LiteDB

Getting Started

LiteDB is a simple, fast and lightweight embedded .NET document database. LiteDB was inspired by the MongoDB database and its API is very similar to the official MongoDB .NET API.

How to install

LiteDB is a serverless database, so there is no installation. Just copy LiteDB.dll into your Bin folder and add it as Reference. Or, if you prefer, you can install via NuGet: `Install-Package LiteDB` . If you are running in a web environment, make sure that your IIS user has write permission to the data folder.

First example

A quick example to store and search for documents:

```
// Create your POCO class entity
public class Customer
{
    public int Id { get; set; }
```

```
    public string Name { get; set; }
    public string[] Phones { get; set; }
    public bool IsActive { get; set; }
}

// Open database (or create if doesn't exist)
using(var db = new LiteDatabase(@"C:\Temp\MyData.db"))
{
    // Get a collection (or create, if doesn't exist)
    var col = db.GetCollection<Customer>("customers");

    // Create your new customer instance
    var customer = new Customer
    {
        Name = "John Doe",
        Phones = new string[] { "8000-0000", "9000-0000" },
        IsActive = true
    };

    // Insert new customer document (Id will be auto-incremented)
    col.Insert(customer);

    // Update a document inside a collection
    customer.Name = "Jane Doe";

    col.Update(customer);

    // Index document using document Name property
    col.EnsureIndex(x => x.Name);

    // Use LINQ to query documents (filter, sort, transform)
    var results = col.Query()
        .Where(x => x.Name.StartsWith("J"))
        .OrderBy(x => x.Name)
        .Select(x => new { x.Name, NameUpper = x.Name.ToUpper() })
        .Limit(10)
        .ToList();

    // Let's create an index in phone numbers (using expression). It's a multike
    col.EnsureIndex(x => x.Phones);

    // and now we can query phones
    var r = col.FindOne(x => x.Phones.Contains("8888-5555"));
}
```

Working with files

Need to store files? No problem: use FileStorage.

```
// Get file storage with Int Id
var storage = db.GetStorage<int>();

// Upload a file from file system to database
storage.Upload(123, @"C:\Temp\picture-01.jpg");

// And download later
storage.Download(123, @"C:\Temp\copy-of-picture-01.jpg");
```

Made with ♥ by LiteDB team - @mbdavid - MIT License