## Build Acceptance Automation

##### Description**:**

Build Acceptance Automation will be a set of test cases before QA accepts code from Application Development. It consist of chosen test cases from the Test Case Suite to hit all major paths of the application (API, Call Flow and Transfers) to ensure basic functionality. This automation will easily help QA to verify that the application is working well and no major blockers or bugs found. It will be automated using Jenkins.

##### Reason for creating:

With the Build Acceptance Automation, critical paths such as API, Call flows and Transfers can be tested on a multiple run to check if the application is working well and if QA can accept the code for major testing effort.

With the Build Acceptance Automation, bugs or any blockers can be identified earlier

##### Benefits:

* Easily identify bugs on the application
* Allows the user to do other tasks while the program is running on its own
* Saves time and effort

##### Who will benefit:

This project will be beneficial for QA. *<To ask Ms. K what else can be inserted here>*

##### Cost Saving Analysis:

*<On research>*

##### Tools to be used:

* JMeter
* Jenkins
* GitHub

##### Dependencies:

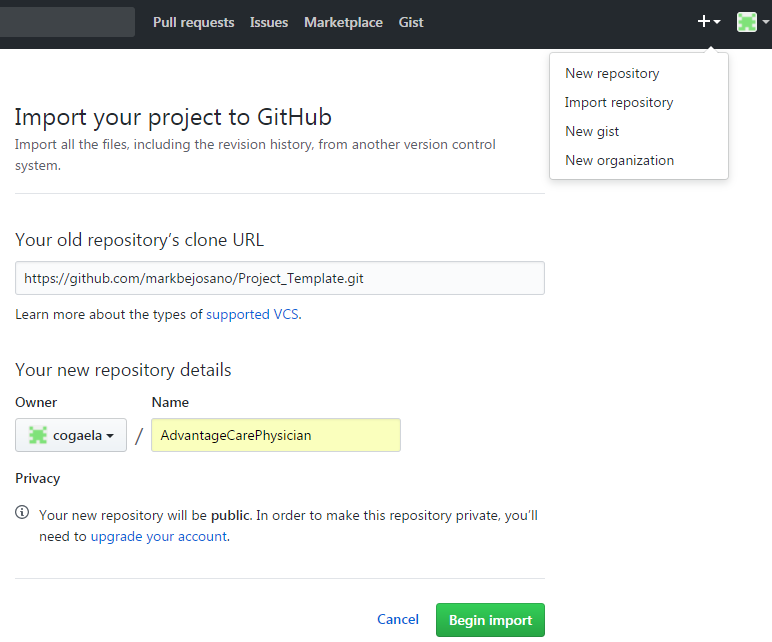
* <http://10.53.32.253:8080/>
* https://github.com/markbejosano/Project\_Template.git

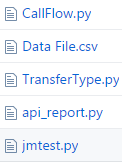
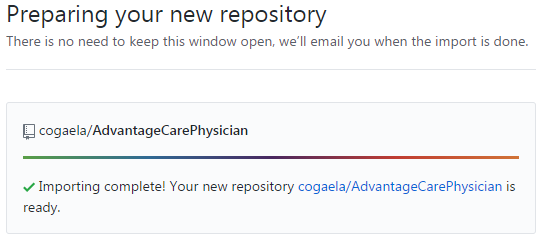
##### How to automate using Jenkins?



• User needs to have a GitHub account.

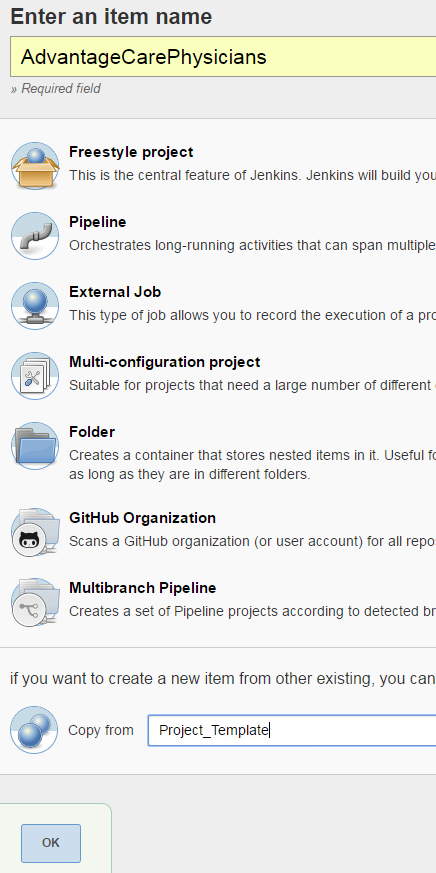
###### User Create Project Repository on GitHub



