Jmeter Fundamentals and Advanced Course Works

Contents

[1. Week 4 Course Works: Handle Dynamic Token, Assertions, Using CSV Data File 4](#_Toc526624461)

# Week 4 Course Works: Handle Dynamic Token, Assertions, Using CSV Data File

* **Create a Test Plan called “KIJIJI Login”**
* Recording and running a Performance Test for www.kijiji.ca website with log in function:

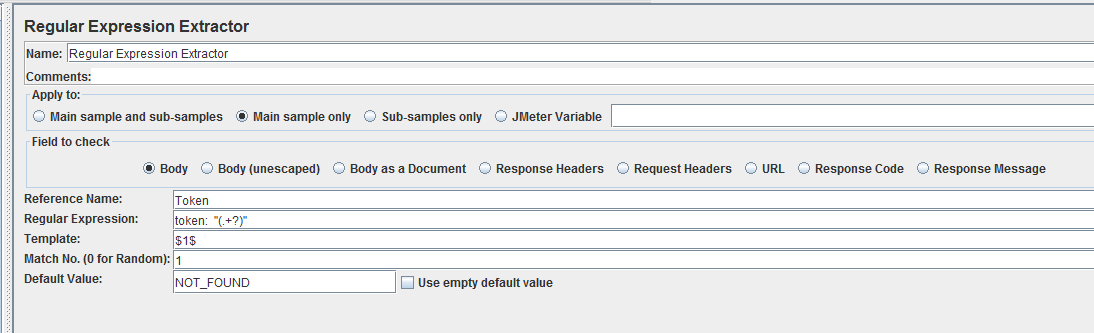
(Pre-condition: an account is needed to create before running this test)

* + Add HTTP(S) Test Script Recorder into WorkBench, then click on Add Suggested Excludes
  + Open www.kijiji.ca website
  + Click on “Sign In”, then enter Email Address and Password, then click on “Sign In”
  + After Sign In, click on Sign out
  + Save Test Plan
  + Add Summary Report, View Result In Table and View Result Tree from Listeners
  + Search for the HTTP Request that has PASSWORD to log in
  + Change the Request Description to “KIJIJI Login”
* **Create Dynamic Token value for KIJIJI Login Test Plan using Regular Expression Extractor**
* Run “KIJIJI Login” Test Plan:
  + in View Results Tree, Search for “token”, under the first HTTP Request that has “token” in the Response, using RegExp Tester to test for Regular Expression:

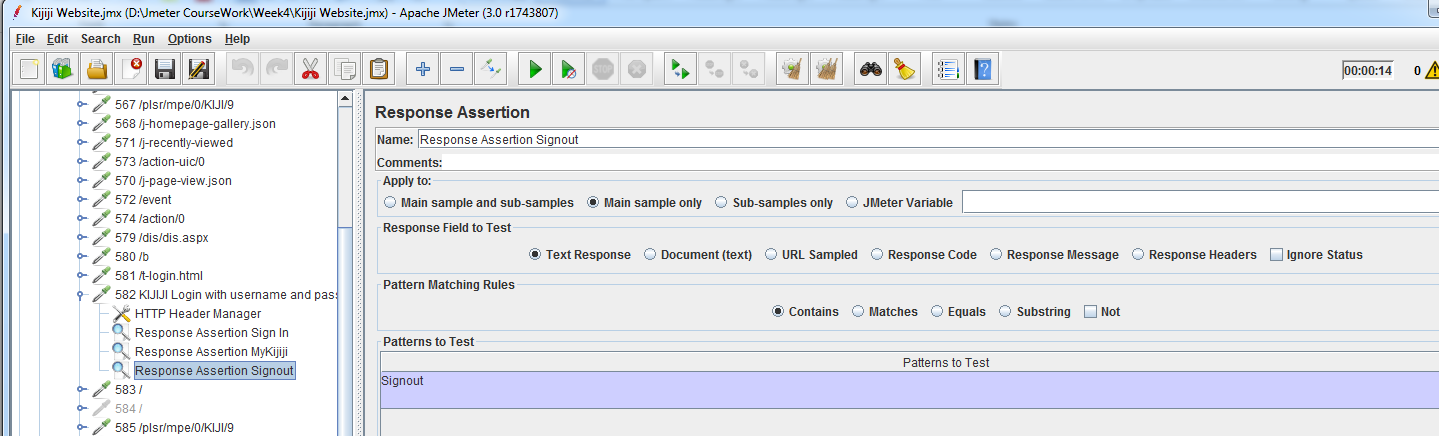
token: "(.+?)"

* + Under the first HTTP Request that returns “token” in the Response, add Regular Expression Extractor to extract value from Login response that match

