

# RavenDB: NoSQL Database for .NET

12 de dezembro de 2016 rbento 0

---

Fala Galera,

Hoje venho falar sobre um banco de dados noSQL, feito nativamente em .NET e ainda por cima open-source, o nome dele é o **RavenDB**.

O **RavenDB** é um banco 100% nativo .NET, Open Source, extremamente rápido e facilmente escalável permitindo que a sua aplicação .NET faça leituras e escritas de maneira rápida e com uma performance muito boa.

O **RavenDB** é um banco NoSQL do tipo “Document Database”, o que significa que ele armazena e recupera documentos, que podem estar em formato XML, JSON, entre outros.

Um banco do tipo “Document” é orientado a agregações que vem do **Domain-Driven Design** “Aggregates” que significa que cada documento é uma coleção de objetos relacionados que age sobre um contexto.

## Exemplo de utilização do RavenDB

Para esse exemplo precisamos instalar o **RavenDB**, para isso devemos ir em <https://ravendb.net/downloads> e pegar o zip ou o seu executável.

Dentro do zip deverá ter um command chamado **start.cmd**, selecione este arquivo e execute-o. Isso irá startar o servidor do **RavenDB** (Raven.Server.exe) e ainda abrirá no browser o **Management Studio**, ferramenta de administração do **RavenDB**.

Nosso foco será o Client .NET que podemos adicionar via NuGet no nosso projeto, ao adicionar o Client .NET do **RavenDB** estaremos fazendo referência a dois pacotes chamado **Raven.Abstractions** e **Raven.Client.Lightweight**

Para o nosso exemplo vamos criar um **Console Application** e adicionar os pacotes do **RavenDB**. Neste exemplo vamos usar um exemplo de **Autor, Post e Comentários**. Abaixo temos nosso modelo de classe que fará representação do domínio citado acima.

```
1. public class Author
2. {
3.     public Guid Id { get; set; }
4.
5.     public String Name { get; set; }
6.
7.     public String Email { get; set; }
8.
9.     public List<Post> Post { get; set; }
10. }
11.
12. public class Post
13. {
14.     public Guid Id { get; set; }
15.
16.     public String Title { get; set; }
17.
18.     public String Article { get; set; }
19.
20.     public DateTime DtPost { get; set; }
```

```

21.
22.         public List<Comment> Comments { get; set; }
23.
24.     }
25.
26.     public class Comment
27.     {
28.         public Guid Id { get; set; }
29.
30.         public String Title { get; set; }
31.
32.         public String Text { get; set; }
33.
34.         public String CommentUser { get; set; }
35.
36.         public DateTime DtComment { get; set; }
37.
38.     }

```

Acima está a implementação do nosso modelo, agora vamos fazer um test-drive do **RavenDB**, para isso no **Main** do nosso **Console Application**, vamos gravar um documento, alterar e consultar, com isso vamos poder ver como funciona o **RavenDB**

Vamos adicionar o Client do **RavenDB**, como foi mencionado acima, ele está disponível no **NuGet**. No **Package Manager Console** execute o comando abaixo:

- Install-Package RavenDB.Client

Com o pacote instalado, vamos fazer nosso test-drive do **RavenDB**, no método **Main** adicione o código abaixo:

```

1.     static void Main(string[] args)
2.     {
3.         var documentStore = new DocumentStore();
4.         documentStore.Url = "http://localhost:8080";
5.         documentStore.Initialize();
6.
7.         //Cria a base de dados caso necessario
8.         documentStore.DatabaseCommands.GlobalAdmin.EnsureDatabaseExists("Post");
9.
10.        //Armazena o Id para posteriormente seguir o exemplo de alteração e
delete
11.        var idAuthor = Guid.NewGuid();
12.
13.        //Cria um novo documento
14.        using (var session = documentStore.OpenSession("Post"))
15.        {
16.
17.            var author = new Author()
18.            {
19.                Id = idAuthor,
20.                Email = "xpto@teste.com.br",
21.                Name = "Rafael Cruz",
22.                Post = new List<Post>() {
23.                    new Post()
24.                    {
25.                        Id = Guid.NewGuid(),
26.                        Article = "Lorem ipsum",

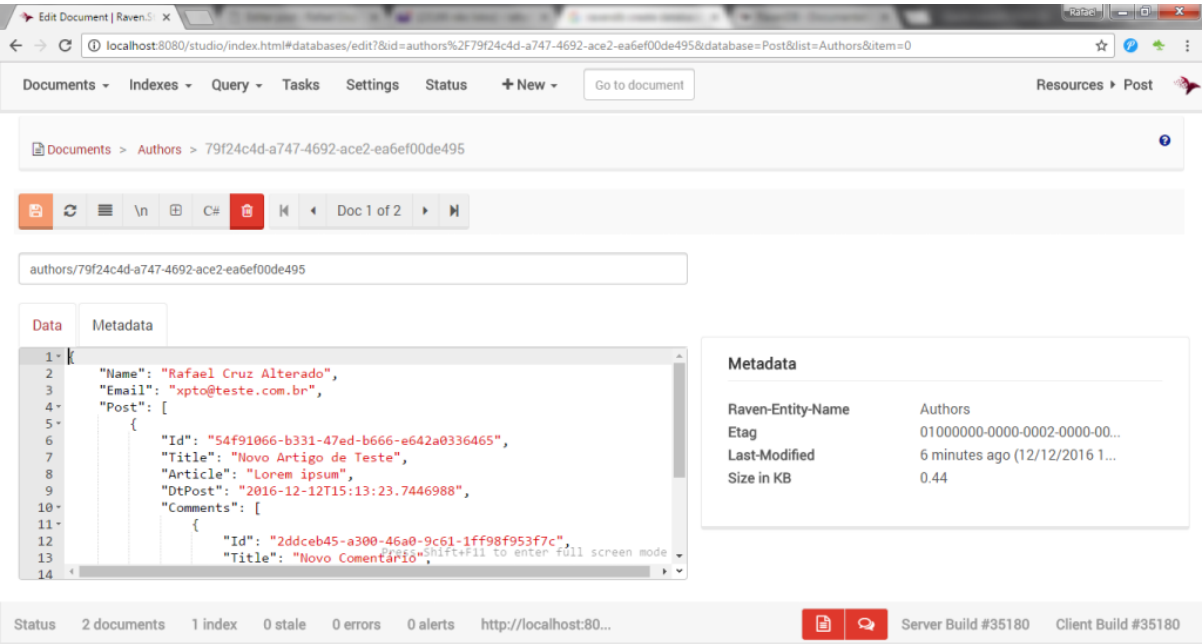
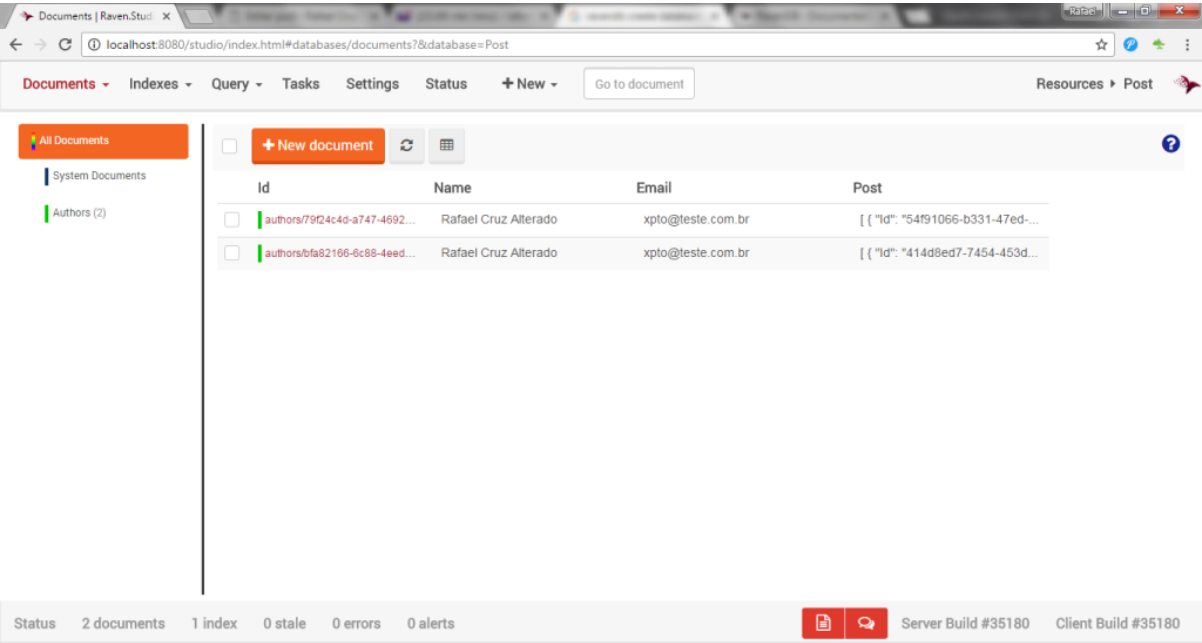
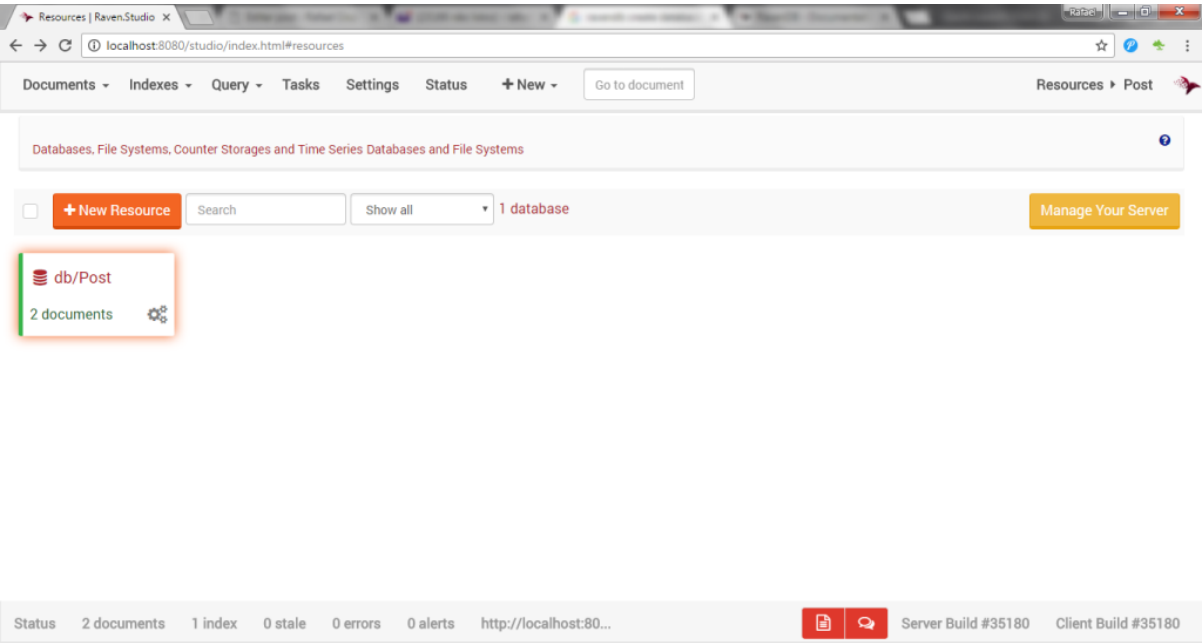
```

```
27.         DtPost = DateTime.Now,
28.         Title = "Novo Artigo de Teste",
29.         Comments = new List<Comment>()
30.     {
31.         new Comment()
32.         {
33.             Title = "Novo Comentário",
34.             Text = "Lorem Ipsum",
35.             CommentUser = "User Comment",
36.             DtComment = DateTime.Now,
37.             Id = Guid.NewGuid(),
38.         }
39.     }
40. }
41.
42. }
43. }
44. };
45.
46.     session.Store(author);