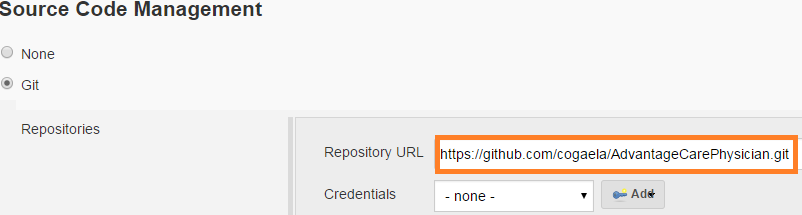
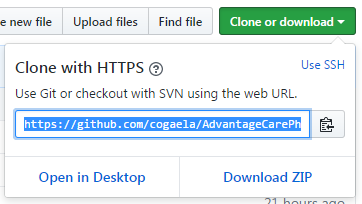


1. Click on the “Add” icon > select “Import Repository”
2. Copy this repository URL <https://github.com/markbejosano/Project_Template.git>
3. Paste the copied URL on the text field
4. Enter Project Name
5. Click button
6. Click on the link to view the cloned repository
7. Files that should be displayed after cloning the Template Repository

###### User Create Project Workspace on Jenkins



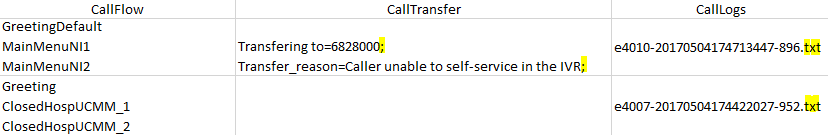
1. Go to Jenkins. Click - <http://10.53.32.253:8080/> and click 
2. Enter your Project Name
3. Enter “Project\_Template”
4. Click “Ok” button
5. Go back On your GitHub, click  and copy the URL
6. On Jenkins scroll down > Source Code Management > Repository URL > Paste the copied URL
7. Click  button