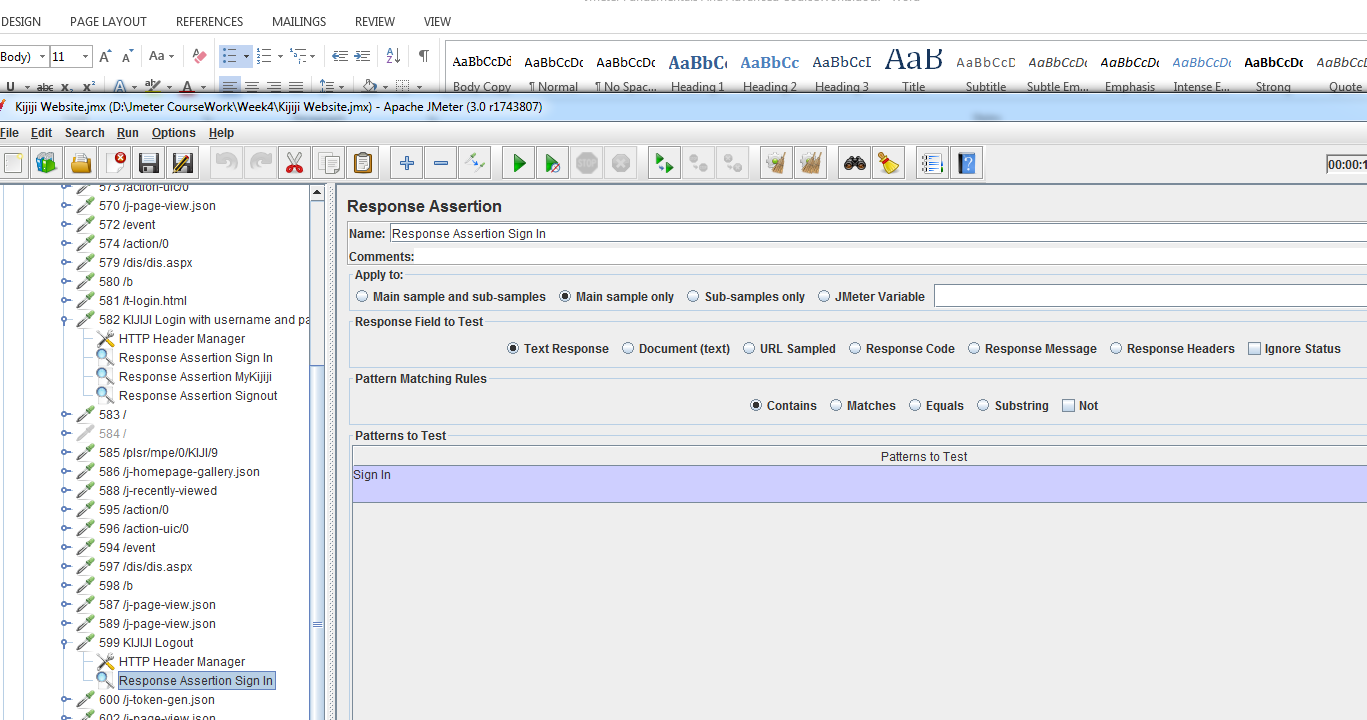
token: "(.+?)"



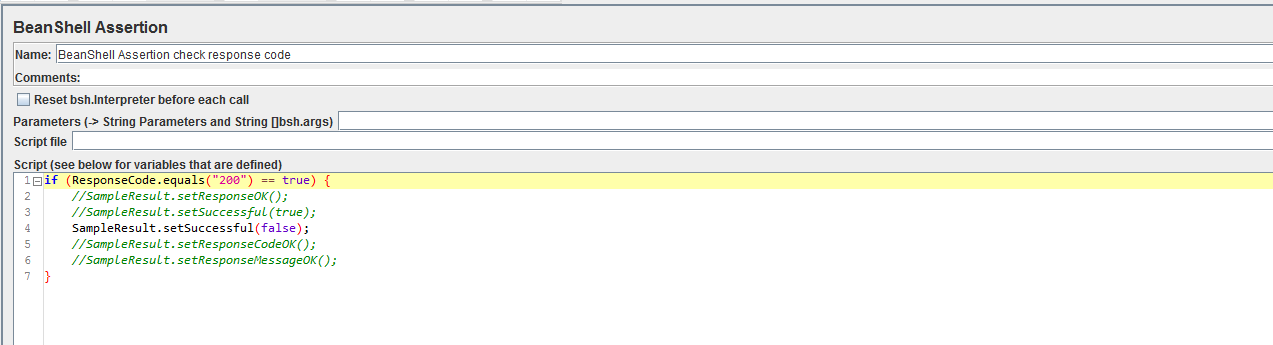
* Replace value of ca.kijiji.xsrf.token in all HTTP Requests that have been failed by ${Token}
* Run Test Plan again
* **Create Assertions for KIJIJI Login Test Plan** 
  + To make sure that the Login is successful, we add Response Assertions under the Login Request to check for a value of the page after log in successful. Add Response Assertions with “Sign Out”



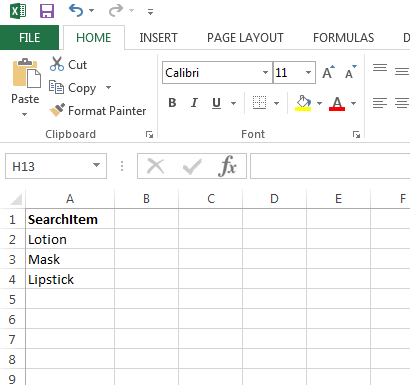
* + To make sure that the Sign out successful, we add Response Assertions under the Sign out Request to check for a value of the page after sign out successful. Add Response Assertions with “Sign In”



* + Beanshell Assertion: add a Beanshell Assertion to check for the reponse code.



* + Size Assertion: add a size assertion to make sure the size should be smaller than 500KB
  + Duration Assertion: add a Duration Assertion to make sure the response time is less than 3 seconds.
* **Using csv data file to Search for different item in** [**www.clinique.ca**](http://www.clinique.ca) **website:**
  + Create csv file data (CliniqueSearchData.csv) same as the screen shot below:



* + Open Clinique Login Test plan
  + Adding CSV Data Set Config into Thread Group, set Recycle on EOF=False, set Stop thread on EOF=True
  + Change Number of Thread (users)=3
  + Under the request for Searching Lotion, change Value of Search in the Request by ${SearchItem}
  + Run Test Plan