47.     session.SaveChanges();
48. }
49.
50. //Alterando uma ou mais propriedades
51. using (var session = documentStore.OpenSession("Post"))
52. {
53.     // usamos o método Load para recuperar um documento pelo Id
54.     var author = session.Load<Author>($"authors/{idAuthor}");
55.
56.     author.Name = "Rafael Cruz Alterado";
57.     session.SaveChanges();
58. }
59.
60. //Deletando um documento
61. using (var session = documentStore.OpenSession("Post"))
62. {
63.     // usamos o método Load para recuperar um documento pelo Id
64.     var author = session.Load<Author>($"authors/{idAuthor}");
65.
66.     session.Delete<Author>(author);
67.     session.SaveChanges();
68. }
69.
70. documentStore.Dispose();
71.
72. Console.WriteLine("Operação concluída");
73. Console.ReadKey();
74.
75. }
```

Neste código estamos criando uma nova base de dados chamada Post e inserindo um objeto chamado Author.

Ao instalar o **RavenDB**, ele vem com uma parte administrativa chamada **Management Studio**, nele é possível fazer administração da nossa base de dados.

Ao entrar na nossa base de dados chamada **Post** podemos ver um documento gravado como **JSON** conforme figura abaixo



## Conclus\u00e3o

O **RavenDB** é um banco de dados NoSQL, Open Source, escrito em .NET e que funciona como uma API REST. Ele é muito simples de usar pois se assemelha muito com os ORM's já conhecidos como **Entity Framework** e **NHibernate**

Além de contar com uma interface de administração muito fácil de se usar e com muitos recursos.

Quer saber mais sobre o **RavenDB**, [clique aqui](#) e conheça sua documentação.

O exemplo acima pode ser encontrado no meu GitHub através deste [link](#)

Abs e até a próxima