

عنوان: RavenDB؛ تجربه متفاوت از پایگاه داده

نویسنده: سروش ترک زاده

تاریخ: ۱۹:۳۱ ۱۳۹۱/۰۴/۱۵


















آدرس: [www.dotnettips.info](http://www.dotnettips.info)

برچسب‌ها: C#, JSON, NoSQL, RavenDB

" به شما خواننده گرامی پیشنهاد می‌کنم [مطلب قبلی](#) را مطالعه کنید تا پیش زمینه مناسبی در باره این مطلب کسب کنید. "

ماهیت این پایگاه داده وب سرویسی مبتنی بر REST است و فرمت اطلاعاتی که از سرور دریافت می‌شود، [JSON](#) است.

**گام اول:** باید آخرین نسخه RavenDB را دریافت کنید. همان طور که مشاهده می‌کنید، ویرایش‌های مختلف کتابخانه‌هایی که برای نسخه Client و همچنین Server طراحی شده است، در این فایل قرار گرفته است.

Name	Date modified	Type	Size
 Backup	۲۰۱۲/۰۲/۰۶ ۰۴:۲۰ ...	File folder	
 Bundles	۲۰۱۲/۰۲/۰۶ ۰۴:۲۰ ...	File folder	
 Client	۲۰۱۲/۰۲/۰۶ ۰۴:۲۰ ...	File folder	
 Client-3.5	۲۰۱۲/۰۲/۰۶ ۰۴:۲۰ ...	File folder	
 EmbeddedClient	۲۰۱۲/۰۲/۰۶ ۰۴:۲۰ ...	File folder	
 Samples	۲۰۱۲/۰۲/۰۶ ۰۴:۲۰ ...	File folder	
 Server	۲۰۱۲/۰۲/۰۶ ۰۴:۲۰ ...	File folder	
 Silverlight	۲۰۱۲/۰۲/۰۶ ۰۴:۲۰ ...	File folder	
 Silverlight-4	۲۰۱۲/۰۲/۰۶ ۰۴:۲۰ ...	File folder	
 Smuggler	۲۰۱۲/۰۲/۰۶ ۰۴:۲۰ ...	File folder	
 Web	۲۰۱۲/۰۲/۰۶ ۰۴:۲۰ ...	File folder	
 acknowledgments	۲۰۱۲/۰۲/۰۶ ۰۴:۱۸ ...	Text Document	
 license	۲۰۱۲/۰۲/۰۶ ۰۴:۱۸ ...	Text Document	
 Raven-GetBundles	۲۰۱۲/۰۲/۰۶ ۰۴:۱۸ ...	PS1 File	
 Raven-UpdateBundles	۲۰۱۲/۰۲/۰۶ ۰۴:۱۸ ...	PS1 File	
 readme	۲۰۱۲/۰۲/۰۶ ۰۴:۱۸ ...	Text Document	
 Start	۲۰۱۲/۰۲/۰۶ ۰۴:۱۸ ...	Windows Comma...	

برای راه اندازی Server باید فایل Start را اجرا کنید، چند ثانیه بعد محیط مدیریتی آن را در مرورگر خود مشاهده می‌کنید. در بالای صفحه روی لینک Databases کلیک کنید و در صفحه باز شده گزینه New Database را انتخاب کنید. با دادن یک نام دلخواه حالا شما یک پایگاه داده ایجاد کرده اید. تا همین جا دست نگه دارید و اجازه دهید با این محیط دوست داشتنی و قابلیت‌های آن بعداً آشنا شویم.

در گام دوم به Visual Studio می‌رویم و نحوه ارتباط با پایگاه داده و استفاده از دستورات آن را فرا می‌گیریم.

## گام دوم:

با یک پروژه Test شروع می‌کنیم که در هر گام تکمیل می‌شود و می‌توانید پروژه کامل را در پایان این پست دانلود کنید.