###### Test Planning

Do not forget to create a text file of the call logs you will

A. Call Flow and Call Transfer Set-up on Excel File

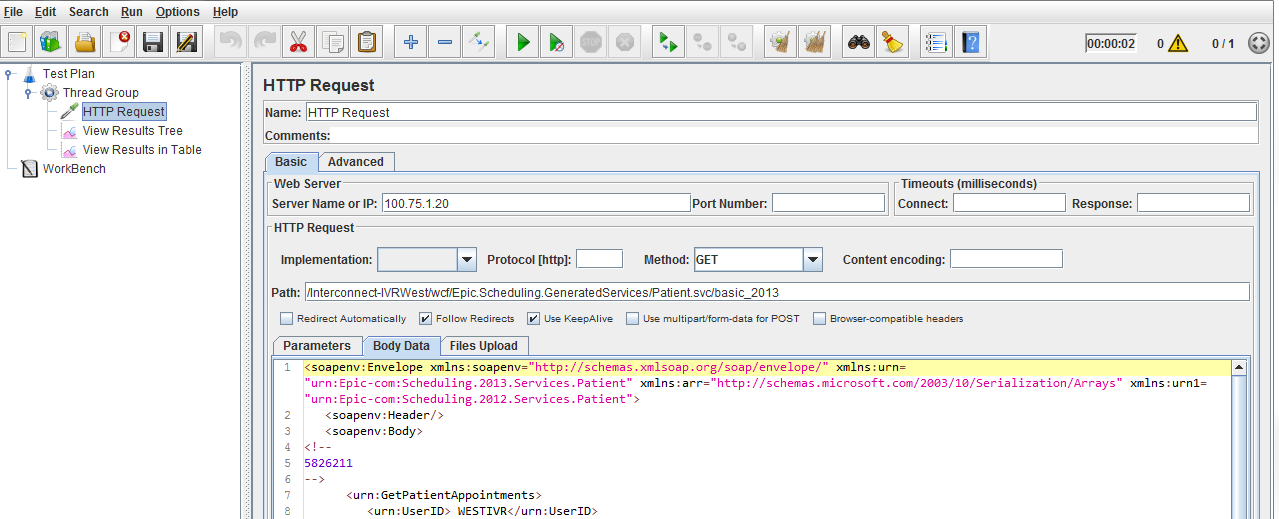


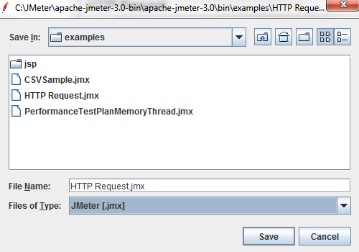
1. Download Data File.csv from your GitHub
2. On the “CallFlow” column, enter the expected prompts to be displayed
3. On the “CallTransfer” column, enter the expected Transfer Values
   1. Don’t forget to add a semi-colon after transfer values
4. On the “CallLogs” column, enter the call id
   1. Don’t forget to include **.txt** extension
5. Save the excel file as **.csv** file.

B. API Set-up on JMeter

1. Open JMeter > Click on New
2. Right click on Test Plan, select Add > Thread (Users) > Thread Group
3. Right click on Thread Group, select Add > Sampler > HTTP Request
4. On HTTP Request Page, enter ‘Server name or IP’, ‘Path’ and ‘Body Data’
5. Follow Step 3, select Add > Listener > View Results in Table
6. Follow Step 3, select Add > Listener > View Results Tree
7. Click File > Save Test Plan as > Rename File name as **HTTPRequest** > select Files of Type ‘JMeter [.jmx] then ‘Save’

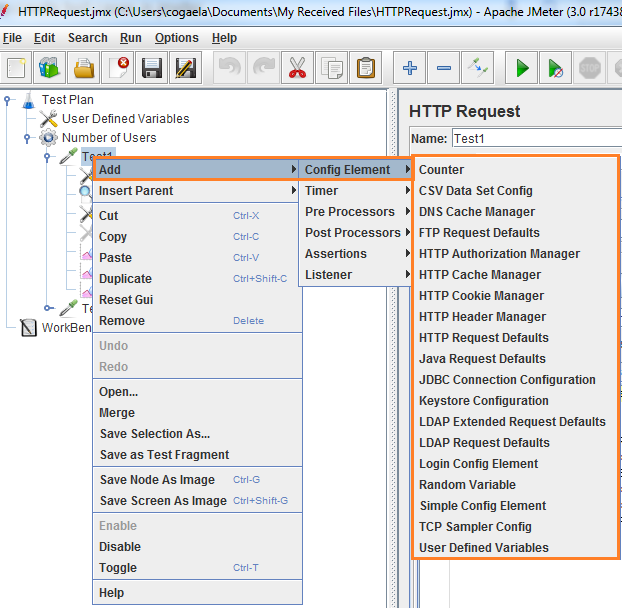
• FileName should always be save as “**HTTPRequest.jmx”**



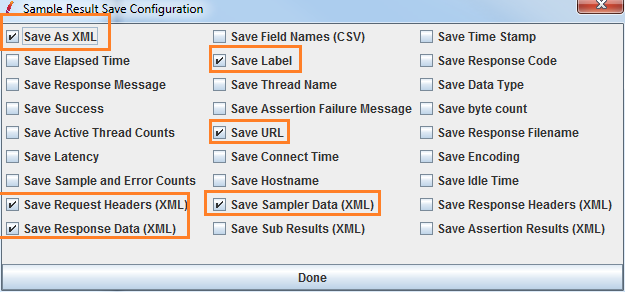




• User will have an additional “Configuration Element” to modify request. Might need to have username and password of the API.



