

React Uygulaması Otomasyon Testleri

VeriTabanı Testi

<code>\${DBConnect}</code>	–	Bağlanılacak veri tabanı
<code>\${DBName}</code>	–	DataBase ismi
<code>\${DBUser}</code>	–	DataBase Kullanıcı Adı
<code>\${DBPass}</code>	–	DataBase Parolası
<code>\${DBHost}</code>	–	Veritabanı Host
<code>\${DBPort}</code>	–	Veritabanı Port

```
1  *** Settings ***
2  Library      DatabaseLibrary
3  Library      OperatingSystem
4
5
6  Suite Setup   Connect To Database  ${DBConnect}  ${DBName}  ${DBUser}  ${DBPass}  ${DBHost}  ${DBPort}
7  Suite Teardown  Disconnect From Database
8
9  *** Variables ***
10 ${DBConnect}  pyodbc
11 ${DBName}    NORTHWIND
12 ${DBUser}    sa
13 ${DBPass}    12341234
14 ${DBHost}    localhost
15 ${DBPort}    49820
16
17 *** Test Cases ***
18 Create Test table
19
20 ${output}=    Execute Sql String  Create table dbo.RobotF(id integer,name varchar(20),type varchar(20))
21 Log To Console  ${output}
22 Should Be Equal As Strings  ${output}  None
23
24 Inserting Data in Test table
25 ${output}    Execute Sql String  INSERT INTO Categories(categoryName, description, picture) VALUES ('Test', 'Çalıştı', 'null')
26 Log to console  ${output}
27 Should Be Equal As Strings  ${output}  None
28
29
30 Check Test record present in Categories Table
31 Check If Exists In Database  SELECT CategoryID FROM Categories WHERE categoryName = 'Test';
32
33
34 Check Test1 record not present in dbo.Categories Table
35 Check If Not Exists In Database  SELECT CategoryID FROM Categories WHERE categoryName = 'Test1';
36
37
38 Verify Row Count is Equal to X
39 Row Count is Equal to X  SELECT CategoryID FROM Categories;  30
40
```

Bu otomasyon testinde 8 adet test senaryomuz bulunmaktadır. React uygulamamızın veritabanıyla bağlantısı ve beklenen tüm durumlar test edilir.

İlk case'de bir tablo oluşturulur ve değişkenleri belirlenir.

İkinci case'de ise bu tabloya veriler eklenir.

Diğer iki case'de sorgulanan verinin tabloda bulunup bulunmadığı test edilir.

İlk senaryoda verilen 'Test' satırının sorguladığımız '2' değerinden fazla bulunup bulunmadığı test edilir.

Diğer case'de tablodaki verilerin güncelleme işlemi yapılır. Aynı zamanda istenilen tabloları silmemiz de mümkündür.

```
41
42 ► Verify Row Count is Greater to Some Value
43     Row Count Is Greater Than X      SELECT * FROM dbo.Categories WHERE CategoryName= 'Test';    2
44
45
46 ► Update record in Categories table
47     ${output}= Execute Sql String    Update dbo.Categories set CategoryName='Robot' where CategoryID=11;
48     Log To Console    ${output}
49     Should Be Equal As Strings    ${output}    None
50
51 ► Verify that you can Delete a Table
52     ${output}= Execute SQL String    DROP TABLE Testing;
53     Should Be Equal As Strings    ${output}    None
54
55
56 *** Keywords ***
```

