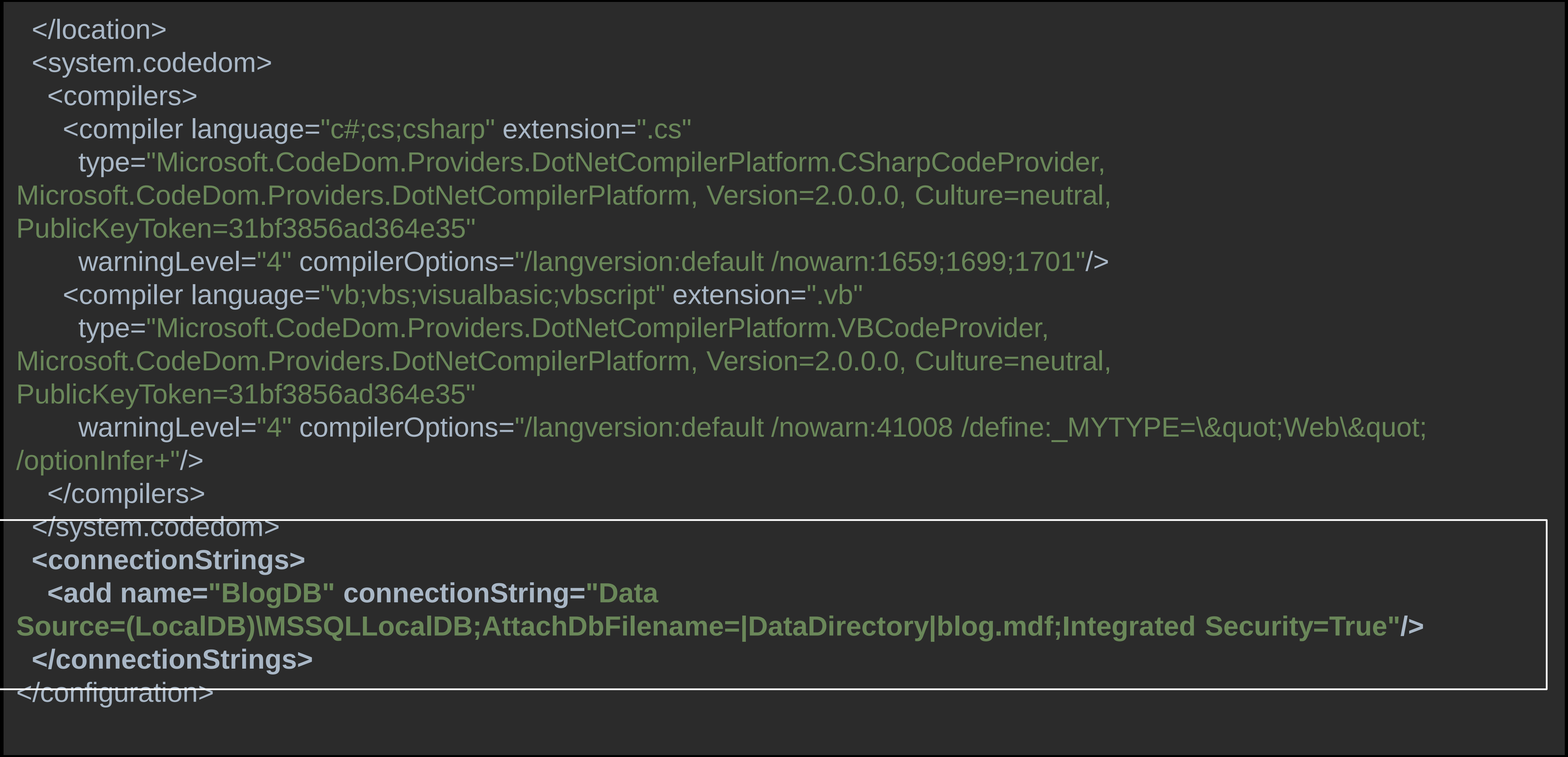
Creating_a_DB_Data_Adapter

- Creating the DB DataProvider
- Hooking up the new DataProvider to the Article Manager

