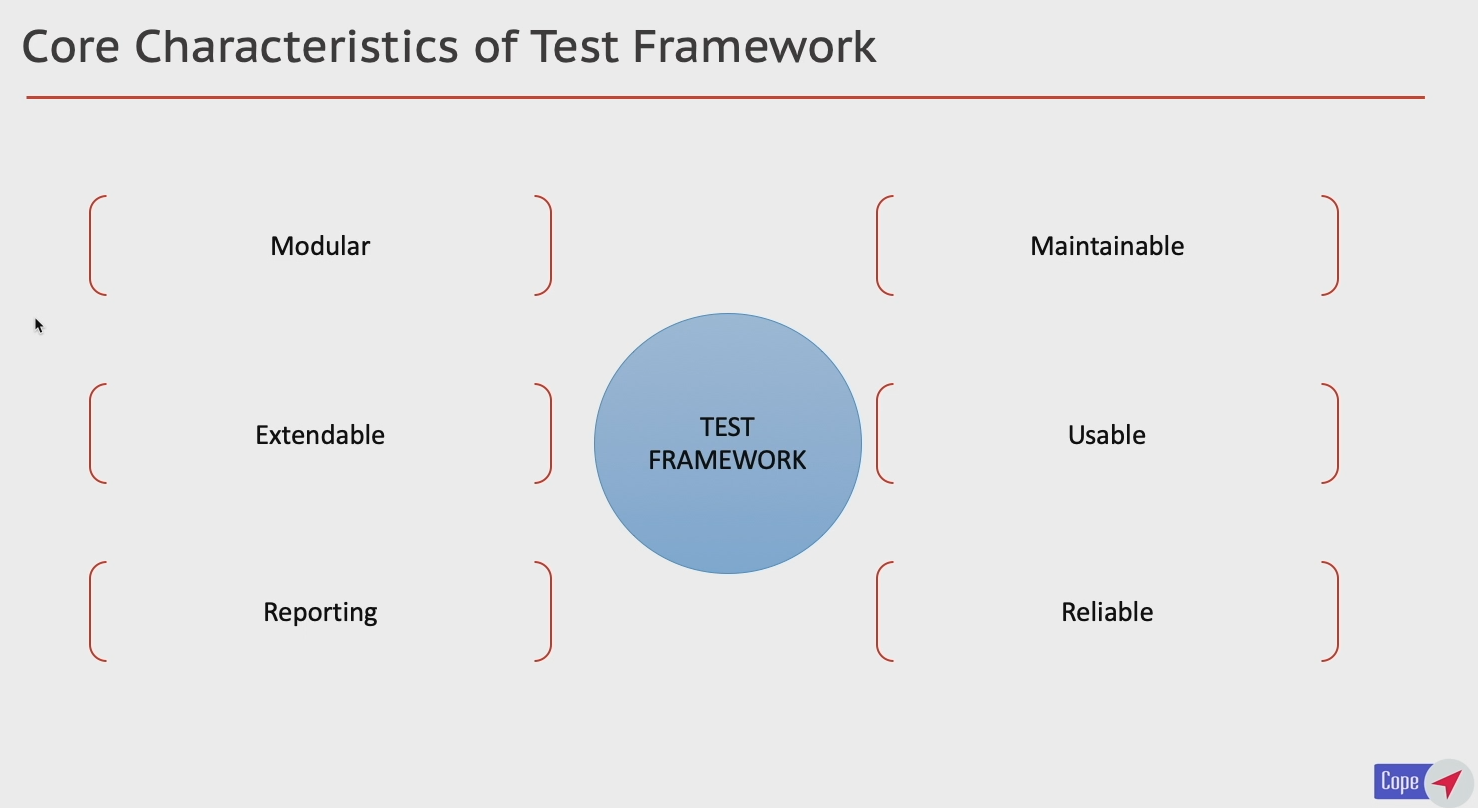
npm install -g allure-commandline –save-dev

look for package in windows:

where allure



allure serve ./results/allure-results



Modular – as levels of abstraction this makes it easier to maintain

Extendable – handle not only web

c