Report.html

Test Execution Log

REPORT

- [SUITE] AspNet			00:00:00.075
Full Name: AspNet			
Source: C:\Users\IPC_4077\PycharmProjects\pythonProject1\Automation\AspNet.robot			
Start / End / Elapsed: 20230513 10:08:42.748 / 20230513 10:08:42.823 / 00:00:00.075			
Status: 8 tests total, 8 passed, 0 failed, 0 skipped			
- [SETUP] DatabaseLibrary Connect To Database \${DBConnect}, \${DBName}, \${DBUser}, \${DBPass}, \${DBHost}, \${DBPort}			00:00:00.023
Documentation: Loads the DB API 2.0 module given 'dbapiModuleName' then uses it to connect to the database using 'dbName', 'dbUsername', and 'dbPassword'			
Start / End / Elapsed: 20230513 10:08:42.777 / 20230513 10:08:42.800 / 00:00:00.023			
10:08:42.788	[INFO]	Connecting using 'pyodbc.connect(DRIVER={%q; Server=\${SERVER%localhost,49829;DATA SOURCE=Microsoft%CD;UID=%u;PWD=%d2341234})'	
- [TEARDOWN] DatabaseLibrary Disconnect From Database			00:00:00.000
Documentation: Disconnects from the database			
Start / End / Elapsed: 20230513 10:08:42.823 / 20230513 10:08:42.823 / 00:00:00.000			
10:08:42.823	[INFO]	Executing : Disconnect From database	
- [TEST] Create Test table			00:00:00.009
Full Name: AspNet Create Test table			
Start / End / Elapsed: 20230513 10:08:42.800 / 20230513 10:08:42.809 / 00:00:00.009			
Status: PASS			
- [KEYWORD] DatabaseLibrary Execute Sql String Create table dbo.RobotFramework(Id Integer,name varchar(20),type varchar(20))			00:00:00.007
Documentation: Executes the sqlString as SQL commands. Useful to pass arguments to your sql. Set optional input 'sansTran' to True to run command without an explicit transaction commit or rollback.			
Start / End / Elapsed: 20230513 10:08:42.801 / 20230513 10:08:42.808 / 00:00:00.007			
10:08:42.808	[INFO]	Executing : Execute sql String create table dbo.RobotFramework(Id Integer,name varchar(20),type varchar(20))	
10:08:42.808	[INFO]	\$(output) = None	
- [KEYWORD] BuiltIn Log To Console \${output}			00:00:00.000
Documentation: Logs the given message to the console			
Start / End / Elapsed: 20230513 10:08:42.808 / 20230513 10:08:42.808 / 00:00:00.000			
- [KEYWORD] BuiltIn Should Be Equal As Strings \${output}, None			00:00:00.001
Documentation: Fails if objects are unequal after converting them to strings.			
Start / End / Elapsed: 20230513 10:08:42.808 / 20230513 10:08:42.809 / 00:00:00.001			
10:08:42.809	[INFO]	Argument types are: <class 'NoneType'> <class 'str'>	
- [TEST] Inserting Data in Test table			00:00:00.002
Full Name: AspNet Inserting Data in Test table			
Start / End / Elapsed: 20230513 10:08:42.810 / 20230513 10:08:42.812 / 00:00:00.002			
Status: PASS			
- [KEYWORD] DatabaseLibrary Execute Sql String INSERT INTO Categories(categoryName, description, picture) VALUES ('Test', 'Caligi', 'null')			00:00:00.001
Documentation: Executes the sqlString as SQL commands. Useful to pass arguments to your sql. Set optional input 'sansTran' to True to run command without an explicit transaction commit or rollback.			
Start / End / Elapsed: 20230513 10:08:42.810 / 20230513 10:08:42.811 / 00:00:00.001			
10:08:42.810	[INFO]	Executing : Execute sql String insert into Categories(categoryName, description, picture) VALUES ('Test', 'Caligi', 'null')	