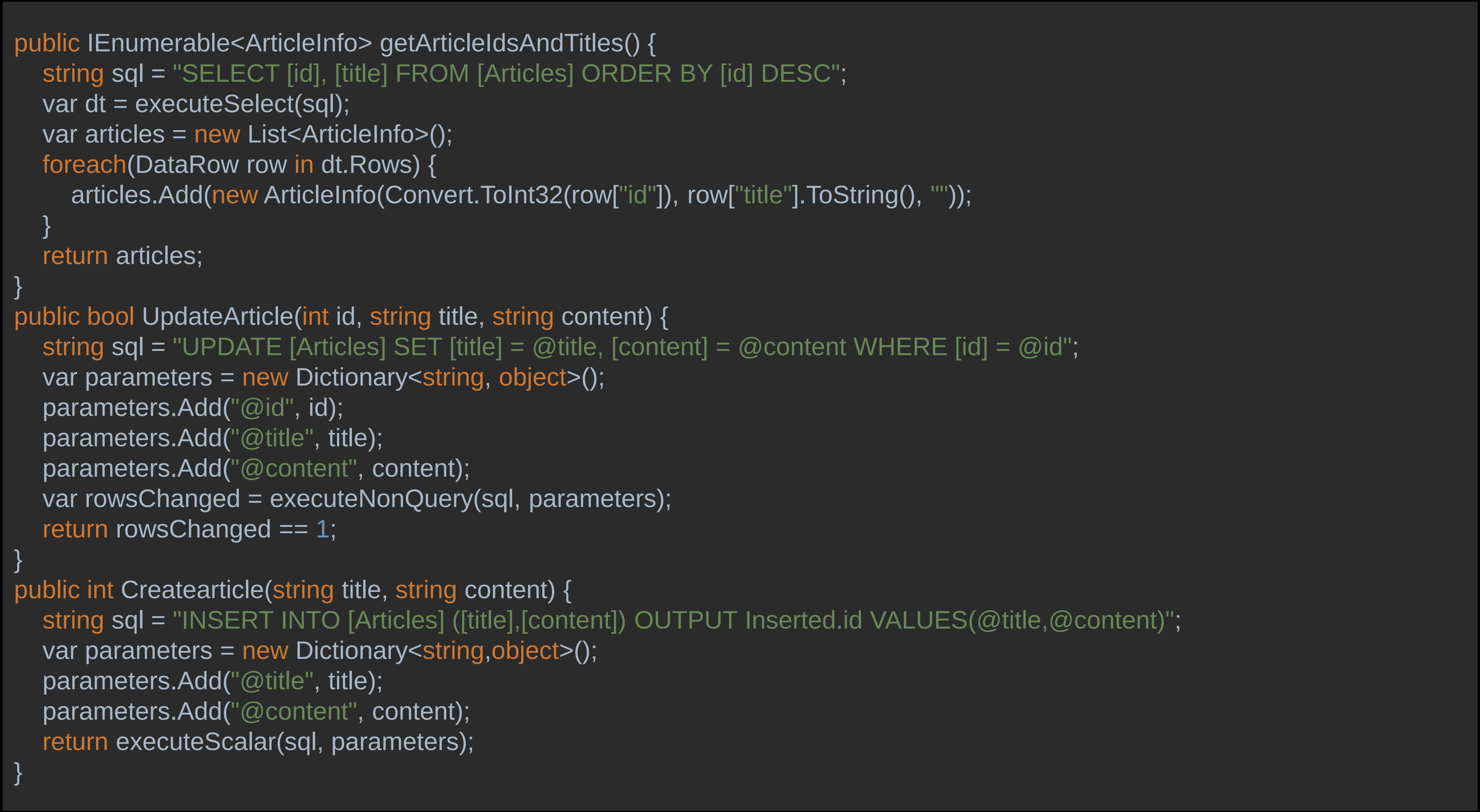
- Updating web.config



```
  </location>
  <system.codedom>
    <compilers>
      <compiler language="c#;cs;csharp" extension=".cs"
        type="Microsoft.CodeDom.Providers.DotNetCompilerPlatform.CSharpCodeProvider,
Microsoft.CodeDom.Providers.DotNetCompilerPlatform, Version=2.0.0.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35"
        warningLevel="4" compilerOptions="/langversion:default /nowarn:1659;1699;1701"/>
      <compiler language="vb;vbs;visualbasic;vbscript" extension=".vb"
        type="Microsoft.CodeDom.Providers.DotNetCompilerPlatform.VBCodeProvider,
Microsoft.CodeDom.Providers.DotNetCompilerPlatform, Version=2.0.0.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35"
        warningLevel="4" compilerOptions="/langversion:default /nowarn:41008 /define:_MYTYPE=\&quot;Web\&quot;
/optionInfer+"/>
    </compilers>
  </system.codedom>
  <connectionStrings>
    <add name="BlogDB" connectionString="Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=|DataDirectory|blog.mdf;Integrated Security=True"/>
  </connectionStrings>
</configuration>
```

