

عنوان: اتصال Node.js به SQL Server با استفاده از Edge.js

نویسنده: وحید محمدطاهری

تاریخ: ۱۳۹۴/۰۵/۰۹

آدرس: www.dotnettips.info

گروه‌ها: SQL Server, node.js, Edge.js

اگر خواسته باشید که با استفاده از [Node.js](#) به SQL Server متصل شوید، احتمالاً متوجه شده‌اید [ماژولی](#) که میکروسافت منتشر کرده است، ناقص بوده و به صورت پیش نمایش است که بسیاری از ویژگی‌ها و مسائل مهم، در آن در نظر گرفته نشده است.

یکی دیگر از ماژول‌هایی که امکان اتصال Node.js را به SQL Server ممکن می‌کند، [Edge.js](#) است. Edge.js یک ماژول Node.js است که امکان اجرای کدهای دات نت را در همان پروسه توسط Node.js فراهم می‌کند. این مسئله، توسعه دهندگان Node.js را قادر می‌سازد تا از فناوری‌هایی که به صورت سنتی استفاده‌ای از آنها سخت یا غیر ممکن بوده است را به راحتی استفاده کنند. برای نمونه:

SQL Server

Active Directory

Nuget packages

استفاده از سخت افزار کامپیوتر (مانند وب کم، میکروفن و چاپگر)

نصب Node.js

اگر Node.js را بر روی سیستم خود نصب ندارید، می‌توانید از [اینجا](#) آن را دانلود کنید. بعد از نصب برای اطمینان از کارکرد آن، command prompt را باز کرده و دستور زیر را تایپ کنید:

```
node -v
```

شما باید نسخه‌ی نصب شده‌ی Node.js را مشاهده کنید.

ایجاد پوشه پروژه

سپس پوشه‌ای را برای پروژه Node.js خود ایجاد کنید. مثلاً با استفاده از command prompt و دستور زیر:

```
md \projects\node-edge-test1
cd \projects\node-edge-test1
```

نصب Edge.js

Node با استفاده از package manager خود دانلود و نصب ماژول‌ها را خیلی آسان کرده است. برای نصب، در command prompt عبارت زیر را تایپ کنید:

```
npm install edge
npm install edge-sql
```

فرمان اول باعث نصب Edge.js و دومین فرمان سبب نصب پشتیبانی از SQL Server می‌شود.

Hello World ایجاد یک فایل متنی با نام **server.js** و نوشتن کد زیر در آن:

```
var edge = require('edge');

// The text in edge.func() is C# code
var helloWorld = edge.func('async (input) => { return input.ToString(); }');

helloWorld('Hello World!', function (error, result) {
    if (error) throw error;
    console.log(result);
});
```

حالا برای اجرای این Node.js application از طریق command prompt کافی است به صورت زیر عمل کنید:

```
node server.js
```

همانطور که مشاهده می‌کنید "Hello World!" در خروجی چاپ شد.

ایجاد پایگاه داده تست

در مثال‌های بعدی، نیاز به یک پایگاه داده داریم تا queryها را اجرا کنیم. در صورتی که SQL Server بر روی سیستم شما نصب نیست، می‌توانید نسخه‌ی رایگان آن را از [اینجا](#) دانلود و نصب کنید. همچنین SQL Management Studio Express را نیز نصب کنید.

در SQL Management Studio، یک پایگاه داده را با نام node-test با تنظیمات پیش فرض ایجاد کنید.

بر روی پایگاه داده node-test راست کلیک کرده و **New Query** را انتخاب کنید.

اسکرپت زیر را copy کرده و در آنجا paste کنید، سپس بر روی **Execute** کلیک کنید.

```
IF EXISTS(SELECT 1 FROM sys.tables WHERE object_id = OBJECT_ID('SampleUsers')) BEGIN; DROP TABLE SampleUsers; END; GO

CREATE TABLE SampleUsers ( Id INTEGER NOT NULL IDENTITY(1, 1), FirstName VARCHAR(255) NOT NULL, LastName VARCHAR(255) NOT NULL, Email VARCHAR(255) NOT NULL, CreateDate DATETIME NOT NULL DEFAULT(getdate()), PRIMARY KEY (Id) ); GO

INSERT INTO SampleUsers(FirstName,LastName,Email,CreateDate)
VALUES('Orla','Sweeney','nunc@convallisincursus.ca','Apr 13, 2014');
INSERT INTO SampleUsers(FirstName,LastName,Email,CreateDate)
VALUES('Zia','Pickett','porttitor.tellus.non@Duis.com','Aug 31, 2014');
INSERT INTO SampleUsers(FirstName,LastName,Email,CreateDate)
VALUES('Justina','Ayala','neque.tellus.imperdiet@temporestac.com','Jul 28, 2014');
INSERT INTO SampleUsers(FirstName,LastName,Email,CreateDate)
VALUES('Levi','Parrish','adipiscing.elit@velarcueu.com','Jun 21, 2014');
INSERT INTO SampleUsers(FirstName,LastName,Email,CreateDate)
VALUES('Pearl','Warren','In@dignissimpharetra.org','Mar 3, 2014');
```