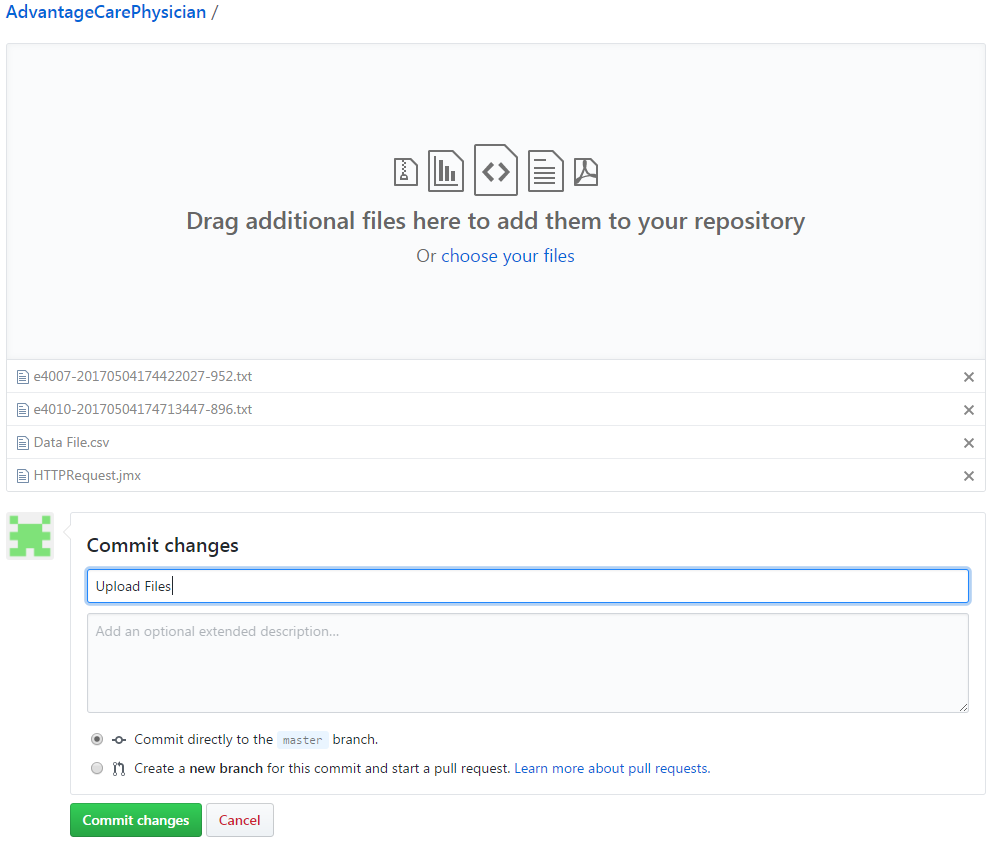




1. Right click on the Test Case > Select Add > Config Element > select any element to modify the request
2. Click on the 
3. Enter “**.xml**” extension on the FileName
4. Click  button
5. Tick on the needed elements
6. Click 
7. Save the file.

###### Test Execution

A. To upload .csv, .txt and .jmx files on Repository

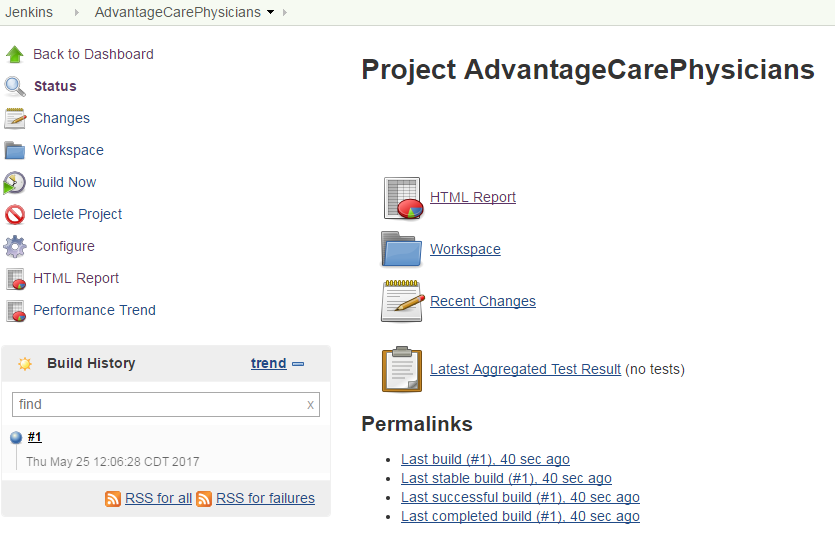


1. On your GitHub repository > on your Project > then click 
2. Drag files or click “Choose Files” to upload **.csv**, **.txt** and **.jmx** files
3. Provide a comment then click “Commit Changes”