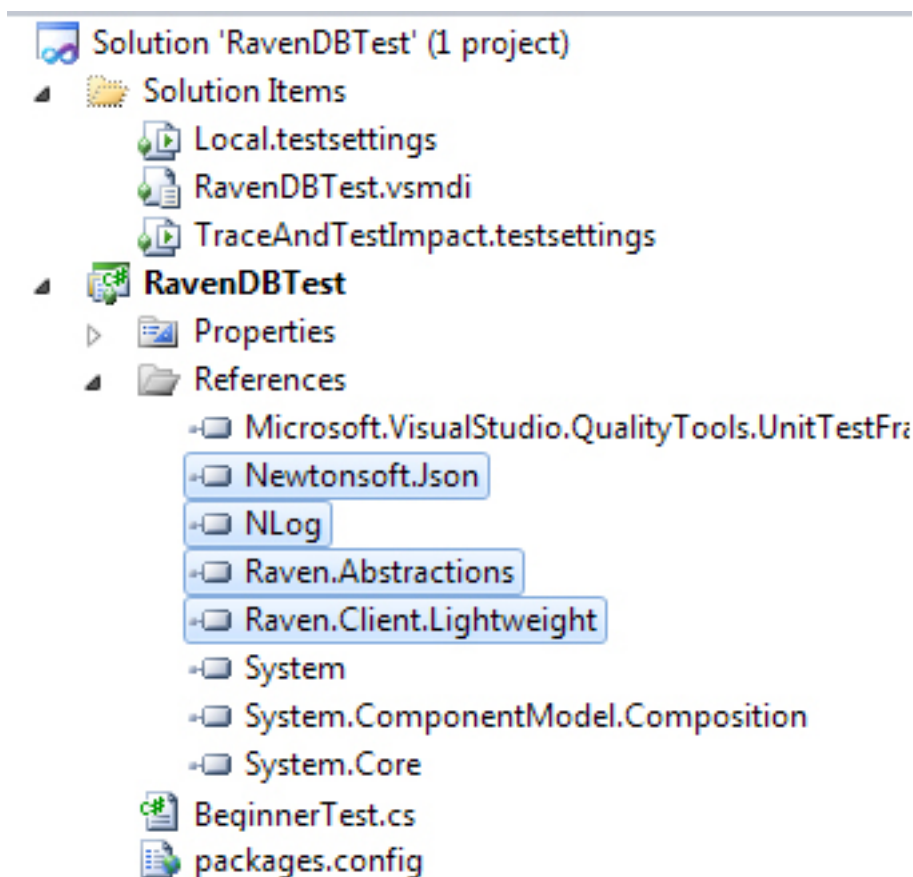
برای استفاده از کتابخانه‌های مورد نیاز دو راه وجود دارد:

استفاده از NuGet : با استفاده از دستور زیر Package مورد نیاز به پروژه شما افزوده می‌شود.

```
PM> Install-Package RavenDB -Version 1.0.919
```

اضافه کردن کتابخانه‌ها به صورت دستی : کتابخانه‌های مورد نیاز شما در همان فایل‌ها که دانلود شده بود و در پوشه Client قرار دارند.

کتابخانه‌هایی را که NuGet به پروژه من اضافه کرد، در تصویر زیر مشاهده می‌کنید :



با Newtonsoft.Json در اولین بخش بحث آشنا شدید. NLog هم یک کتابخانه قوی و مستقل برای مدیریت Log است که این پایگاه داده از آن بهره برده است.

" دلیل اینکه از پروژه تست استفاده کردم ؛ تمرکز روی کدها و مشاهده تاثیر آن‌ها ، مستقل از UI و لایه‌های دیگر نرم افزار است. بدیهی است که استفاده از آن‌ها در هر پروژه امکان پذیر است. "

برای شروع نیاز به آدرس Server و نام پایگاه داده داریم که می‌توانید در App.config به عنوان تنظیمات نرم افزار شما ذخیره شود و هنگام اجرای نرم افزار مقدار آن‌ها را خوانده و در متغیرهای readonly ذخیره شوند.

```
<appSettings>
  <add key="ServerName" value="http://SorousH-HP:8080/" />
  <add key="DatabaseName" value="TestDatabase" />
</appSettings>
```

هنگامی که صفحه Management Studio در مرورگر باز است، می‌توانید از نوار آدرس مرورگر خود آدرس سرور را به دست آورید.

```
[TestClass]
public class BeginnerTest
{
    private readonly string serverName;
    private readonly string databaseName;

    public BeginnerTest()
    {
        serverName = ConfigurationManager.AppSettings["ServerName"];
        databaseName = ConfigurationManager.AppSettings["DatabaseName"];
    }
}
```

