

ASP.NET Core 2.1 app not picking up appsettings.Development.json even though I specify ASPNETCORE_ENVIRONMENT=Development explicitly. Why?

So I am working on an ASP.NET Core 2.1 Web API. I specified two sets of configurations for "Development" and "Production" explicitly by the following two functions:

0

```
public void ConfigureServices(IServiceCollection services)
{
    services.AddDbContext<DataContext>(option => option
        .UseMySQL(Configuration.GetConnectionString("DefaultConnection")))
}

public void ConfigureDevelopmentServices(IServiceCollection services)
{
    services.AddDbContext<DataContext>(option => option
        .UseSqlite(Configuration.GetConnectionString("DefaultConnection")));
}
```

In the appsettings.json, I have:

```
{
  "ConnectionStrings": {
    "DefaultConnection": "Server=localhost; Database=datingApp; Uid=appuser;
    Pwd=zhanxucong"
  }
}
```

In the appsettings.Development.json, I have:

```
{
  "Logging": {
    "LogLevel": {

```

Join **Stack Overflow** to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH

 Google

Facebook 

```
}  
}  
}
```

When I learned through the tutorial, the instructor said that by convention, when `ASPNETCORE_ENVIRONMENT=Development` is set, ASP.NET Core will call the more specific configure function (i.e. `ConfigureDevelopmentServices`) and use the more specific appsettings config file (i.e. `appsettings.Development.json`). And I can confirm that it had worked fine on my Windows and Mac. However, when I run this app on Linux, after running ef migration with the above settings, I get the following errors:

```
info: Microsoft.EntityFrameworkCore.Infrastructure[10403]  
      Entity Framework Core 2.1.4-rtm-31024 initialized 'DataContext' using provider  
'Microsoft.EntityFrameworkCore.Sqlite' with options: None  
System.ArgumentException: Keyword not supported: 'server'.  
   at Microsoft.Data.Sqlite.SqliteConnectionStringBuilder.GetIndex(String keyword)  
   at Microsoft.Data.Sqlite.SqliteConnectionStringBuilder.set_Item(String keyword,  
Object value)  
   at System.Data.Common.DbConnectionStringBuilder.set_ConnectionString(String value)  
   at  
Microsoft.EntityFrameworkCore.Sqlite.Storage.Internal.SqliteRelationalConnection.CreateRea  
  
   at  
Microsoft.EntityFrameworkCore.Sqlite.Storage.Internal.SqliteDatabaseCreator.Exists()  
   at Microsoft.EntityFrameworkCore.Migrations.HistoryRepository.Exists()  
   at Microsoft.EntityFrameworkCore.Migrations.Internal.Migrator.Migrate(String  
targetMigration)  
   at  
Microsoft.EntityFrameworkCore.Design.Internal.MigrationsOperations.UpdateDatabase(String  
targetMigration, String contextType)  
   at Microsoft.EntityFrameworkCore.Design.OperationExecutor.UpdateDatabase.  
<>c__DisplayClass0_1.<.ctor>b__0()  
   at  
Microsoft.EntityFrameworkCore.Design.OperationExecutor.OperationBase.Execute(Action  
action)  
Keyword not supported: 'server'.
```



Any idea why this error will happen? Is it related to Linux being case-sensitive? Thanks!

Update: My Program class and my StartUp class (default ctor provided by dotnet-cli tool by running "dotnet new webapi"):

Join Stack Overflow to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH



Google



```
public static void Main(string[] args)
{
    CreateWebHostBuilder(args).Build().Run();
}

public static IWebHostBuilder CreateWebHostBuilder(string[] args) =>
    WebHost.CreateDefaultBuilder(args)
        .UseStartup<Startup>();
}
```

StartUp class:

```
public class Startup
{
    public IConfiguration Configuration { get; }

    public Startup(IConfiguration configuration)
    {
        Configuration = configuration;
    }

    // ConfigureServices, ConfigureDevelopmentServices, Configure methods.
}
```

asp.net-core-2.1

edited Oct 4 '18 at 18:13

asked Oct 4 '18 at 3:18



Xucong

31 5

Hi Xucong, just for funsies, put this in one of your views and run the app on windows and linux and see if there is any difference <p> ASPNETCORE_ENVIRONMENT = @hostingEnv.EnvironmentName</p> Linux is indeed case sensitive and would cause an issue if you do not have matching case. — Adam Vincent Oct 4 '18 at 18:52

2 Answers

Join **Stack Overflow** to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH



Google



2

As for my bug, it is actually due to a very stupid mistake:



In my appsettings.Development.json file, the "ConnectionStrings" is nested within "Logging" . Therefore, when the `ConfigureDevelopmentServices` method is invoked, it will not be able to find `ConnectionStrings` in the appsettings.Development.json file and will fall back to configurations specified in appsettings.json, which is configured for connecting to a MySQL database. Hence the errors shown in my problem statement.

This observation is inspired by [this GitHub issue](#).

answered Oct 8 '18 at 4:07



Xucong

31 5

0

Keep three nested files like `appsettings.json` , `appsettings.Development.json` and `appsettings.Production.json` . Then environment name from env. Mention which file to use at `Startup.cs` and in bash set environment using,

```
$ export ASPNETCORE_ENVIRONMENT=Development
```

Take the appropriate file depending on the environment. Replace the startup constructor with this.

```
public class Startup
{
    public Startup(IConfiguration configuration, IHostingEnvironment env)
    {
        var builder = new ConfigurationBuilder()
            .SetBasePath(env.ContentRootPath)
            .AddJsonFile("appsettings.json", optional: false, reloadOnChange: true)
            .AddJsonFile($"appsettings.{env.EnvironmentName}.json", optional: true)
            .AddEnvironmentVariables();

        Configuration = builder.Build();
    }

    public IConfiguration Configuration { get; }
```

Join **Stack Overflow** to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH



Google



edited Oct 5 '18 at 4:11

answered Oct 4 '18 at 3:59



t-prisar

891 1 9 20

Thanks for your suggestions. I am thinking about making the configuration more explicit as well. But I still do not understand why on Linux it behaves differently than on Windows or Mac. – [Xucong](#) Oct 4 '18 at 18:18

Also, I noticed that your StartUp class is setup a bit differently than mine. Are you using ASP.NET Core 2.0? Cos I believe they made some changes about the StartUp class and Program class in 2.1. I have updated my defaults from 2.1 in case you want to have a look. Thanks. – [Xucong](#) Oct 4 '18 at 18:23

i am not aware of the changes between 2.0 and 2.1 – [t-prisar](#) Oct 4 '18 at 18:52

The part i mentioned is same. – [t-prisar](#) Oct 5 '18 at 4:11

@Xucong Please change the startup constructor and export environment variable name while running in linux – [t-prisar](#) Oct 5 '18 at 4:13

Join **Stack Overflow** to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH



Google

