BKM for Script 1

The provided code allows creation and updating the test result for each test case present in a given test cycle, with options to specify the link type, status, reason, and owner. The input is a JSON file that can be easily modified to suit the user's needs and the script can be run using either Basic AUTH or Kerberos AUTH authentication APIs.

JSON File Inputs:

a) **Path**: It is the path to csv file which contains all the TC_IDs and their respective TC_Title and configuration.
   NOTE: Order should be maintained in the csv file [TC_ID, TC_Title, Configuration] and first row should contain the Title of each column.

b) **Test Cycle:** It is the name of a particular test cycle. In creation of new TR's this test cycle name will be written in the appropriate field.

c) **Link-Type:** It should be "parent-child" in case of updating and creating the TR's.

d) **Status:** String indicating the status of a TC for a given test cycle and configuration.

e) **Reason:** String indicating the reason (pass, fail, optimized, warning, new, etc.)

f) **Release_Aff:** It will be the program release name to which TC refers, example: bios.alderlake

g) **Owner:** User-id of person making the changes, example "namanaga"

h) **Reason_other**: This field is required when status_reason is set to "blocked.other"

There are two Python Scripts included in the folder: "script1_preprod.py" for testing on the pre-prod HSDES server, and "script1_prod.py" for use on the HSDES production server. These two files fetch input from JSON file. There are two other files "script1_preprod_cmd.py" and "script1_prod_cmd.py" which takes input from command line and is independent of ant JSON file.

The new TEST RESULT created will have "title", "parent-id", "owner", "link-type", "status", "reason" and "release" fields fetched from the JSON input file.

It should be noted that values provided in the JSON file will be applied to all the test results created or updated. Additionally, if using Basic AUTH, valid credentials can be obtained using provided link https://hsdes.intel.com/appstore/token/.

If user wants to give input through command line, then use "script1_preprod_cmd.py" for testing purpose on PreProd HSDES Server and "script1_prod_cmd.py" for actual Production HSDES Server.

Input format of CMD:

```python
    parser = argparse.ArgumentParser()
    parser.add_argument('-p', help="csv file path containing tc_id, tc_tile and
configuration")
    parser.add_argument('-tc', help="Name of test cycle")
    parser.add_argument('-sv', help="value of status field(complete, blocked,
etc)")
    parser.add_argument('-rv', help="value of reason field(pass, fail, optimized,
etc)")
    parser.add_argument('-rl', help="Name of release affected (bios.raptorlake,
etc)")
    parser.add_argument('-lt', help="value of link type (parent-child)")
    parser.add_argument('-o', help="name of owner(eg: namanaga)")
    parser.add_argument('-ro', help="Value of reason.other when
status.reason==blocked.other (eg: manual_result)", required=False)
    args = parser.parse_args()
```

Example 1:

py script1_prod_cmd.py -p C:\NAMAN\Share\Create_Update_TR_using_TC\Book5.csv -tc "bios.a21.A0 PO Exit.Client-BIOS.bob_test" -sv "complete" -rv "pass" -rl "bios.alderlake" -lt "parent-child" -o "namanaga"

Example 2:

py script1_preprod_cmd.py -p C:\NAMAN\Share\Create_Update_TR_using_TC\Book5.csv -tc "bios.a21.A0 PO Exit.Client-BIOS.bob_test" -sv "blocked" -rv "other" -rl "bios.alderlake" -lt "parent-child" -o "namanaga" -ro "manual_result"

NOTE: argument -ro (reason.other) is required only when status.reason == "blocked.other"

<u>**Observations:**</u>

a) In a single API call to update or create, only 25 test results can be created or updated.
b) For a csv containing 700-800 TC IDs, the script may take up to 20 mins to process and execute.