

The screenshot shows the Microsoft Visual Studio IDE interface. The main window displays a C# code editor with the following code:

```
1  using Microsoft.AspNetCore.Mvc;
2  using Microsoft.Extensions.Logging;
3  using Microsoft.EntityFrameworkCore;
4  using System;
5  using System.Collections.Generic;
6  using System.Linq;
7  using System.Threading.Tasks;
8  using System.Net.Http;
9  using System.IO;
10 using System.Reflection;
11 [Route("api/[controller]")]
12 [ApiController]
13 public class GetDataController : ControllerBase
14 {
15     private readonly ILogger<GetDataController> _logger;
16     private readonly RatesContext _context;
17
18     public GetDataController(ILogger<GetDataController> logger, RatesContext context)
19     {
20         _logger = logger;
21         _context = context;
22     }
23 }
```

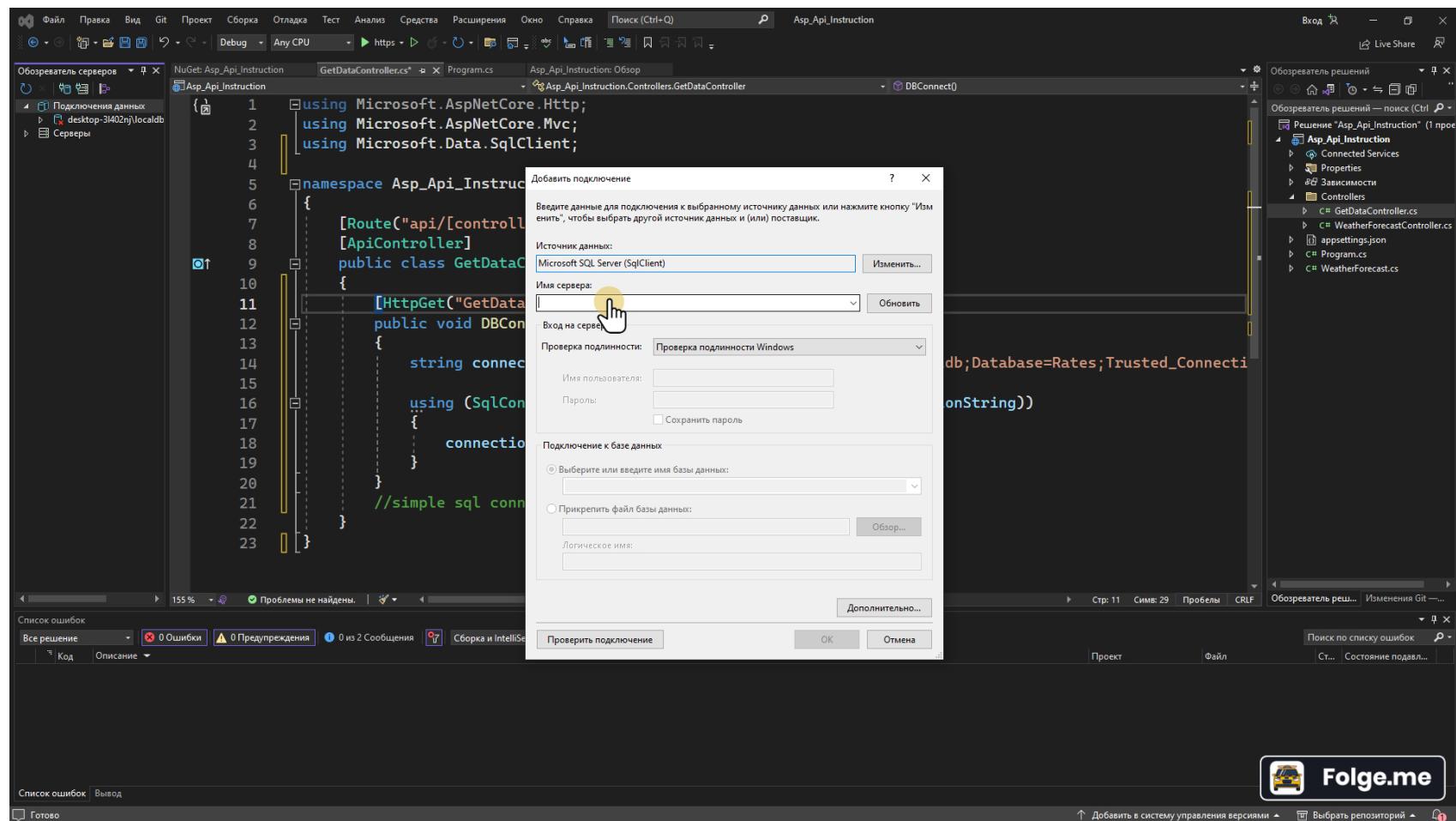
A context menu is open over the line of code where the cursor is positioned, specifically over the `using` statement for `Microsoft.EntityFrameworkCore`. The menu items visible include:

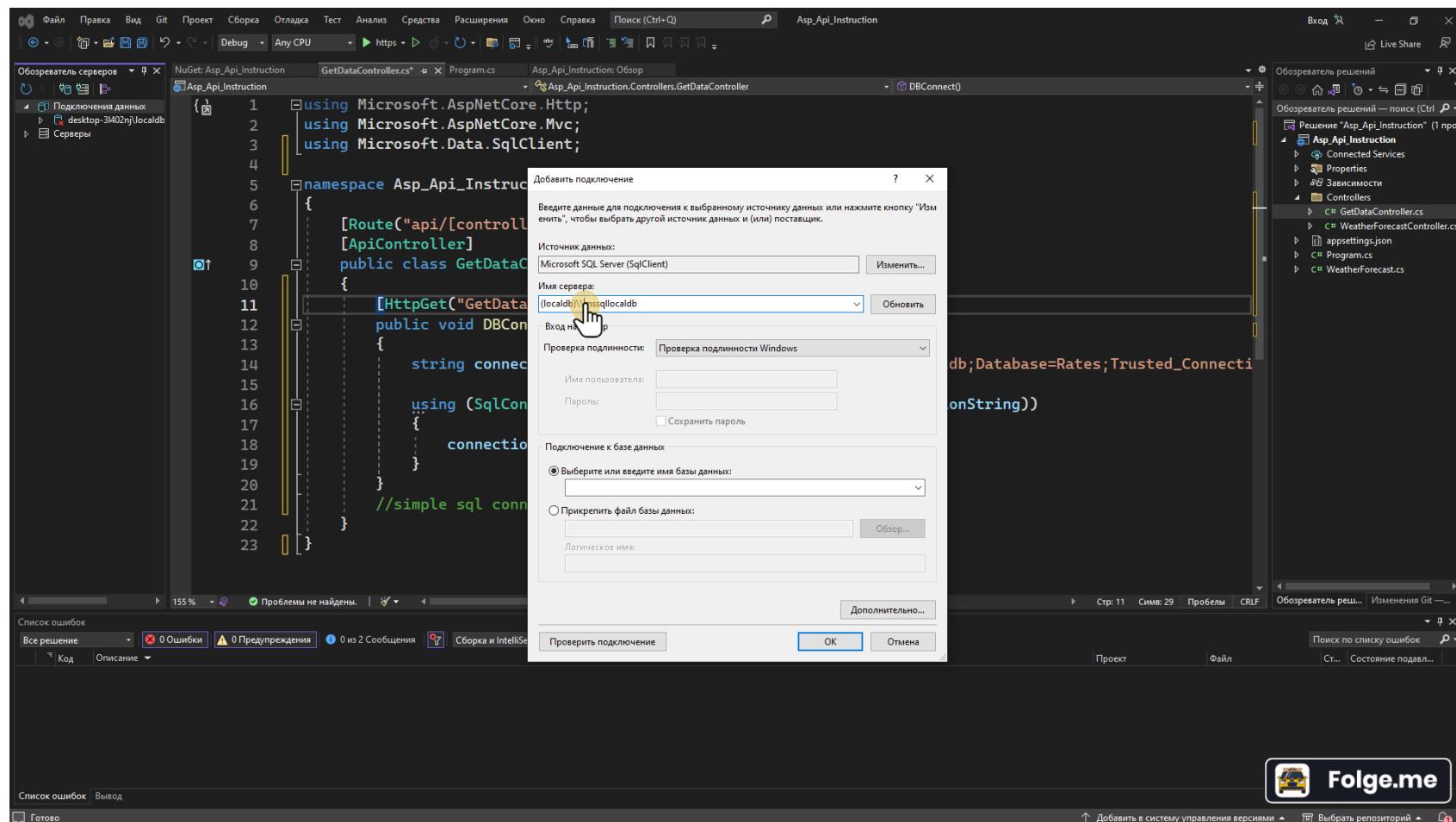
- Получить средства и компоненты...