برای برقراری ارتباط با پایگاه داده نیاز به یک شی از جنس DocumentStore و جهت انجام عملیات مختلف (ذخیره، حذف و ...) نیاز به یک شی از جنس IDocumentSession است. کد زیر، نحوه کار با آن‌ها را به شما نشان می‌دهد:

```
[TestClass]
public class BeginnerTest
{
    private readonly string serverName;
    private readonly string databaseName;

    private DocumentStore documentStore;
    private IDocumentSession session;

    public BeginnerTest()
    {
        serverName = ConfigurationManager.AppSettings["ServerName"];
        databaseName = ConfigurationManager.AppSettings["DatabaseName"];
    }

    [TestInitialize]
    public void TestStart()
    {
        documentStore = new DocumentStore { Url = serverName };
        documentStore.Initialize();
        session = documentStore.OpenSession(databaseName);
    }

    [TestCleanup]
    public void TestEnd()
    {
        session.SaveChanges();
        documentStore.Dispose();
        session.Dispose();
    }
}
```

در طراحی این پایگاه داده از الگوی Unit Of Work استفاده شده است. به این معنی که تمام تغییرات در حافظه ذخیره می‌شوند و به محض اجرای دستور session.SaveChanges() ارتباط برقرار شده و تمام تغییرات ذخیره خواهند شد.

هنگام شروع (تابع TestStart) متغیر session مقدار دهی می‌شود و در پایان کار (تابع TestEnd) تغییرات ذخیره شده و منابعی که توسط این دو شی در حافظه استفاده شده است، رها می‌شود.

البته بر مبنای طراحی شما، دستور `session.SaveChanges();` می‌تواند پس از انجام هر عملیات اجرا شود.

برای آشنا شدن با نحوه ذخیره کردن اطلاعات، به کد زیر دقت کنید:

```
class User
{
    public int Id { get; set; }
    public string Name { get; set; }
    public string Address { get; set; }
    public int Zip { get; set; }
}

[TestMethod]
public void Insert()
{
    var user = new User
    {
        Id = 1,
        Name = "John Doe",
        Address = "no-address",
        Zip = 65826
    };
    session.Store(user);
}
```

اگر همه چیز درست پیش رفته باشد، وقتی به محیط RavenDB Studio که هنوز در مرورگر شما باز است، نگاهی می‌اندازید، یک سند جدید ایجاد شده است که با کلیک روی آن، اطلاعات آن قابل مشاهده است. لحظه‌ی لذت بخشی است... یکی از روش‌های خواندن اطلاعات هم به صورت زیر است:

```
[TestMethod]
public void Select()
{
    var user = session.Load<User>(1);
}
```

نتیجه خروجی این دستور هم یک شیء از جنس کلاس `User` است.

تا این جا، ساده‌ترین مثال‌های ممکن را مشاهده کردید و حتما در بحث بعد مثال‌های جالب‌تر و دقیق‌تری را بررسی می‌کنیم و همچنین نگاهی به جزئیات طراحی و قراردادهای از پیش تعیین شده می‌اندازیم.

" به شما پیشنهاد می‌کنم که منتظر بحث بعدی نباشید! همین حالا دست به کار شوید... "

نسخه بدون کتابخانه‌های موردنیاز ( 2 مگابایت ) : [RavenDBTest\\_Small.zip](#)

نسخه کامل ( 15 مگابایت ) : [RavenDBTest.zip](#)

## نظرات خوانندگان

نویسنده: ناصر طاهری  
تاریخ: ۱۷:۱۲ ۱۳۹۱/۰۴/۱۶

سلام.  
مقوله‌ی جالبیه برای من.  
منتظر ادامه هستم. موفق باشید.

نویسنده: حسین  
تاریخ: ۱۹:۱۳ ۱۳۹۱/۰۴/۱۶

بسیار ممنون که کاربردی پیش میرین. من پروژمو با همین روش پیاده سازی می‌کنم و از اطلاعات خوبتون استفاده کردم.

