

How to use CAT

---- It is necessary to have Powershell 7 to run CAT-----

1. Installation

From version 0.4.0, we use this way to install it.

RUN THIS script:

```
$credential = Get-Credential
$feedUrl =
"https://pkgs.dev.azure.com/joyfulcraftsmen/Products/_packaging/CAT/nuget/v2"
Register-PSRepository -Name JoyfulDevOps -SourceLocation $feedUrl -
PublishLocation $feedUrl -InstallationPolicy Trusted -Credential $credential
Install-Module CAT -Repository JoyfulDevOps -Credential $credential -Force -
AcceptLicense
```

When you are prompted for credentials, fill in your email as user name and the PAT token as password.

2. Define \$credentials:

```
$token = "*****" #token must be written in
between ""
$credentials = Get-Credential
```

```
PowerShell credential request
Enter your credentials.
User: your email address
```

Then Run this script:

```
Install-module CAT -RequiredVersion '0.5.0' -Repository JoyfulDevOps -
AcceptLicense -force -Credential $credentials
```

-> Module CAT is installed and ready to use

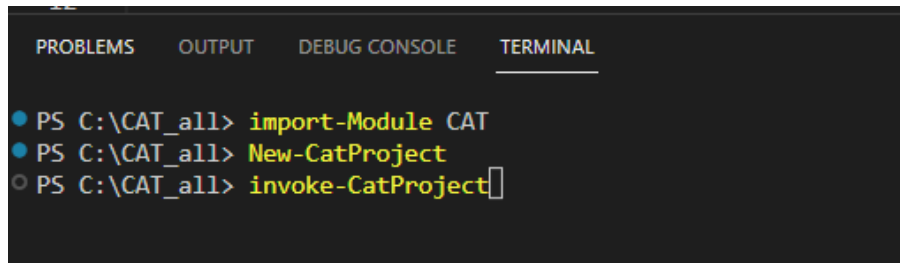
3. How to work with CAT

We can recommend using Visual studio code & yaml extension.

Open folder with CAT and be in this directory while using it.

Commands must be run from Terminal of PowerShell 7

4. Commands for CAT :



```
PS C:\CAT_all> import-Module CAT
PS C:\CAT_all> New-CatProject
PS C:\CAT_all> invoke-CatProject
```

`Import-module CAT`

#run first to import CAT module

`New-CatProject`

#this command create project file with first tests

`Get-Command -Module CAT`

#this shows all CAT commands, try it and experiment with it

`Invoke-CatProject`

#to run tests&see results use this command

`Get-CatTestResult | ? { $_.TestResult -eq 'Failed' }`

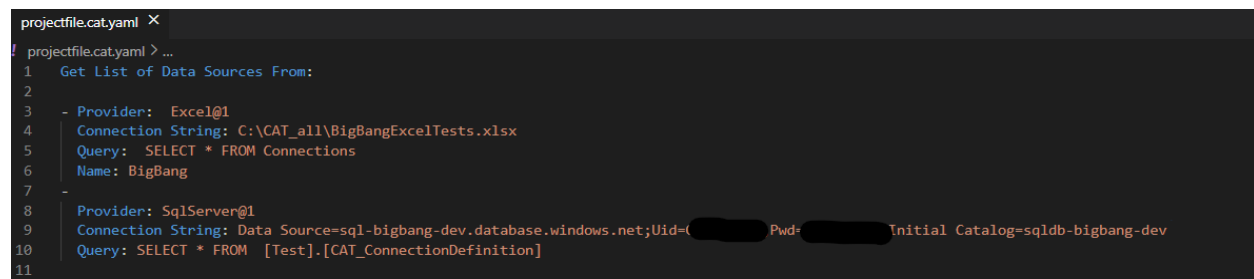
shows us only 'Failed' tests, it's good for debugging

5. How to define project file and set up tests

Example of data source definition:

Get List of Data Sources From:

```
-
  Provider: SqlServer@1 # Excel@1 # Yaml@1 Exceloledb@1
# these are basic types of providers
  Connection String: Data Source=sql-bigbang-
dev.database.windows.net;Uid=%%%;Pwd=%%%;Initial Catalog=%%% or path to file
  Query: SELECT * FROM [Test].[CAT_ConnectionDefinition]
#select where is in database the connection definition
  Name : AWD
# nickname of database, we use it for FirstDataSource or SecondDataSource
```



```
projectfile.catyaml X
! projectfile.catyaml > ...
1  Get List of Data Sources From:
2
3  - Provider: Excel@1
4    Connection String: C:\CAT_all\BigBangExcelTests.xlsx
5    Query: SELECT * FROM Connections
6    Name: BigBang
7
8  - Provider: SqlServer@1
9    Connection String: Data Source=sql-bigbang-dev.database.windows.net;Uid=( ) Pwd=( ) Initial Catalog=sqlldb-bigbang-dev
10   Query: SELECT * FROM [Test].[CAT_ConnectionDefinition]
11
```

Example of tests definition:

Get List of Tests From:

```
-
  Provider:  SqlServer@1
  Connection String: Data Source=sql-bigbang-
dev.database.windows.net;Uid=%;Pwd=%;Initial Catalog=%
  Query: EXEC [Test].[CAT_Smoke_GetAllEmptyTables_Staging] # we can run
stored procedures which produce tests
-
  Provider:  SqlServer@1
  Connection String: Data Source= sql-bigbang-
dev.database.windows.net;Uid=%;Pwd=%;Initial Catalog=%
  Query: SELECT * FROM [Test].[CAT_TestDefinition] # we can select tests from
table
-
  Provider:  Excel@1
  Connection String: C:\CAT_all\BigBangExcelTests.xlsx # string to file
where are tests
  Query: SELECT * FROM DataMart_Smoke # select excel sheet where are tests
  Name : excel DataMart_Smoke
-
  Provider: Yaml@1 #set up provider like this, allows me to to run tests ,
which are define directlly in project yaml file
  Connection string: ./projectfile.cat.yaml # name of my projet file
  Query: Tests
  Name: Testy v yamlu
```

```
Get list of tests from:
- Provider:  Excel@1 #DataMart_Smoke
  Connection String: C:\CAT_all\BigBangExcelTests.xlsx
  Query:  SELECT * FROM DataMart_Smoke
  Name :  excel DataMart_Smoke

- Provider:  SqlServer@1
  Connection String: Data Source=sql-bigbang-dev.database.windows.net;Uid=O...;Pwd=...;Initial Catalog=sqldb-bigbang-dev
  Query:  EXEC [Test].[CAT_Smoke_GetAllEmptyTables_Staging]

- Provider:  SqlServer@1
  Connection String: Data Source=sql-bigbang-dev.database.windows.net;Uid=...;Pwd=...;Initial Catalog=sqldb-bigbang-dev
  Query:  SELECT * FROM [Test].[CAT_TestDefinition]

-
  Provider: Yaml@1
  Connection string: ./projectfile.cat.yaml
  Query: Tests
  Name: Testy v yamlu
```

Output:

We can define trx, yaml, json test result , see last printscreen

Output: trx, yaml, json # write it in the end of projectfile

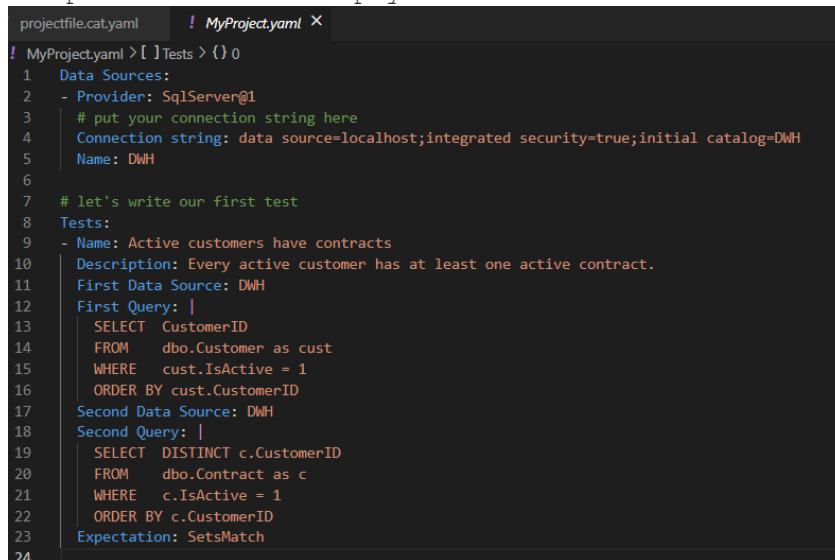
It will generate the file in the TestResults folder next to your project file.