- Управление предварительными версиями функций
- Android
- iOS
- Диспетчер архивов...
- Подключиться к базе данных...
- Подключиться к серверу...
- SQL Server
- Data Lake
- Диспетчер фрагментов кода... Ctrl+K, Ctrl+B
- Выбрать элементы панели элементов...
- Диспетчер пакетов NuGet
- Создать GUID
- Поиск ошибки
- Внешние инструменты...
- Тема
- Командная строка
- Импорт и экспорт параметров...
- Настройка...
- Параметры...

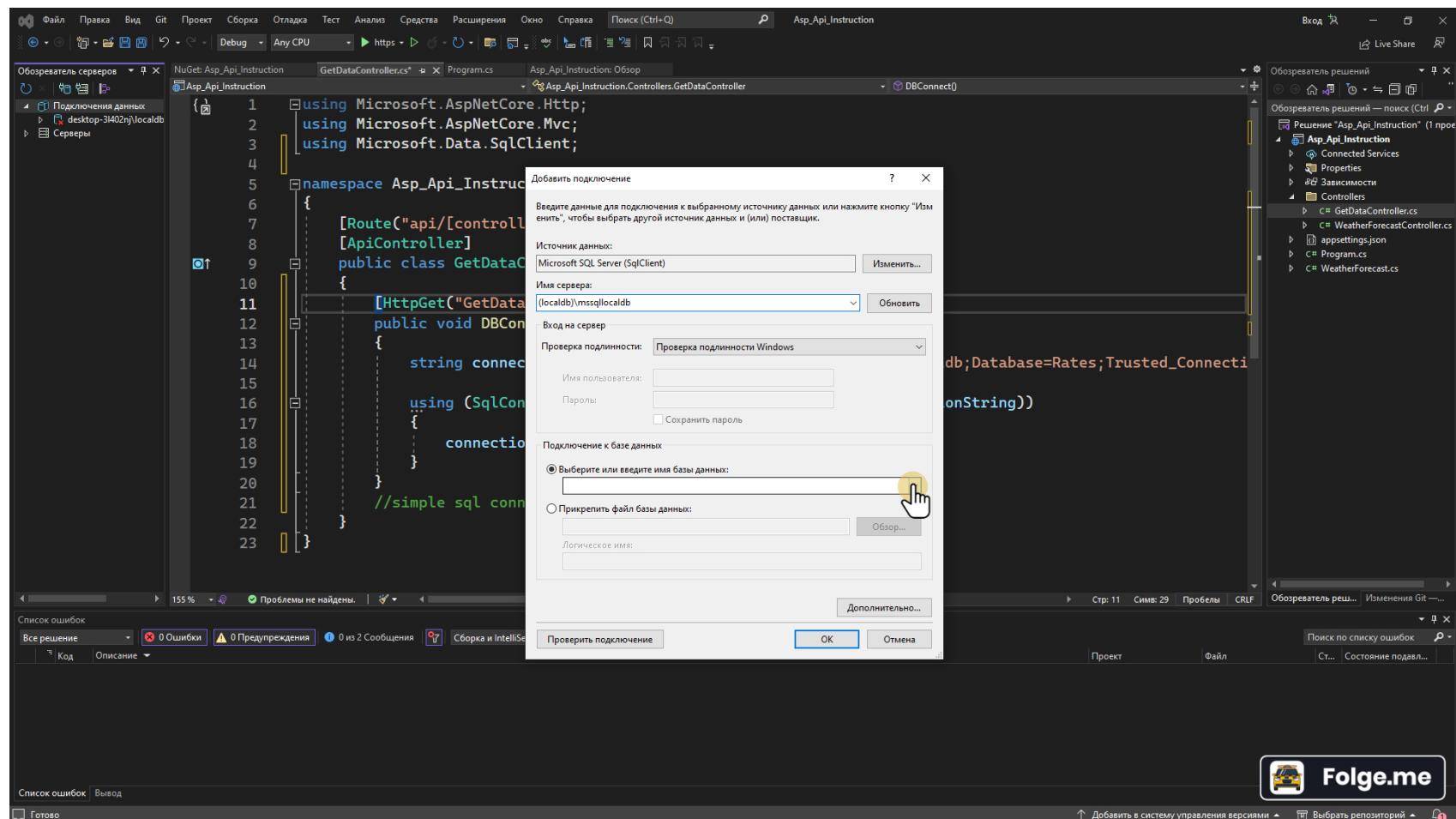
The solution explorer on the right shows the project structure:

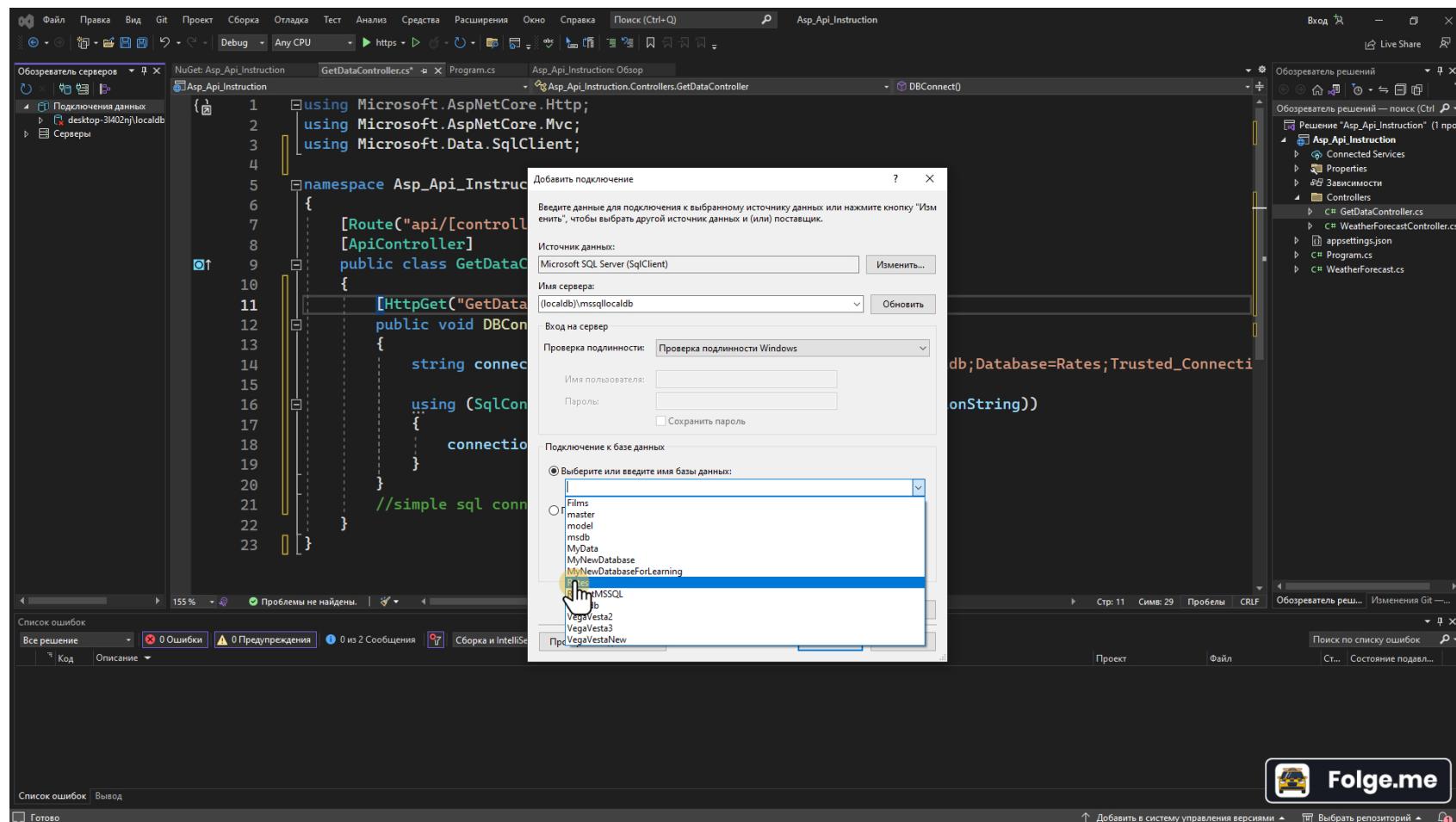
- Решение "Asp_Api_Instruction" (1 проек)
- Asp_Api_Instruction
- Connected Services