B. Run your Project on Jenkins



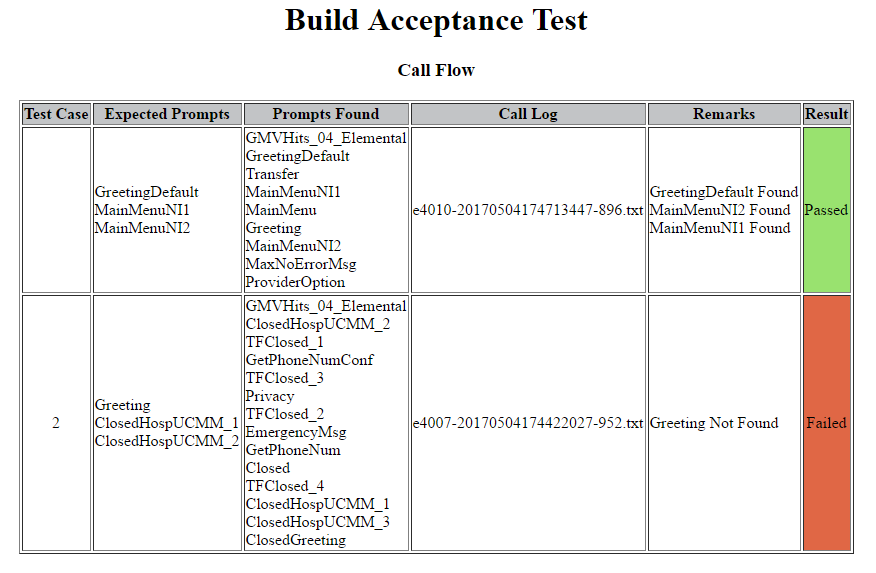
 Result is successful  Result is Failed



1. On Jenkins > your Project > Click 
2. On the Build History, status of your Test will be displayed
   1. Build is done. It will display  or 
   2. If Build is Pass it will display 
   3. If Build is Fail it will display 
3. Refresh the page
4. To see the Overall results, click on the 
5. Test Result will be displayed







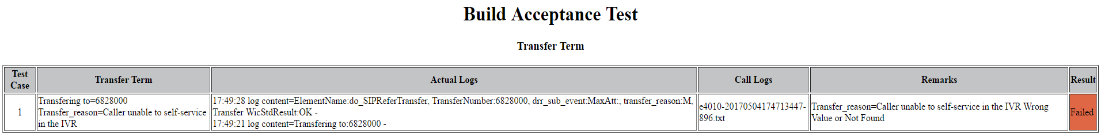
User input on the Data File.csv

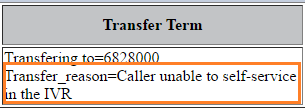
Prompts found on the Call logs

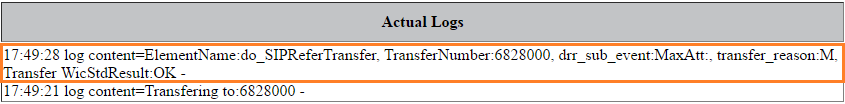
Remarks if expected prompts are found or not found

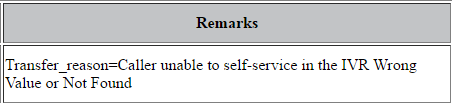
* On Test Case 1 it Passed because all the prompts are found
* On Test Case 2 it Failed because Greeting prompt is not found





