

DELLA



Manual de uso

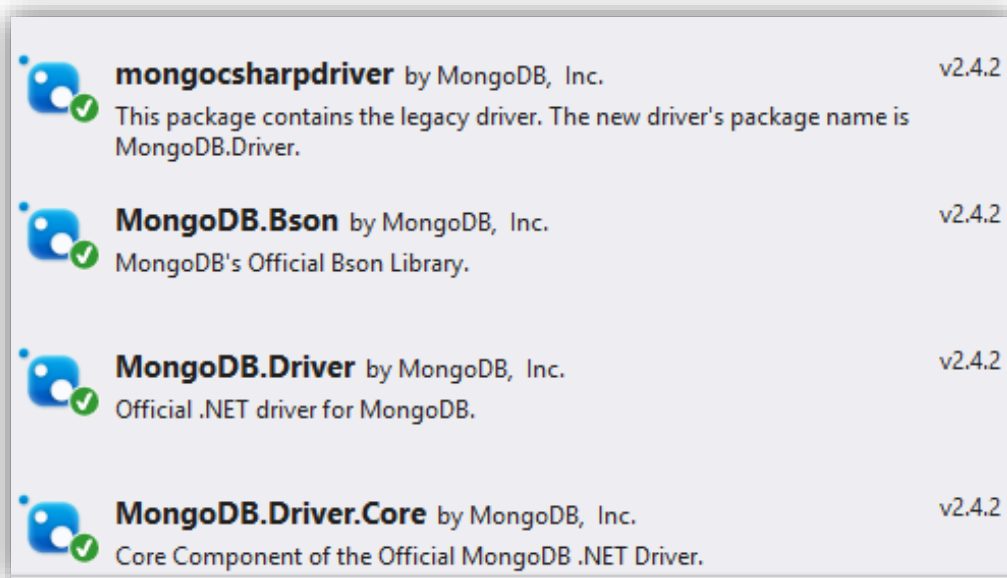
PROYECTO DELLA

PASANTÍAS DE VERANO 2017 - FX2

JIMENA MIGNACO

1 LA APLICACIÓN

La aplicación fue realizada en Visual Studio, en C#. Para su implementación se utilizaron drivers de Mongo DB, los mismos son los siguientes:



El objetivo de la misma es el de migrar los datos de una Base da Datos SQL local a una también local de MongoDB. Para ello, se obtienen los datos de SQL, se traducen a JSON y se importan mediante la herramienta mongoimport, contenida en la carpeta de instalación de MongoDB.

bsondump	1/2/2017 20:51	Aplicación	7.170 KB
libeay32.dll	19/12/2016 18:30	Extensión de la apl...	1.954 KB
mongo	1/2/2017 21:03	Aplicación	10.991 KB
mongod	1/2/2017 21:07	Aplicación	26.391 KB
mongod.pdb	1/2/2017 21:07	Archivo PDB	252.684 KB
mongodump	1/2/2017 20:53	Aplicación	9.232 KB
mongoexport	1/2/2017 20:52	Aplicación	7.427 KB
mongofiles	1/2/2017 20:52	Aplicación	7.342 KB
mongoimport	1/2/2017 20:53	Aplicación	7.520 KB
mongooplog	1/2/2017 20:54	Aplicación	7.171 KB
mongoperf	1/2/2017 21:08	Aplicación	22.364 KB
mongorestore	1/2/2017 20:53	Aplicación	10.540 KB
mongos	1/2/2017 21:06	Aplicación	13.139 KB
mongos.pdb	1/2/2017 21:06	Archivo PDB	124.204 KB
mongostat	1/2/2017 20:51	Aplicación	7.491 KB
mongotop	1/2/2017 20:54	Aplicación	7.302 KB
ssleay32.dll	19/12/2016 18:30	Extensión de la apl...	318 KB

Un detalle importante es que para ser posible la ejecución de todas las herramientas de mongo, es necesario tener iniciado el archivo mongod (el mismo se puede ver en la imagen anterior). Teniendo esto en cuenta, es indispensable tenerlo siempre abierto para que la aplicación pueda funcionar.

El tiempo de ejecución cuando se necesita migrar un único registro de una tabla es de aproximadamente 3 segundos y para 111685 registros, 180 segundos. Por lo tanto, sería interesante investigar modos para mejorar su eficiencia.