- Properties
- Зависимости
- Controllers
 - GetDataController.cs
 - WeatherForecastController.cs
- appsettings.json
- Program.cs
- WeatherForecast.cs

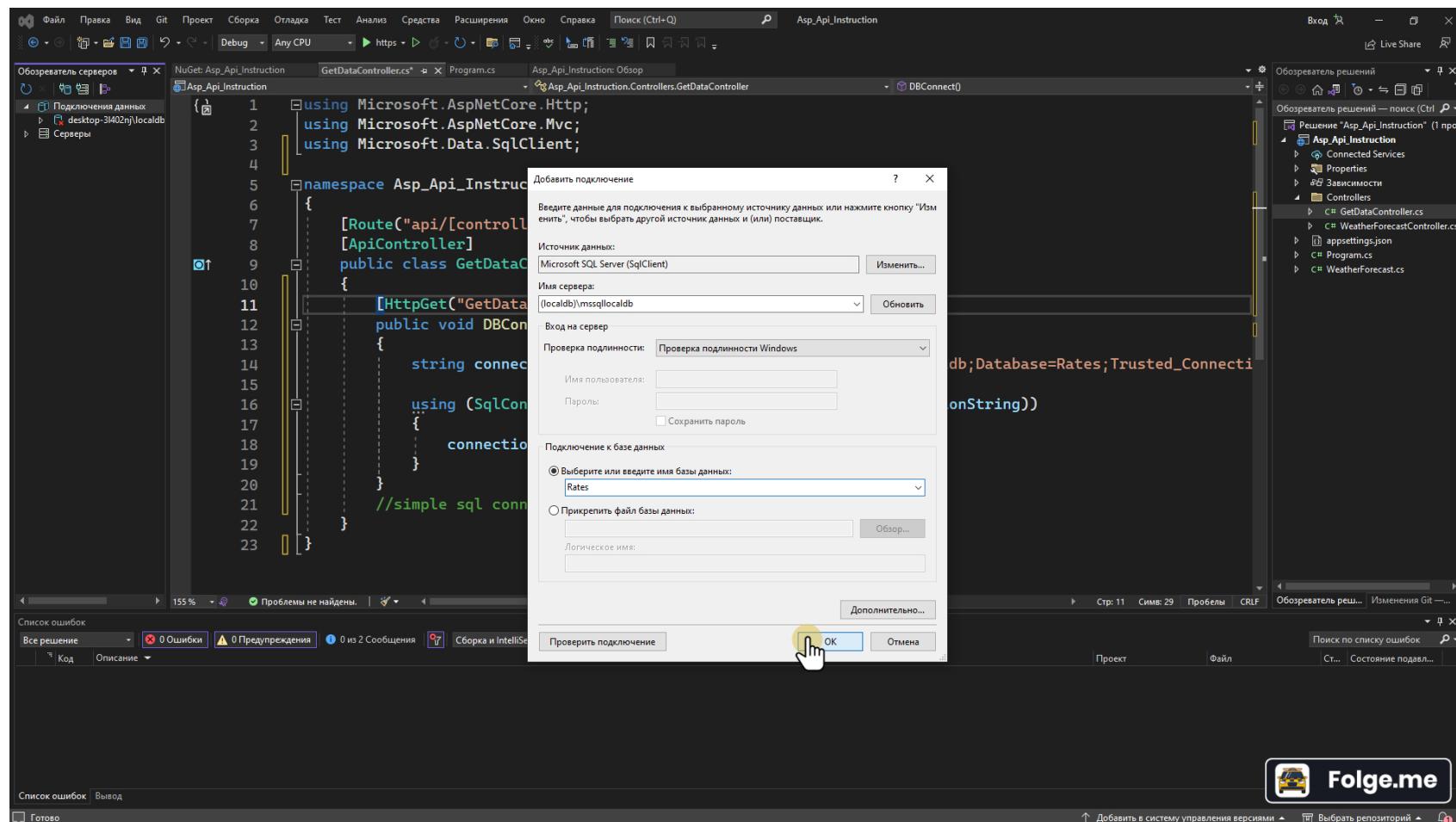
The status bar at the bottom indicates: Список ошибок, 0 Ошибки, 0 Предупреждения, 0 из 2 Сообщения, Сборка и IntelliSense.