نویسنده: ramín\_rp  
تاریخ: ۱۰:۴۳ ۱۳۹۱/۰۴/۱۷

سلام  
وقتی raven db رو استارت میکنم و محیط مدیریت اون تو browser اجرا میشه برای انجام عملیاتی مثل ایجاد دیتابیس user,pass میخواد که هرچی میدم قبول نمیکنه  
حتی raven رو به عنوان service هم نصب کردم ولی مشکل حل نشد  
جستجو تو نت هم نتیجه نداد  
مشکل از چیه؟

(مستندات رسمی raven db خوب نیست، مستندات mongodb واقعا کامل و جامع هست)

نویسنده: وحید نصیری  
تاریخ: ۱۱:۳۷ ۱۳۹۱/۰۴/۱۷

توضیحات بیشتر [در اینجا](#)

By default RavenDB allow anonymous access only for read requests (HTTP GET), and since we creating data, we need to specify a username and password. You can control this by changing the AnonymousAccess setting in the server configuration file. Enter your username and password of your Windows account and a sample data will be generated for you.

نویسنده: سروش ترک زاده  
تاریخ: ۲۱:۱۸ ۱۳۹۱/۰۴/۱۷

سلام  
ببخشید که دیر به سوال شما پاسخ دادم...  
یه راه دیگه، علاوه بر راهی که توسط جناب آقای نصیری ارائه شده است، وجود دارد.  
در پوشه Server فایل Raven.Server.exe را با Notepad باز کنید، سپس مقدار تنظیمات با کلید "Raven/AnonymousAccess" را به "All" تغییر دهید. توجه کنید که به بزرگ و کوچک بودن حروف حساس است.

در ضمن RavenDB از نظر سابقه و تعداد کاربران، قابل مقایسه با پایگاه داده هایی مثل SQL نیست و حق با شماست...

نویسنده: صابر

تاریخ: ۱۱:۱۹ ۱۳۹۱/۰۴/۲۱

سلام

نمی‌دانم مشکل از چیه ؟ ولی وقتی من سعی می‌کنم که بسته‌ی RavenDB رو از طریق Nuget دریافت کنم Error زیر رو می‌ده .

```
Install-Package : The element 'metadata' in namespace
'http://schemas.microsoft.com/packaging/2010/07/nuspec.xsd' has invalid child element
'frameworkAssemblies' in namespace
'http://schemas.microsoft.com/packaging/2010/07/nuspec.xsd'. List of possible elements expected:
'summary' in namespace
'http://schemas.microsoft.com/packaging/2010/07/nuspec.xsd'.
At line:1 char:1
+ Install-Package RavenDB -Version 1.0.919
+ ~~~~~
+ CategoryInfo          : NotSpecified: (:) [Install-Package], InvalidOperationException
+ FullyQualifiedErrorId : NuGet.VisualStudio.Cmdlets.InstallPackageCmdlet
```

نویسنده: وحید نصیری

تاریخ: ۱۱:۵۸ ۱۳۹۱/۰۴/۲۱

به احتمال زیاد VS.NET شما دسترسی به اینترنت ندارد ( [^](#) و [^](#) ).

نویسنده: peyman

تاریخ: ۹:۱۸ ۱۳۹۱/۰۵/۰۸

آقا سروش کی شروع میکنی سری جدید رو منتظریم قربان !

نویسنده: سروش ترک زاده

تاریخ: ۱۸:۵۹ ۱۳۹۱/۰۵/۰۸

سلام

ببخشید که دیر شد، به احتمال زیاد پنجشنبه ادامه آموزش را روی سایت قرار خواهیم داد...

نویسنده: پژمان

تاریخ: ۲۳:۲۵ ۱۳۹۱/۰۶/۰۱

ravendb هم مثل اینکه برای جستجو از لوسین استفاده می‌کنه.

نویسنده: فرزاد

تاریخ: ۲۳:۵۰ ۱۳۹۱/۰۷/۱۰

سلام

می‌خواهم یه نرم افزار تحت ویندوز بنویسم که نیاز دارم از بانک اطلاعاتی استفاده کنم از طرفی هم نمی‌خواهم با Sql server و یا access کار کنم چون نیاز به نرم افزار هایی با حجم زیاد هست که برای کاربر دردسر میشه. می‌خواستم اگه میشه بفرمایید که به نظرتون از چی استفاده کنم بهتره؟

ممنون

نویسنده: حسین مرادی نیا

تاریخ: ۵:۴۸ ۱۳۹۱/۰۷/۱۱

نکته اینکه وقتی بانک اطلاعات Access رو استفاده کنین ، حتما نیازی نیست که Access روی کامپیوتر کاربر نصب باشه تا بتونه از برنامه شما استفاده کنه.

به هر حال میتونید از Sql Server CE استفاده کنید: <http://www.dotnettips.info/search/label/SQL%20Server%20CE>