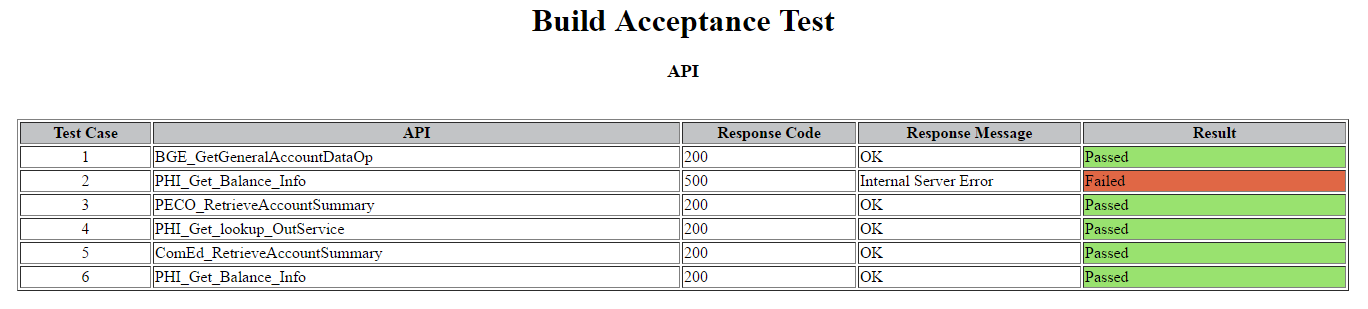






* Test Case Failed because Transfer value (Transfer\_Reason) did not match on the actual call logs



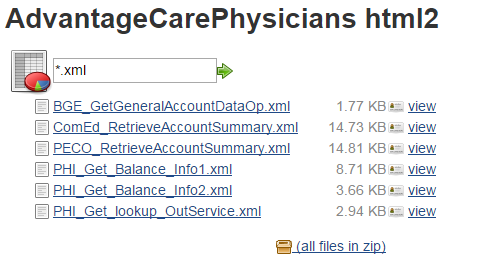


* You may refer to this link for Response code definition

<http://www.restapitutorial.com/httpstatuscodes.html>

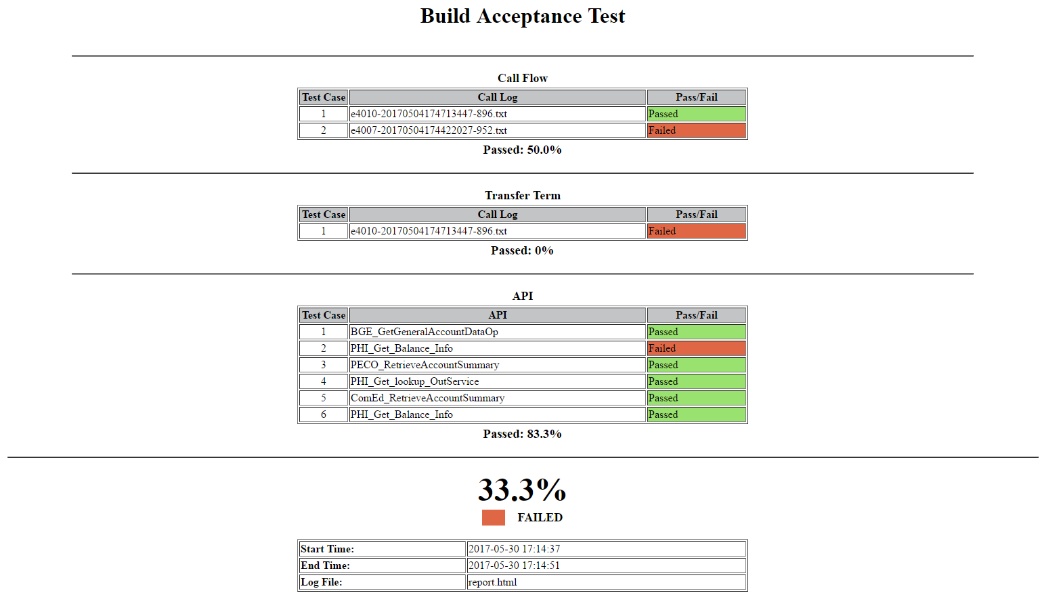
* Test Case 1, 3, 4, 5 and 6 Passed because it response message is 200
* Test Case 2 Failed because there was an Internal Server Error





* Displays the API request used and the corresponding generated response





* Displays the Overall Summary report including the waited average for the Test