نتیجه‌ی اجرای کد بالا، ایجاد جدولی با نام SampleUsers و درج 5 رکورد در آن می‌شود.

تنظیمات ConnectionString

قبل از استفاده از Edge.js با SQL Server، باید متغیر محیطی (environment variable) با نام **EDGE_SQL_CONNECTION_STRING** را تعریف کنید.

```
set EDGE_SQL_CONNECTION_STRING=Data Source=localhost;Initial Catalog=node-test;Integrated Security=True
```

این متغیر تنها برای command prompt جاری تعریف شده است و با بستن آن از دست می‌رود. در صورتیکه از [Node.js Tools for Visual Studio](#) استفاده می‌کنید، نیاز به ایجاد یک متغیر محیطی دائمی و راه اندازی مجدد VS دارید. همچنین در صورتیکه بخواهید متغیر محیطی دائمی ایجاد کنید، فرمان زیر را اجرا کنید:

```
SETX EDGE_SQL_CONNECTION_STRING "Data Source=localhost;Initial Catalog=node-test;Integrated Security=True"
```

روش اول: اجرای مستقیم SQL Server Query در Edge.js

فایلی با نام **server-sql-query.js** را ایجاد کرده و کد زیر را در آن وارد کنید:

```
var http = require('http');
var edge = require('edge');
var port = process.env.PORT || 8080;
```

```
var getTopUsers = edge.func('sql', function () {/*
    SELECT TOP 3 * FROM SampleUsers ORDER BY CreateDate DESC
*/});

function logError(err, res) {
    res.writeHead(200, { 'Content-Type': 'text/plain' });
    res.write("Error: " + err);
    res.end("");
}

http.createServer(function (req, res) {
    res.writeHead(200, { 'Content-Type': 'text/html' });

    getTopUsers(null, function (error, result) {
        if (error) { logError(error, res); return; }
        if (result) {
            res.write("<ul>");
            result.forEach(function(user) {
                res.write("<li>" + user.FirstName + " " + user.LastName + ": " + user.Email + "</li>");
            });
            res.end("</ul>");
        }
        else {
        }
    });
}).listen(port);
console.log("Node server listening on port " + port);
```

سپس با استفاده از command prompt، فرمان زیر را اجرا کنید:

```
node server-sql-query.js
```

حال مرورگر خود را باز و سپس آدرس **http://localhost:8080** را باز کنید. در صورتی که همه چیز به درستی انجام گرفته باشد لیستی از 3 کاربر را خواهید دید.

روش دوم: اجرای کد دات نت برای Edge.js SQL Server Query تنها از دستورات Delete، Insert، Select و Update پشتیبانی می‌کند. در حال حاضر از store procedures و مجموعه‌ای از کد SQL پشتیبانی نمی‌کند. بنابراین، اگر چیزی بیشتر از عملیات CRUD می‌خواهید انجام دهید، باید از دات نت برای این کار استفاده کنید.

یادتان باشد، همیشه async

مدل اجرایی Node.js به صورت یک حلقه‌ی رویداد تک نخی است. بنابراین این **بسیار مهم** است که کد دات نت شما به صورت async باشد. در غیر اینصورت یک فراخوانی به دات نت سبب مسدود شدن و ایجاد خرابی در Node.js می‌شود.

ایجاد یک Class Library

اولین قدم، ایجاد یک پروژه Class Library در Visual Studio که خروجی آن یک فایل DLL است و استفاده از آن در Edge.js است. پروژه Class Library با عنوان **EdgeSampleLibrary** ایجاد کرده و فایل کلاسی با نام **Sample1** را به آن اضافه کنید و سپس کد زیر را در آن وارد کنید:

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SqlClient;
using System.Threading.Tasks;

namespace EdgeSampleLibrary
{
    public class Sample1
    {
        public async Task<object> Invoke(object input)
        {
            // Edge marshalls data to .NET using an IDictionary<string, object>
            var payload = (IDictionary<string, object>) input;
            var pageNumber = (int) payload["pageNumber"];
            var pageSize = (int) payload["pageSize"];
            return await QueryUsers(pageNumber, pageSize);
        }
    }
}
```