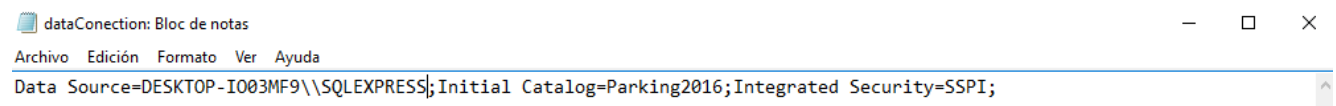
2 COMO FUNCIONA

2.1 CONFIGURAR CONEXIÓN CON LA BASE DE DATOS SQL

La aplicación se conecta con la base de datos SQL local, para ello, necesita tener configurados sus datos. Los mismos se encuentran en la carpeta bin/debug, en un .txt llamado dataconnection:

JSON	3/3/2017 22:42	Carpeta de archivos	
ClassLibrary1.dll	8/3/2017 9:59	Extensión de la apl...	20 KB
ClassLibrary1.pdb	8/3/2017 9:59	Archivo PDB	44 KB
Consola	8/3/2017 11:22	Aplicación	9 KB
Consola.exe	22/2/2017 12:31	XML Configuratio...	1 KB
Consola.pdb	8/3/2017 11:22	Archivo PDB	16 KB
Consola.vshost	8/3/2017 11:33	Aplicación	23 KB
Consola.vshost.exe	22/2/2017 12:31	XML Configuratio...	1 KB
Consola.vshost.exe.manifest	16/7/2016 8:44	Archivo MANIFEST	1 KB
dataConection	6/3/2017 14:27	Documento de tex...	1 KB
Dominio.dll	6/3/2017 14:49	Extensión de la apl...	13 KB
Dominio.pdb	6/3/2017 14:49	Archivo PDB	72 KB
MongoDB.Bson.dll	2/2/2017 8:11	Extensión de la apl...	435 KB
MongoDB.Bson	2/2/2017 8:11	Documento XML	1.043 KB
MongoDB.Driver.Core.dll	2/2/2017 8:12	Extensión de la apl...	555 KB
MongoDB.Driver.Core	2/2/2017 8:12	Documento XML	663 KB
MongoDB.Driver.dll	2/2/2017 8:12	Extensión de la apl...	543 KB
MongoDB.Driver	2/2/2017 8:12	Documento XML	909 KB
prueba	3/3/2017 16:16	Archivo por lotes ...	1 KB
Servicios.dll	8/3/2017 9:59	Extensión de la apl...	22 KB
Servicios.pdb	8/3/2017 9:59	Archivo PDB	40 KB
System.Runtime.InteropServices.Runtime...	11/6/2016 23:14	Extensión de la apl...	33 KB

Simplemente hay que sustituir los datos que tiene por los de la PC en la que se va a ejecutar



Se sustituye DESKTOP-IO03MF9\\SQLEXPRESS y Parking2016 por Data Source={servidor};InitialCatalog={Base de datos};Integrated Security=SSPI;

2.2 ARCHIVOS JSON

Los archivos JSON que contienen la traducción de clase a documento JSON se encuentran en la carpeta bin\debug\json tanto de la consola como del Servicio de Windows.

Se crean tres archivos .JSON: es, dd y dc. Cada uno corresponde a entradaSalida, docDetalle y docCabezal respectivamente. Estos archivos se crean, se importan a mongoDB y luego se reemplazan cuando se vuelve a ejecutar la aplicación.

2.3 OBTENCIÓN DEL ÚLTIMO REGISTRO

Para evitar procesar datos más de una vez, se le consulta a MongoDB cual es el último archivo ingresado de cada colección. Para la traducción se parte desde ese número. MongoDB puede asignar un ID por defecto o cambiar el atributo “_id” por el código deseado. En esta aplicación se realiza ese cambio, siendo el id de las tablas el mismo que se ve reflejado en la colección. Para la obtención del último registro se utilizan métodos async que utilizan el driver de MongoDB para conectarse con MongoDB y obtener el último id ingresado.

2.4 CLASES DE LA APLICACIÓN

Además de entradaSalida, docCabezal y docDetalle, se crean otras clases. Las mismas representan objetos contenidos en las anteriores. Se realiza su traducción por separado y luego se concatenan. Las clases Mapper{nombreTabla} se encargan de obtener los datos de la base de datos. Las clases Gestora{nombreTabla} se encargan de la traducción y luego tanto en consola como en el servicio de Windows se ejecutan sus métodos.

3 EJECUTAR LA APLICACIÓN

Actualmente al iniciar el proyecto, se ejecuta una consola, y se realizan todos los métodos (obtención de SQL, traducción, copiado a documento .JSON) hasta la ejecución de un archivo .bat (ubicado en consola/bin/pruebas.bat), el cual ejecuta la herramienta mongoimport y migra los datos.

Al final, al ejecutarse prueba.bat, se mantiene esa ventana abierta por 30 segundos lo que permite observar los resultados.

Esta creado también un Servicio de Windows. El mismo realiza lo mismo de la consola, cada un determinado tiempo (1 día), sería necesaria su instalación en el sistema y verificar su correcto funcionamiento.

4 CONTACTO

En caso de dudas o inconvenientes, puede comunicarse conmigo por medio de mi correo: jimeemk@gmail.com.