10:08:42.811	[INFO]	\$(output) = None	
- [KEYWORD] BuiltIn Log To Console \${output}			00:00:00.001
Documentation: Logs the given message to the console			
Start / End / Elapsed: 20230513 10:08:42.811 / 20230513 10:08:42.812 / 00:00:00.001			
- [KEYWORD] BuiltIn Should Be Equal As Strings \${output}, None			00:00:00.000
Documentation: Fails if objects are unequal after converting them to strings.			
Start / End / Elapsed: 20230513 10:08:42.812 / 20230513 10:08:42.812 / 00:00:00.000			
- [TEST] Check Test record present in Categories Table			00:00:00.000
Full Name: AspNet Check Test record present in Categories Table			
Start / End / Elapsed: 20230513 10:08:42.812 / 20230513 10:08:42.813 / 00:00:00.001			
Status: PASS			
- [KEYWORD] DatabaseLibrary Check If Exists In Database SELECT CategoryID FROM Categories WHERE categoryName = 'Test'			00:00:00.000
Documentation: Check if any row could be returned by given the input 'selectStatement'. If there are no results, then this will throw an AssertionError. Set optional input 'sansTran' to True to run command without an explicit transaction commit or rollback.			
Start / End / Elapsed: 20230513 10:08:42.813 / 20230513 10:08:42.813 / 00:00:00.000			
10:08:42.813	[INFO]	Executing : Check If Exists In Database SELECT CategoryID FROM Categories WHERE categoryName = 'Test';	
10:08:42.813	[INFO]	Executing : Query SELECT CategoryID FROM Categories WHERE categoryName = 'Test';	
- [TEST] Check Test1 record not present in dbo.Categories Table			00:00:00.000
Full Name: AspNet Check Test1 record not present in dbo.Categories Table			
Start / End / Elapsed: 20230513 10:08:42.814 / 20230513 10:08:42.814 / 00:00:00.000			
Status: PASS			
- [KEYWORD] DatabaseLibrary Check If Not Exists In Database SELECT categoryID FROM Categories WHERE categoryName = 'Test1'			00:00:00.000
Documentation: This is the negation of 'check_if_exists_in_database'			
Start / End / Elapsed: 20230513 10:08:42.814 / 20230513 10:08:42.814 / 00:00:00.000			
10:08:42.814	[INFO]	Executing : Check If Not Exists In Database SELECT CategoryID FROM Categories WHERE categoryName = 'Test1';	
10:08:42.814	[INFO]	Executing : Query SELECT categoryID FROM Categories WHERE categoryName = 'Test1';	
- [TEST] Verify Row Count is Equal to X			00:00:00.000
Full Name: AspNet Verify Row Count is Equal to X			
Start / End / Elapsed: 20230513 10:08:42.815 / 20230513 10:08:42.815 / 00:00:00.000			
Status: PASS			
- [KEYWORD] DatabaseLibrary Row Count is Equal To X SELECT CategoryID FROM Categories, 32			00:00:00.000
Documentation: Check if the number of rows returned from 'selectStatement' is equal to the value submitted. If not, then this will throw an AssertionError. Set optional input 'sansTran' to True to run command without an explicit transaction commit or rollback.			
Start / End / Elapsed: 20230513 10:08:42.815 / 20230513 10:08:42.815 / 00:00:00.000			
10:08:42.815	[INFO]	Executing : Row count is equal To X SELECT CategoryID FROM Categories; 32	
10:08:42.815	[INFO]	Executing : Row Count SELECT CategoryID FROM Categories;	
- [TEST] Verify Row Count is Greater to Some Value			00:00:00.001
