Daisy

No Code - Web Browser, REST & SOAP API and

Mobile application Automation Framework

Daisy2.1-ReleaseNotes December_2022

Daisy2.1 Release includes

❖ **New feature**

➤ Authenticator app support for MFA

➤ API negative testing

➤ Re execute failure scenario

❖ **Enhancement in existing feature**

➤ User choice to print execution statement in report

Persistent

❖ New feature

- Authenticator app support for MFA

➢ Support **MFA** through authenticator app by TOTP method.

➢ User will have to configure authentication app as browser extension

➢ Daisy command **GetAuthenticationCode** will retrieve and store authentication code in provided variable name which can be used in further steps.

➢ User will have to provide app key in encrypted format through config file to this command. App key must be same which user has used while configuring the authentication app

```
DaisyConfig.properties
154
155  # -------------------------------------------
156  # Authenticatior key
157  # -------------------------------------------
158  authenticatior_key=provideYourEncryptedKeyhere
```

Testcase Name	Steps	Field Name	Type	Data	Locator Type	Value
Get Authentication Code	Retrieve code and store in variable ActualCode	Authentication	**GetAuthenticationCode**	ActualCode		
	Enter ActualCode in textbox	password	Textbox	${Var.ActualCode}	locatortype	locatorvaue

3

Persistent

- Authenticator app support for MFA

❖ New feature

API Negative testing support

RunApi:

➢ Old implementation: Steps will marked as fail in execution report for error code >300.

➢ New Enhancement: Provide **NegativeTesting** in value column when user wanted to validate API response for negative testing.

- When in value column **NegativeTesting** is mentioned, In case of error code > 300 , it will not mark as failure in report and user will be able to fetch error message

- When value column will be empty, In case of error code > 300 , it will mark as failure in report.

Testcase Name	Steps	Field Name	Type	Data	Locator Type	Value
In case of error code > 300 , it will not mark as failure in report and user will be able to fetch error message		response	**RunApi**	${DaisyDir}\API\Salesforce_Auth.txt		**NegativeTesting**
In case of error code > 300 , it will mark as failure in report		response	**RunApi**	${DaisyDir}\API\Salesforce_Auth.txt		

Persistent

❖ New feature

- API Negative testing support

STATUS	TIMESTAMP	DETAILS
✅	16:59:11	Running POST call for URL :https://login.salesforce.com/services/oauth2/token
✅	16:59:12	Response for Salesforce_Auth.txt: {"error":"unsupported_grant_type","error_description":"grant type not supported"}
✅	16:59:12	Successfully Ran API for : Salesforce_Auth.txt
✅	16:59:12	Got response Successfully: unsupported_grant_type for the variable: messageErr
✅	16:59:12	Successfully Validated , Actual Result: unsupported_grant_type , Expected Result : unsupported_grant_type
✅	16:59:12	Running POST call for URL :https://login.salesforce.com/services/oauth2/token
❌	16:59:12	Run API failed for : Salesforce_Auth.txt
❌	16:59:12	The API returned a error code of 400Response : {"error":"unsupported_grant_type","error_description":"grant type not supported"}
✅	16:59:12	Got response Successfully: unsupported_grant_type for the variable: messageErr
✅	16:59:12	Successfully Validated , Actual Result: unsupported_grant_type , Expected Result : unsupported_grant_type
❌	16:59:12	This execution report contains failed test steps

APIDemo — Fail

Field Name	Type	Data	Locator Type	Value
responsevar	RunApi	${DaisyDir}\API\Salesforce_Auth.txt		NegativeTesting
messageErr	SetVarApi	$.responsevar.error		
	Verify	nsupported_grant_type;SOFTASSERT		${Var.messageErr}
responsevar	RunApi	${DaisyDir}\API\Salesforce_Auth.txt		
messageErr	SetVarApi	$.responsevar.error		
	Verify	nsupported_grant_type;SOFTASSERT		${Var.messageErr}

Persistent

❖ Enhancement in existing feature

Daisy execution report

➢ Old implementation: After execution of every step, it will print respective execution statement in report

➢ New Enhancement:

  ➢ User choice to print execution statement in report

  ➢ User will have to provide **DNR** in filed name for the steps which user don't want to get printed in execution report
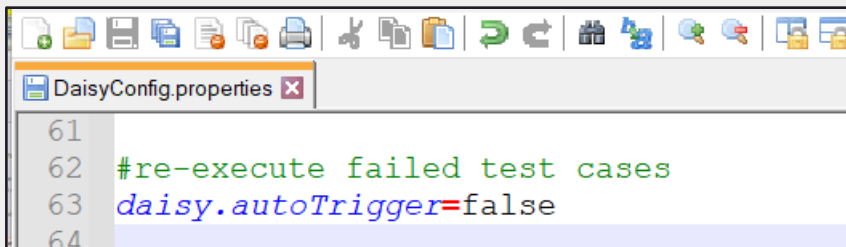
    Note :

    • In case of execution failure/error/exception, it will always get reported in execution report

    • In case of empty filed name, step execution details will get printed in execution report.

Test Case Name	Steps	Field Name	Type	Data	Locator Type	Value
It will print information about this step-in report		Username	Textbox	TestData	LocatorType	Locator value
It will not print information about this step-in report		**DNR**	Textbox	TestData	LocatorType	Locator value
It will not print information about this step-in report		**DNR**	ClickJS		LocatorType	Locator value

Persistent

❖ Enhancement in existing feature

- Re execute failure scenario/script
  - ➢ After completing execution of all scenarios/scripts, Daisy will re trigger the execution of scenarios/scripts which are failed. Execution report of failed scenarios/scripts will be generated agin.
  - ➢ To re executed failed scenarios/scripts immediately after finishing the execution of all scenarios/script, user will have to set **daisy.autoTrigger** flag of respective DaisyConfig.properties file as **true**.
  - ➢ If all scenarios/scripts are passed it will not re execute any scenario/script, on console it will print message **"There are no failed scripts"**
  - ➢ By default, value of **daisy.autoTrigger** flag of respective DaisyConfig.properties file is set as **false.**

  ```
  DaisyConfig.properties
  61
  62   #re-execute failed test cases
  63   daisy.autoTrigger=false
  64
  ```

  - ❖ **Note:** Re triggering failed scenario/script feature is applicable only for sequential execution. This feature is not supported for parallel execution.

Persistent

❖ **Enhancement in existing feature**

- Re execute failure scenario/script
  - ➢ If all scenarios/scripts are passed it will not re execute any scenario/script, on console it will print message **"There are no failed scripts"**

❖ **Enhancement in existing feature**

- Re execute failure scenario/script
  - ➢ To re executed failed scenarios/scripts immediately after finishing the execution of all scenarios/script, user will have to set **daisy.autoTrigger** flag of respective DaisyConfig.properties file as **true**.
    - ▪ Example:
      - ▪ In Master.xls file execution flag for two scripts was marked as Yes, so Daisy has executed these two scripts.
      - • Script **Script for IsSelected** got failed hence only this script was re executed and as per the latest execution report is generated.

Persistent

❖ **Enhancement in existing feature**

- Re execute failure scenario/script

Thanks!