-   DBDataProvider.cs

```
namespace MyBlog.src {
    public class DBDataProvider : IArticleDataProvider{
        private string _connectionString;
        /// <summary>     /// Initializes this provider.     /// </summary>
        public void Initialize() {
            _connectionString = System.Configuration.ConfigurationManager.ConnectionStrings["BlogDB"].ConnectionString;
        }
        public ArticleInfo GetArticleById(int id, bool getNext = false) {
            string sql = "SELECT TOP 1 [id], [title], [content] FROM [Articles] WHERE [id] <= @id ORDER BY [id] DESC";
            var parameters = new Dictionary<string, object>();
            parameters.Add("@id", getNext ? id - 1 : id);
            var dt = executeSelect(sql, parameters);
            if(dt.Rows.Count < 1) {
                // If there is no "next article" to get
                if (getNext) {       return null;            }
                throw new Exception("No articles exist");
            }
            var row = dt.Rows[0];
            return new ArticleInfo(Convert.ToInt32(row["id"]), row["title"].ToString(), row["content"].ToString());
        }
        public ArticleInfo GetLatestArticle() {
            string sql = "SELECT TOP 1 [id], [title], [content] FROM [Articles] ORDER BY [id] DESC";
            var dt = executeSelect(sql);
            if(dt.Rows.Count <1){ return null;    }
            var row = dt.Rows[0];
            return new ArticleInfo(Convert.ToInt32(row["id"]), row["title"].ToString(), row["content"].ToString());
        }
```