Full Name: AspNet Verify Row Count is Greater to Some Value			
Start / End / Elapsed: 20230513 10:08:42.816 / 20230513 10:08:42.817 / 00:00:00.001			
Status: PASS			
- [KEYWORD] DatabaseLibrary Row Count is Greater Than X SELECT " FROM dbo.Categories WHERE CategoryName= 'Test', 2			00:00:00.000
Documentation: Check if the number of rows returned from 'selectStatement' is greater than the value submitted. If not, then this will throw an AssertionError. Set optional input 'sansTran' to True to run command without an explicit transaction commit or rollback.			
Start / End / Elapsed: 20230513 10:08:42.816 / 20230513 10:08:42.816 / 00:00:00.000			
10:08:42.816	[INFO]	Executing : Row count is greater than X SELECT * FROM dbo.Categories WHERE categoryName= 'Test'; 2	
10:08:42.816	[INFO]	Executing : Row Count SELECT * FROM dbo.Categories WHERE categoryName= 'Test';	
- [TEST] Update record in Categories table			00:00:00.001
Full Name: AspNet Update record in Categories table			
Start / End / Elapsed: 20230513 10:08:42.817 / 20230513 10:08:42.818 / 00:00:00.001			
Status: PASS			
- [KEYWORD] DatabaseLibrary Execute Sql String Update dbo.Categories set CategoryName='Robot' where CategoryID=11.			00:00:00.001
Documentation: Executes the sqlString as SQL commands. Useful to pass arguments to your sql. Set optional input 'sansTran' to True to run command without an explicit transaction commit or rollback.			
Start / End / Elapsed: 20230513 10:08:42.817 / 20230513 10:08:42.818 / 00:00:00.001			
10:08:42.817	[INFO]	Executing : Execute SQL String update dbo.Categories set CategoryName='Robot' where CategoryID=11;	
10:08:42.818	[INFO]	\$(output) = None	
- [KEYWORD] BuiltIn Log To Console \${output}			00:00:00.000
Documentation: Logs the given message to the console			
Start / End / Elapsed: 20230513 10:08:42.818 / 20230513 10:08:42.818 / 00:00:00.000			
- [KEYWORD] BuiltIn Should Be Equal As Strings \${output}, None			00:00:00.000
Documentation: Fails if objects are unequal after converting them to strings.			
Start / End / Elapsed: 20230513 10:08:42.818 / 20230513 10:08:42.818 / 00:00:00.000			
10:08:42.818	[INFO]	Argument types are: <class 'NoneType'> <class 'str'>	
- [TEST] Verify that you can Delete a Table			00:00:00.003
Full Name: AspNet Verify that you can Delete a Table			
Start / End / Elapsed: 20230513 10:08:42.819 / 20230513 10:08:42.822 / 00:00:00.003			
Status: PASS			
- [KEYWORD] DatabaseLibrary Execute Sql String DROP TABLE Robot.			00:00:00.002
Documentation: Executes the sqlString as SQL commands. Useful to pass arguments to your sql. Set optional input 'sansTran' to True to run command without an explicit transaction commit or rollback.			
Start / End / Elapsed: 20230513 10:08:42.820 / 20230513 10:08:42.822 / 00:00:00.002			
10:08:42.820	[INFO]	Executing : Execute SQL String DROP TABLE Robot;	
10:08:42.822	[INFO]	\$(output) = None	
- [KEYWORD] BuiltIn Should Be Equal As Strings \${output}, None			00:00:00.000
Documentation: Fails if objects are unequal after converting them to strings.			
Start / End / Elapsed: 20230513 10:08:42.822 / 20230513 10:08:42.822 / 00:00:00.000			
10:08:42.822	[INFO]	Argument types are: <class 'NoneType'> <class 'str'>	