The screenshot shows the Microsoft Visual Studio IDE interface. The main window displays the code for a controller named `GetDataController.cs`. The code uses `Microsoft.AspNetCore.Http`, `Microsoft.AspNetCore.Mvc`, and `Microsoft.Data.SqlClient` namespaces. It defines a class `GetDataController` that inherits from `ControllerBase`. The `DBConnect()` method is annotated with `[HttpGet("GetData")]` and `[ApiController]`. Inside this method, a connection string is defined, and a `SqlConnection` object is created and opened. A comment `//simple sql connection` is present. The solution explorer on the right shows the project structure with files like `GetDataController.cs`, `Program.cs`, and `appsettings.json`. The status bar at the bottom includes a watermark for "Folge.me".

```
1  using Microsoft.AspNetCore.Http;
2  using Microsoft.AspNetCore.Mvc;
3  using Microsoft.Data.SqlClient;
4
5  namespace Asp_Api_Instruction.Controllers
6  {
7      [Route("api/[controller]")]
8      [ApiController]
9      public class GetDataController : ControllerBase
10     {
11         [HttpGet("GetData")]
12         public void DBConnect()
13         {
14             string connectionString = "Data Source=(localdb)\mssqllocaldb;Database=Rates;Trusted_Connection=True";
15
16             using (SqlConnection connection = new SqlConnection(connectionString))
17             {
18                 connection.Open();
19             }
20         }
21         //simple sql connection
22     }
23 }
```

The screenshot shows the Microsoft Visual Studio IDE interface. The main window displays the code for a C# controller named `GetDataController.cs`. The code uses the `Microsoft.AspNetCore.Http`, `Microsoft.AspNetCore.Mvc`, and `Microsoft.Data.SqlClient` namespaces. It defines a class `GetDataController` that inherits from `ControllerBase`. The class contains a single action method `DBConnect()` annotated with `[HttpGet("GetData")]`. The method connects to a local database using a connection string and opens a `SqlConnection`.

```
1  using Microsoft.AspNetCore.Http;
2  using Microsoft.AspNetCore.Mvc;
3  using Microsoft.Data.SqlClient;
4
5  namespace Asp_Api_Instruction.Controllers
6  {
7      [Route("api/[controller]")]
8      [ApiController]
9      public class GetDataController : ControllerBase
10     {
11         [HttpGet("GetData")]
12         public void DBConnect()
13         {
14             string connectionString = "Data Source=(localdb)\\mssqllocaldb;Database=Rates;Trusted_Connection=True";
15
16             using (SqlConnection connection = new SqlConnection(connectionString))
17             {
18                 connection.Open();
19             }
20         }
21         //simple sql connection
22     }
23 }
```

The left sidebar shows the database connections for the project, specifically a connection to `desktop-3M402n\localdb`. The right sidebar shows the solution explorer with files like `appsettings.json`, `Program.cs`, and `WeatherForecast.cs`. The bottom status bar includes a watermark for `Folge.me`.

The screenshot shows the Microsoft Visual Studio IDE interface. The main window displays the code for a controller named `GetDataController.cs`. The code uses `Microsoft.AspNetCore.Http`, `Microsoft.AspNetCore.Mvc`, and `Microsoft.Data.SqlClient` namespaces. It defines a class `GetDataController` that inherits from `ControllerBase`. The `DBConnect()` method is annotated with `[HttpGet("GetData")]` and `[ApiController]`. Inside this method, a connection string is defined, and a `SqlConnection` object is created and opened. A comment `//simple sql connection` is present at the end of the method. The left sidebar shows the database connection configuration for "localdb". The right sidebar shows the project structure for the solution "Asp_Api_Instruction". The bottom status bar includes a watermark for "Folge.me".

```
1  using Microsoft.AspNetCore.Http;
2  using Microsoft.AspNetCore.Mvc;
3  using Microsoft.Data.SqlClient;
4
5  namespace Asp_Api_Instruction.Controllers
6  {
7      [Route("api/[controller]")]
8      [ApiController]
9      public class GetDataController : ControllerBase
10     {
11         [HttpGet("GetData")]
12         public void DBConnect()
13         {
14             string connectionString = "Data Source=(localdb)\mssqllocaldb;Database=Rates;Trusted_Connection=True";
15
16             using (SqlConnection connection = new SqlConnection(connectionString))
17             {
18                 connection.Open();
19             }
20         }
21         //simple sql connection
22     }
23 }
```