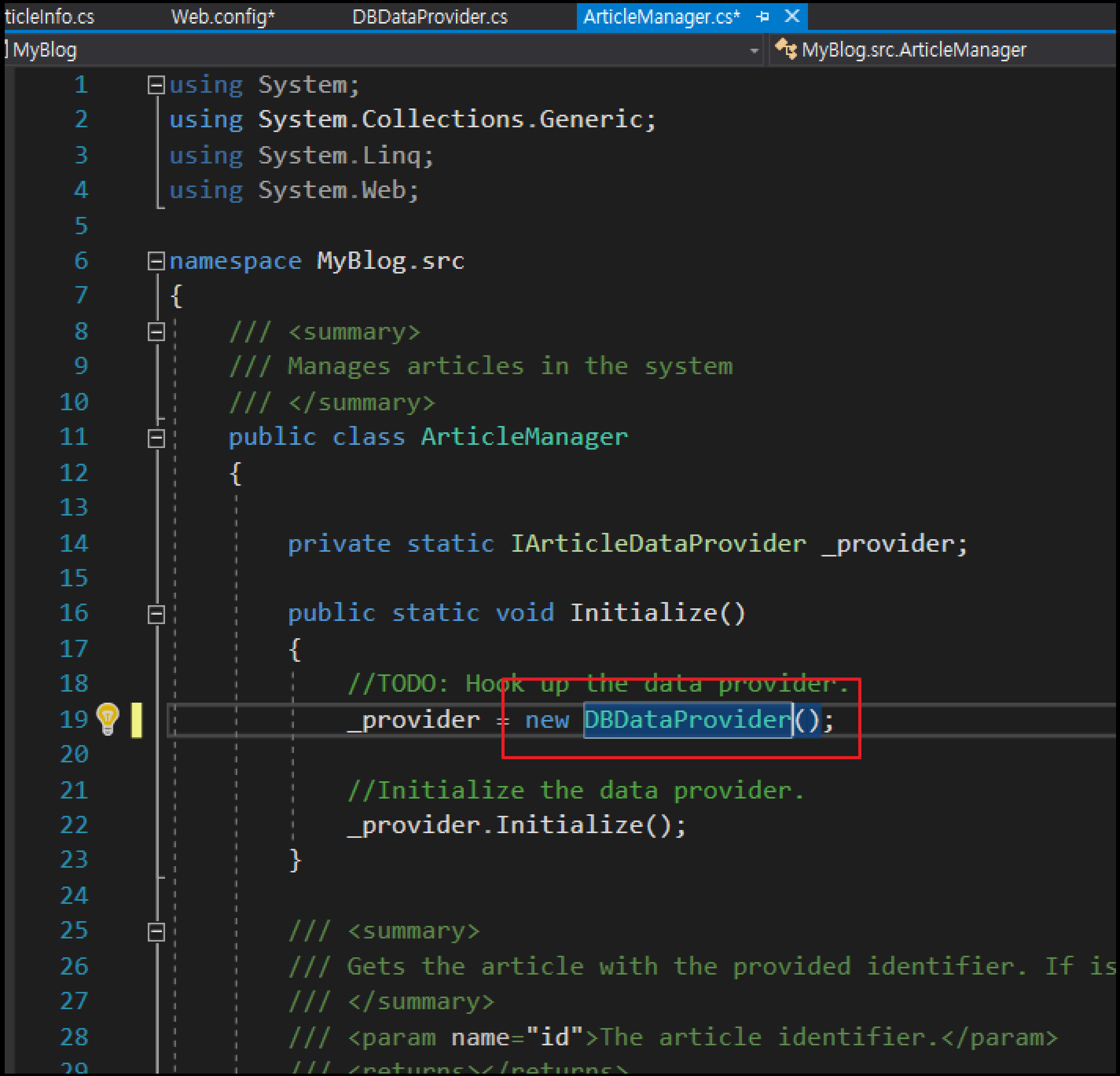


```
public IEnumerable<ArticleInfo> getArticleIdsAndTitles() {
    string sql = "SELECT [id], [title] FROM [Articles] ORDER BY [id] DESC";
    var dt = executeSelect(sql);
    var articles = new List<ArticleInfo>();
    foreach(DataRow row in dt.Rows) {
        articles.Add(new ArticleInfo(Convert.ToInt32(row["id"]), row["title"].ToString(), ""));
    }
    return articles;
}
public bool UpdateArticle(int id, string title, string content) {
    string sql = "UPDATE [Articles] SET [title] = @title, [content] = @content WHERE [id] = @id";
    var parameters = new Dictionary<string, object>();
    parameters.Add("@id", id);
    parameters.Add("@title", title);
    parameters.Add("@content", content);
    var rowsChanged = executeNonQuery(sql, parameters);
    return rowsChanged == 1;
}
public int Createarticle(string title, string content) {
    string sql = "INSERT INTO [Articles] ([title],[content]) OUTPUT Inserted.id VALUES(@title,@content)";
    var parameters = new Dictionary<string,object>();
    parameters.Add("@title", title);
    parameters.Add("@content", content);
    return executeScalar(sql, parameters);
}
```

```
private DataTable executeSelect(string sql, Dictionary<string,object> parameters = null) {
    var dt = new DataTable();
    using (var conn = new SqlConnection(_connectionString)) {
        conn.Open();
        using(var cmd= new SqlCommand(sql,conn)) {
            if(parameters != null) {
                foreach(var param in parameters) {
                    cmd.Parameters.AddWithValue(param.Key, param.Value ?? DBNull.Value);
                }
            }
            dt.Load(cmd.ExecuteReader());
        }   }
    return dt;
}
private int executeNonQuery(string sql, Dictionary<string, object> parameters) {
    using (var conn = new SqlConnection(_connectionString)) {
        conn.Open();
        using (var cmd = new SqlCommand(sql,conn)) {
            if(parameters !=null) {
                foreach(var param in parameters) {
                    cmd.Parameters.AddWithValue(param.Key, param.Value ?? DBNull.Value);
                }
            }
            return cmd.ExecuteNonQuery();
        }
    }}
```

```
private int executeScalar (string sql, Dictionary<string,object>parameters) {
            using (var conn = new SqlConnection(_connectionString)) {
                conn.Open();
                using (var cmd = new SqlCommand(sql,conn)) {
                    if(parameters != null) {
                        foreach(var param in parameters) {
                            cmd.Parameters.AddWithValue(param.Key, param.Value);
                        }
                    }
                    return (int)cmd.ExecuteScalar();
                }
            }
        }
    }

}
```



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace MyBlog.src
{
    /// <summary>
    /// Manages articles in the system
    /// </summary>
    public class ArticleManager
    {

        private static IArticleDataProvider _provider;

        public static void Initialize()
        {
            //TODO: Hook up the data provider.
            _provider = new DBDataProvider();

            //Initialize the data provider.
            _provider.Initialize();
        }

        /// <summary>
        /// Gets the article with the provided identifier. If is
        /// </summary>
        /// <param name="id">The article identifier.</param>
        /// <returns></returns>
```