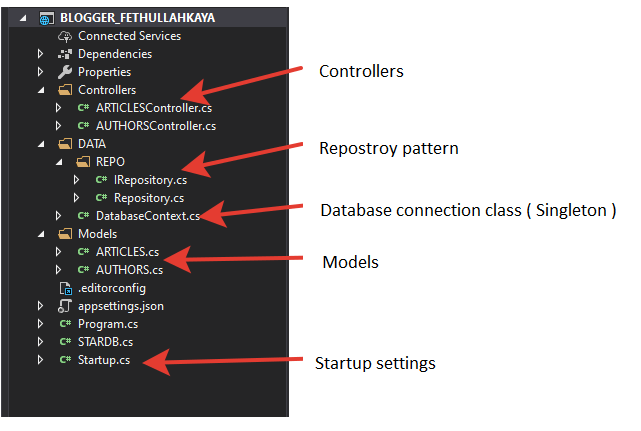
**README- BLOGGER**

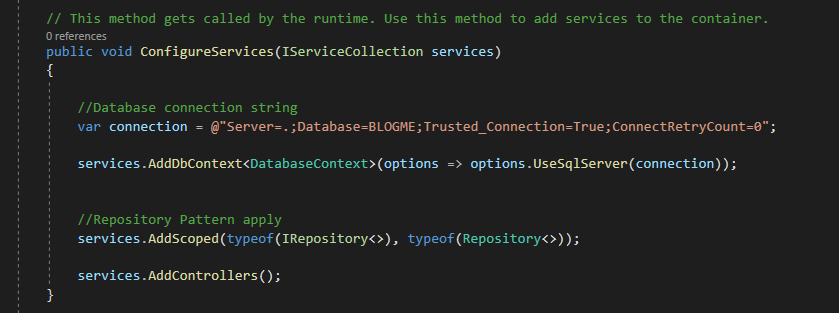
**A-Technology-design and implementations:**

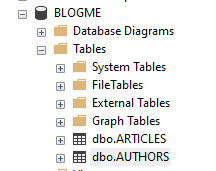
The blogger app is created with asp.net mvc core technology. ****

As you see on the above picture, the app has 2 Controllers as Articles and Authors. To demonstrate the usage and benefits of Repository pattern, the Author controller reaches out the database directly with the Database Context class. In the future, if we would make some changes or maintenance, we had to work too much. Whereas in the Articles Controllers, we get the database context through with repository class which interfaced from IRepostory class. In the future, if we would alter database or make some changes, it will be easier. And it will be easy to change for anybody too.

On the other hand, when we start the app, the Database Context gets instance one time. And each database operation is held with it. To improve performance, it’s necessary and this principle is described as singleton principle.

Note: The database context gets the connection string while startup.



HOW TO USE THE BLOGGER APP

The database designed as Microsoft Sql Server. The table creation scripts are below. But before please create a database named as BLOGME

1-Create the ARTICLES TABLE

CREATE TABLE [dbo].[ARTICLES](

[ARTICLE\_ID] [int] IDENTITY(1,1) NOT NULL,

[ARTICLE\_AUTHOR\_ID] [int] NULL,

[ARTICLE\_CREATE\_DATE] [date] NULL,

[ARTICLE\_EDIT\_AUTHOR\_ID] [int] NULL,

[ARTICLE\_EDIT\_DATE] [date] NULL,

[ARTICLE\_TITLE] [nvarchar](50) NULL,

[ARTICLE\_IMAGEURL] [nvarchar](max) NULL,

[ARTICLE\_DESCRIPTION] [nvarchar](max) NULL,

[ARTICLE\_CONTENT] [nvarchar](max) NULL,

[ARTICLE\_ISACTIVE] [bit] NULL,

[ARTICLE\_LIKE] [int] NULL

) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]

GO

2- Create the AUTHORS TABLE

CREATE TABLE [dbo].[AUTHORS](

[AUTOHOR\_ID] [int] IDENTITY(1,1) NOT NULL,

[AUTHOR\_NAME] [nvarchar](30) NULL,

[AUTHOR\_SURNAME] [nvarchar](30) NULL,

[AUTHOR\_CREATE\_DATE] [datetime] NULL,

[AUTHOR\_ISACTIVE] [bit] NULL,

CONSTRAINT [PK\_AUTHORS] PRIMARY KEY CLUSTERED

(

[AUTOHOR\_ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[AUTHORS] ADD CONSTRAINT [DF\_AUTHORS\_AUTHOR\_CREATE\_DATE] DEFAULT (getdate()) FOR [AUTHOR\_CREATE\_DATE]

GO

**HOW TO USE**

In the Articles Controller. The blogger app has 5 operations as List, Find, Add, Change and Delete. To call the methods, we need an app as Postman etc. The route is localhost+ port + api + controller name. As example 192.168.1.5:1600/api/Articles.

GET: <http://localhost:1600/API/ARTICLES>

GET: <http://localhost:1600/API/ARTICLES/1>

PUT: <http://localhost:1600/API/ARTICLES/1>

RAW DATA

{

"articlE\_ID": 1, "articlE\_AUTHOR\_ID": 1, "articlE\_CREATE\_DATE": "2019-11-11T00:00:00",

"articlE\_EDIT\_AUTHOR\_ID": 1, "articlE\_EDIT\_DATE": "2018-10-10T00:00:00",

"articlE\_TITLE": "PUT TEST", "articlE\_IMAGEURL": "PUT TEST", "articlE\_DESCRIPTION": "PUT TEST",

"articlE\_CONTENT": "PUT TEST", "articlE\_ISACTIVE": true, "articlE\_LIKE": 11

}

POST: <http://localhost:1600/API/ARTICLES>

RAW DATA

{ "authoR\_NAME": "zzz ", "authoR\_SURNAME": "zzz" }

DELETE: <http://localhost:1600/API/ARTICLES/4>