```

    }

    public async Task<List<SampleUser>> QueryUsers(int pageNumber, int pageSize)
    {
        // Use the same connection string env variable
        var connectionString = Environment.GetEnvironmentVariable("EDGE_SQL_CONNECTION_STRING");
        if (connectionString == null)
            throw new ArgumentException("You must set the EDGE_SQL_CONNECTION_STRING environment
variable.");

        // Paging the result set using a common table expression (CTE).
        // You may rather do this in a stored procedure or use an
        // ORM that supports async.
        var sql = @"
DECLARE @RowStart int, @RowEnd int;
SET @RowStart = (@PageNumber - 1) * @PageSize + 1;
SET @RowEnd = @PageNumber * @PageSize;

WITH Paging AS
(
    SELECT ROW_NUMBER() OVER (ORDER BY CreateDate DESC) AS RowNum,
           Id, FirstName, LastName, Email, CreateDate
    FROM SampleUsers
)
SELECT Id, FirstName, LastName, Email, CreateDate
FROM Paging
WHERE RowNum BETWEEN @RowStart AND @RowEnd
ORDER BY RowNum;
";

        var users = new List<SampleUser>();

        using (var cnx = new SqlConnection(connectionString))
        {
            using (var cmd = new SqlCommand(sql, cnx))
            {
                await cnx.OpenAsync();

                cmd.Parameters.Add(new SqlParameter("@PageNumber", SqlDbType.Int) { Value =
pageNumber });
                cmd.Parameters.Add(new SqlParameter("@PageSize", SqlDbType.Int) { Value = pageSize
});

                using (var reader = await cmd.ExecuteReaderAsync(CommandBehavior.CloseConnection))
                {
                    while (await reader.ReadAsync())
                    {
                        var user = new SampleUser
                        {
                            Id = reader.GetInt32(0),
                            FirstName = reader.GetString(1),
                            LastName = reader.GetString(2),
                            Email = reader.GetString(3),
                            CreateDate = reader.GetDateTime(4)
                        };
                        users.Add(user);
                    }
                }
            }
        }

        return users;
    }
}

public class SampleUser
{
    public int Id { get; set; }
    public string FirstName { get; set; }
    public string LastName { get; set; }
    public string Email { get; set; }
    public DateTime CreateDate { get; set; }
}
}

```

سپس ذخیره و کامپایل کنید. فایل DLL خروجی که در مسیر

[project]/bin/Debug/EdgeSampleLibrary.dll

قرار دارد را در پوشه‌ی پروژه Node کپی کنید. فایل جدیدی را با نام **server-dotnet-query.js** در پروژه Node ایجاد کنید و کد زیر را در آن وارد کنید:

```
var http = require('http');
var edge = require('edge');
var port = process.env.PORT || 8080;

// Set up the assembly to call from Node.js
var querySample = edge.func({ assemblyFile: 'EdgeSampleLibrary.dll', typeName:
'EdgeSampleLibrary.Sample1', methodName: 'Invoke' });

function logError(err, res) { res.writeHead(200, { 'Content-Type': 'text/plain' }); res.write("Got
error: " + err); res.end(""); }

http.createServer(function (req, res) { res.writeHead(200, { 'Content-Type': 'text/html' });

    // This is the data we will pass to .NET
    var data = { pageNumber: 1, pageSize: 3 };

    // Invoke the .NET function
    querySample(data, function (error, result) {
        if (error) { logError(error, res); return; }
        if (result) {
            res.write("<ul>");
            result.forEach(function (user) {
                res.write("<li>" + user.FirstName + " " + user.LastName + ": " + user.Email + "</li>");
            });
            res.end("</ul>");
        }
        else {
            res.end("No results");
        }
    });
}).listen(port);

console.log("Node server listening on port " + port);
```

سپس از طریق command prompt آن را اجرا کنید:

```
node server-dotnet-query.js
```

حال مرورگر خود را باز کرده و به آدرس **http://localhost:8080** بروید. در صورتیکه همه چیز به درستی انجام گرفته باشد، لیستی از 3 کاربر را خواهید دید. مقادیر **pageNumber** و **pageSize** را در فایل جاوااسکریپت تغییر دهید و تاثیر آن را بر روی خروجی مشاهده کنید.

نکته: برای ایجاد **pageNumber** و **pageSize** داینامیک با استفاده از ارسال مقادیر توسط **QueryString**، می‌توانید از [ماژول connect](#) استفاده کنید.