UJ

```
{ "categoryId":3,"categoryName":"Confections","description":"Desserts, candies, and sweet  
breads","picture":"FRwvAAIAAANA4AFAAhAP////9CaXRtYXAgSW1hZ2UAUGFpbnQuUGljdHVyZQABBQAAAgAAAAcAAA  
[] }  
{ "categoryId":4,"categoryName":"Dairy  
Products","description":"Cheeses","picture":"FRwvAAIAAANA4AFAAhAP////9CaXRtYXAgSW1hZ2UAUGFpbnQuUGljdHVy  
[] }  
{ "categoryId":5,"categoryName":"Grains/Cereals","description":"Breads, crackers, pasta, and  
cereal","picture":"FRwvAAIAAANA4AFAAhAP////9CaXRtYXAgSW1hZ2UAUGFpbnQuUGljdHVyZQABBQAAAgAAAAcAAAB  
[] }  
{ "categoryId":6,"categoryName":"Meat/Poultry","description":"Prepared  
meats","picture":"FRwvAAIAAANA4AFAAhAP////9CaXRtYXAgSW1hZ2UAUGFpbnQuUGljdHVyZQABBQAAAgAAAAcAAAE  
[] }  
{ "categoryId":7,"categoryName":"Produce","description":"Dried fruit and bean  
curd","picture":"FRwvAAIAAANA4AFAAhAP////9CaXRtYXAgSW1hZ2UAUGFpbnQuUGljdHVyZQABBQAAAgAAAAcAAABC  
[] }  
{ "categoryId":8,"categoryName":"Seafood","description":"Seaweed and  
fish","picture":"FRwvAAIAAANA4AFAAhAP////9CaXRtYXAgSW1hZ2UAUGFpbnQuUGljdHVyZQABBQAAAgAAAAcAAABQ  
[] }  
{ "categoryId":9,"categoryName":"hilal","description":null,"picture":null,"products":[] }  
{ "categoryId":10,"categoryName":"software","description":"testing","picture":null,"products":[] }  
{ "categoryId":11,"categoryName":"Robot","description":"solmaz","picture":null,"products":[] }  
{ "categoryId":12,"categoryName":"Test","description":"Çalıştı","picture":"bnVsbA==","products":[] }  
{ "categoryId":13,"categoryName":"Test","description":"Çalıştı","picture":"bnVsbA==","products":[] }  
{ "categoryId":14,"categoryName":"Test","description":"Çalıştı","picture":"bnVsbA==","products":[] }  
{ "categoryId":15,"categoryName":"Test","description":"Çalıştı","picture":"bnVsbA==","products":[] }  
{ "categoryId":16,"categoryName":"Test","description":"Çalıştı","picture":"bnVsbA==","products":[] }  
{ "categoryId":17,"categoryName":"Test","description":"Çalıştı","picture":"bnVsbA==","products":[] }  
{ "categoryId":18,"categoryName":"Test","description":"Çalıştı","picture":"bnVsbA==","products":[] }  
{ "categoryId":19,"categoryName":"Test","description":"Çalıştı","picture":"bnVsbA==","products":[] }  
{ "categoryId":20,"categoryName":"Test","description":"Çalıştı","picture":"bnVsbA==","products":[] }  
{ "categoryId":21,"categoryName":"Yazılım Test","description":null,"picture":null,"products":[] }  
{ "categoryId":22,"categoryName":"Yazılım Test","description":"Mühendislik","picture":null,"products":[] }  
{ "categoryId":23,"categoryName":"Yazılım Test","description":"Mühendislik","picture":null,"products":[] }  
{ "categoryId":24,"categoryName":"Yazılım Test","description":"Mühendislik","picture":null,"products":[] }  
{ "categoryId":25,"categoryName":"Yazılım Test","description":"Mühendislik","picture":null,"products":[] }  
{ "categoryId":26,"categoryName":"Yazılım Test","description":"Mühendislik","picture":null,"products":[] }  
{ "categoryId":27,"categoryName":"Yazılım Test","description":"Mühendislik","picture":null,"products":[] }  
{ "categoryId":28,"categoryName":"Yazılım Test","description":"Mühendislik","picture":null,"products":[] }  
{ "categoryId":29,"categoryName":"Yazılım Test","description":"Mühendislik","picture":null,"products":[] }
```

Results		Messages		
	CategoryID	CategoryName	Description	Picture
11	11	Robot	solmaz	NULL
12	12	Test	Çalıştı	0x6E756C6C
13	13	Test	Çalıştı	0x6E756C6C
14	14	Test	Çalıştı	0x6E756C6C
15	15	Test	Çalıştı	0x6E756C6C
16	16	Test	Çalıştı	0x6E756C6C
17	17	Test	Çalıştı	0x6E756C6C
18	18	Test	Çalıştı	0x6E756C6C
19	19	Test	Çalıştı	0x6E756C6C
20	20	Test	Çalıştı	0x6E756C6C
21	21	Yazılım Test	NULL	NULL
22	22	Yazılım Test	Mühendislik	NULL
23	23	Yazılım Test	Mühendislik	NULL
24	24	Yazılım Test	Mühendislik	NULL
25	25	Yazılım Test	Mühendislik	NULL
26	26	Yazılım Test	Mühendislik	NULL
27	27	Yazılım Test	Mühendislik	NULL
28	28	Yazılım Test	Mühendislik	NULL
29	29	Yazılım Test	Mühendislik	NULL
✓ Query executed successfully.				