

Connectivity - PowerShell script and code samples

Last updated by | Keith Elmore | Aug 5, 2020 at 1:39 PM PDT

Contents

- Scope :
 - PowerShell script for connectivity test - includes AAD and ...
 - PowerShell script for connectivity test in loop with 5 secon...
 - C# code for sample console application using both SQL au...
 - Java code for sample console application using SQL auth.

Scope :

Connectivity test script, loop test script with time delay for automated connectivity test in observation and console application sample code.

PowerShell script for connectivity test - includes AAD and SQL auth |

```
# Provide server name, initial DB, user and password

$SqlConnection = New-Object System.Data.SqlClient.SqlConnection

# ----- Using SQL Auth ----- uncomment to use
# $SqlConnection.ConnectionString = "Server=tcp:tapanm.database.windows.net;Initial Catalog=HelloWorld;User ID

# ----- Using AAD auth ----- uncomment to use
# $SqlConnection.ConnectionString = "Server=tcp:tapanm.database.windows.net;Initial Catalog=HelloWorld;Authent

$SqlConnection.Open()

# If no errors, connection succeeded

# Now run a sample query - modify query as you like

$SqlCommand = New-Object System.Data.SqlClient.SqlCommand

#Modify below query

$SqlCommand.CommandText = "SELECT count(*) from saleslt.customer;"
$SqlCommand.Connection = $SqlConnection
$SqlAdapter = New-Object System.Data.SqlClient.SqlDataAdapter
$SqlAdapter.SelectCommand = $SqlCommand
$DataSet = New-Object System.Data.DataSet
$SqlAdapter.Fill($DataSet)
$dataset.Tables[0]
```

PowerShell script for connectivity test in loop with 5 second interval of infinite time

```

# Infinite loop for connectivity with currently sleep set to 5 seconds, which can be changed as required.
# Update connection string with your server name, database name, user ID and password, or use AAD authenticati
# Default output is both on PowerShell script execution console for timestamp and in the Output.txt text file
# Change query the way you'd like it to return content. Sample just counts rows in SalesLT.Customer in default
$i = 1;
do
{
$SqlConnection = New-Object System.Data.SqlClient.SqlConnection
$SqlConnection.ConnectionString = "Server=tcp:tapanm.database.windows.net;Initial Catalog=HelloWorld;User ID=t
# Uncomment below for using AAD Integrated auth and comment SQL auth above as required.
#$SqlConnection.ConnectionString = "Server=tcp:tapanm.database.windows.net;Initial Catalog=HelloWorldDB;Authen
$SqlConnection.Open()
$SqlCommand = New-Object System.Data.SqlClient.SqlCommand
# Update query as you like below
$SqlCommand.CommandText = "SELECT count(*) from saleslt.customer;"
$SqlCommand.Connection = $SqlConnection
$SqlAdapter = New-Object System.Data.SqlClient.SqlDataAdapter
$SqlAdapter.SelectCommand = $SqlCommand
$DataSet = New-Object System.Data.DataSet
$SqlAdapter.Fill($DataSet)
#Update output file location as you like below
$dataset.Tables[0] | Out-File .\output.txt -Append
$time = Get-Date
$time.ToUniversalTime()
#Update output file location as you like below
$time | Out-File .\output.txt -Append
$SqlAdapter.Dispose()
$SqlConnection.Close()
$SqlConnection.Dispose()
# Update sleep interval in seconds as you like below
sleep 5
}
while ($i -eq 1);

```

C# code for sample console application using both SQL auth and AAD auth

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Data;
using System.Data.Sql;
using System.Data.SqlClient;

namespace ConsoleApplication5
{
    class Program
    {
        static void Main(string[] args)
        {
            SqlConnection cn;
            string strCn;
            SqlCommand cmd;
            SqlParameter prm;
            //Comment or uncomment the SQL auth or Azure AD integrated auth as you like for switching between
            //String using SQL auth
            //strCn = "Server=tcp:tapanm.database.windows.net,1433;Initial Catalog=HelloWorldDB;Persist Security
            //String using Azure AD integrated auth
            strCn = "Server=tcp:tapanm.database.windows.net,1433;Initial Catalog=HelloWorldDB;Persist Security
            cn = new SqlConnection(strCn);
            cmd = new SqlCommand("getcustomer2", cn);
            cmd.CommandType = (CommandType.StoredProcedure);
            cn.Open();
            SqlDataReader dr = cmd.ExecuteReader();
            //SqlDataAdapter da = new SqlDataAdapter(cmd);
            //DataTable dt = new DataTable();
            //da.Fill(dt);
            if (dr.HasRows)
            {
                while (dr.Read())
                {
                    Console.WriteLine("{0}",
                        dr.GetString(0));
                }
            }
            else
            {
                Console.WriteLine("No rows found.");
            }
            Console.ReadKey();
            dr.Close();
            cn.Close();
        }
    }
}

```

Java code for sample console application using SQL auth.

```

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

/**
 *
 * @author josvil
 */
public class MySQLDatabaseConnectionTest {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {

        String hostName = "<server>.database.windows.net";
        String dbName = "<catalog>";
        String user = "<user>";
        String password = "<password>!";
        String url = String.format("jdbc:sqlserver://%s:1433;database=%s;user=%s;password=%s;encrypt=true;trus

        Connection connection = null;

        try{
            Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
            connection = DriverManager.getConnection(url);
            String schema = connection.getSchema();

            System.out.println("Successful connection - Schema: " + schema);
            System.out.println("Query data example:");

            String selectSql = "<T-SQL statement here>";

            try (Statement statement = connection.createStatement();

                ResultSet resultSet = statement.executeQuery(selectSql)) {

                // Print results from select statement
                System.out.println("Result:");

                while (resultSet.next())
                {
                    System.out.println(resultSet.getString(1));
                }

                connection.close();
            }
        } catch (Exception ex)
        {
            ex.printStackTrace();
        }
    }
}

```

How good have you found this content?