Define tests directly in project file.

```
- Provider: SqlServer@1
  # put your connection string here
  Connection string: data source=localhost;integrated security=true;initial
catalog=DWH
  Name: DWH # Data Sources
```

Tests:

```
- Name: Len of Phone_number must be 15
  Order: 1
  Description: Len of Phone_number must be 15 numbers
  First Data Source: DWH # name of data source
  First Query: Select phone_number from [Phones$] where LEN(phone_number) <>
15
  Expectation: SetIsEmpty
```



```
projectfile.cat.yaml  ! MyProject.yaml X
! MyProject.yaml >[ ] Tests > {} 0
1  Data Sources:
2  - Provider: SqlServer@1
3    # put your connection string here
4    Connection string: data source=localhost;integrated security=true;initial catalog=DWH
5    Name: DWH
6
7  # let's write our first test
8  Tests:
9  - Name: Active customers have contracts
10     Description: Every active customer has at least one active contract.
11     First Data Source: DWH
12     First Query: |
13       SELECT CustomerID
14       FROM   dbo.Customer as cust
15       WHERE  cust.IsActive = 1
16       ORDER BY cust.CustomerID
17     Second Data Source: DWH
18     Second Query: |
19       SELECT DISTINCT c.CustomerID
20       FROM   dbo.Contract as c
21       WHERE  c.IsActive = 1
22       ORDER BY c.CustomerID
23     Expectation: SetsMatch
24
```

6.How to define tests

TestSuite #optional

Order #optional

TestCase #optional

TestName #must be define

Description #optional

FirstDataSource #must be define

FirstQuery #must be define

SecondDataSource # Must be define for integration tests, with Expectation SetsMatch

SecondQuery # Must be define for integration tests, with Expectation SetsMatch

Expectation #must be define

Categories #optional

Tolerance #optional

Timeout #optional

Expectation:

SetIsEmpty #Expectation is zero rows as a test result

SetsMatch # Expectation is the same results for both queries

SetRowCount # We can define the number of expected row count

Example of project File :

```
! projectfile.catyaml X
! projectfile.catyaml > [ ] Tests > { } 0 > [ ] Name
1  Get List of Data Sources From:
2
3  - Provider: Excel@1
4    Connection String: C:\CAT_all\BigBangExcelTests.xlsx
5    Query: SELECT * FROM Connections
6    Name: BigBang
7  -
8    Provider: SqlServer@1
9    Connection String: Data Source=sql-bigbang-dev.database.windows.net;Uid=Craftsman;Pwd=1qaz!@WSX;Initial Catalog=sqldb-bigbang-dev
10   Query: SELECT * FROM [Test].[CAT_ConnectionDefinition]
11
12
13  Get list of tests from:
14
15  - Provider: Excel@1 #DataMart_Smoke
16    Connection String: C:\CAT_all\BigBangExcelTests.xlsx
17    Query: SELECT * FROM DataMart_Smoke
18    Name : excel DataMart_Smoke
19
20  - Provider: SqlServer@1
21    Connection String: Data Source=sql-bigbang-dev.database.windows.net;Uid=Craftsman;Pwd=1qaz!@WSX;Initial Catalog=sqldb-bigbang-dev
22    Query: EXEC [Test].[CAT_Smoke_GetAllEmptyTables_Staging]
23
24  - Provider: SqlServer@1
25    Connection String: Data Source=sql-bigbang-dev.database.windows.net;Uid=Craftsman;Pwd=1qaz!@WSX;Initial Catalog=sqldb-bigbang-dev
26    Query: SELECT * FROM [Test].[CAT_TestDefinition]
27
28  -
29    Provider: Yaml@1
30    Connection string: ./projectfile.cat.yaml
31    Query: Tests
32    Name: Testy v yamlu
33
34  Tests:
35  - Name: tests
36    Description: test
37    First Data Source: BigBang
38    First Query: |
39      SELECT *
40      FROM [Fact].[Person Capacity]
41      where [Working Days] IS NULL or [Working Hours] IS NULL
42    Expectation: SetIsEmpty
43
44  Output: yaml
45
```