

# iLoy Group

## Task management system

---

### Requirements

- Visual Studio 2019 (link)
- Ms SQL Server

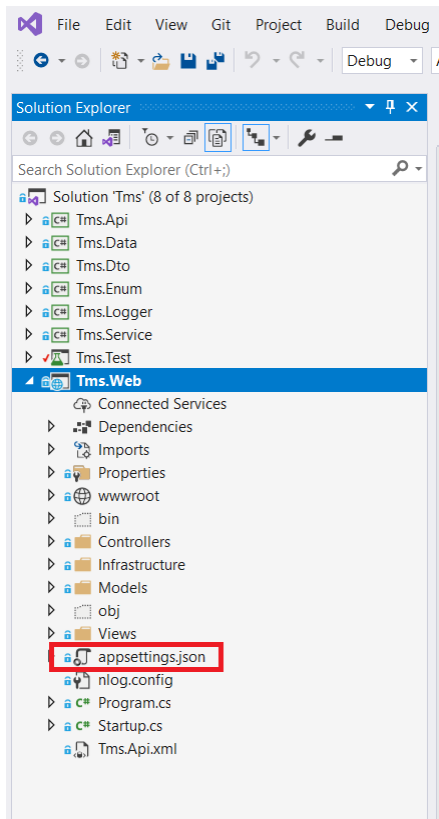
### External libraries

- CsvHelper, a library for reading and writing csv-files. Used for fast and easy to use
- NLog, is a logging platform for .NET with rich log routing and management capabilities. Used for debugging and logging exceptions. NLog.Config used for getting configuration from file "nlog.config"
- Swashbuckle.AspNetCore, swagger tools for documenting APIs built on ASP.NET Core. Used for building APIs list on the Swagger page: Home (page) -> Swagger (menu)
- NUnit and Moq used for unit-tests
- Newtonsoft.Json used for serialize and deserialize objects for tests
- AutoMapper, a convention-based object-object mapper
- Microsoft.EntityFrameworkCore.Tools is a Entity Framework Core Tools for the NuGet Package Manager Console, used for commands: add-migration, remove-migration, update-database and etc.

### Configuration database

#### Configuration file

Tms.Web project -> appsettings.json



## Database connection

Section "ConnectionStrings" -> "DefaultConnection":

"Data Source=localhost;Initial Catalog=Tms;Integrated Security=True;Persist Security Info=False"



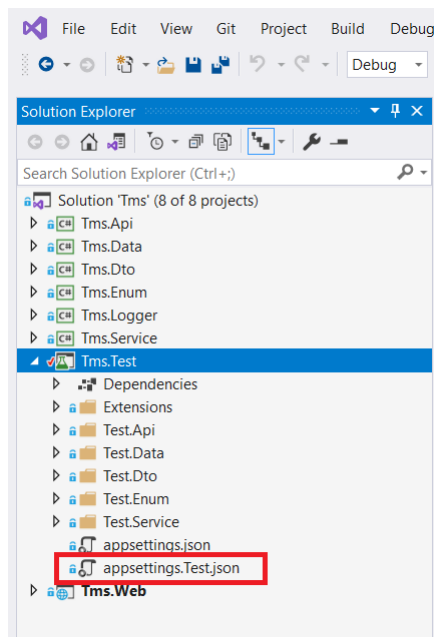
## Configuration tests

### Configuration file

-> appsettings.Test.json

The path to the configuration file is written in:

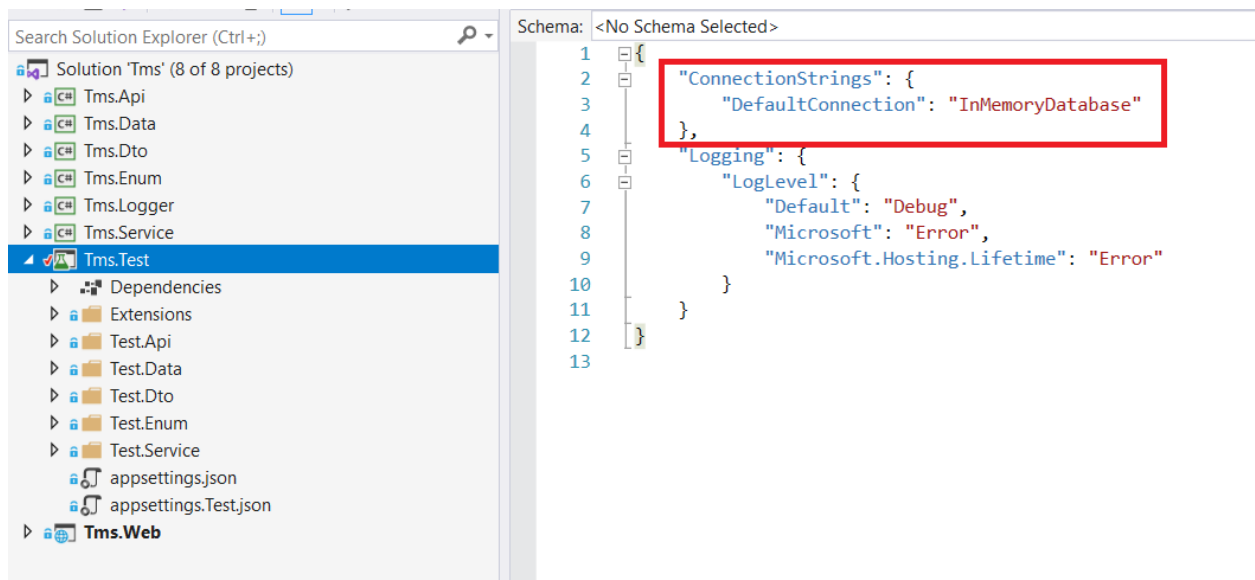
Tms.Test project -> TestServerFactory.cs -> the AppsettingsJsonFile constant



## Database connection

Section "ConnectionStrings" -> "DefaultConnection":

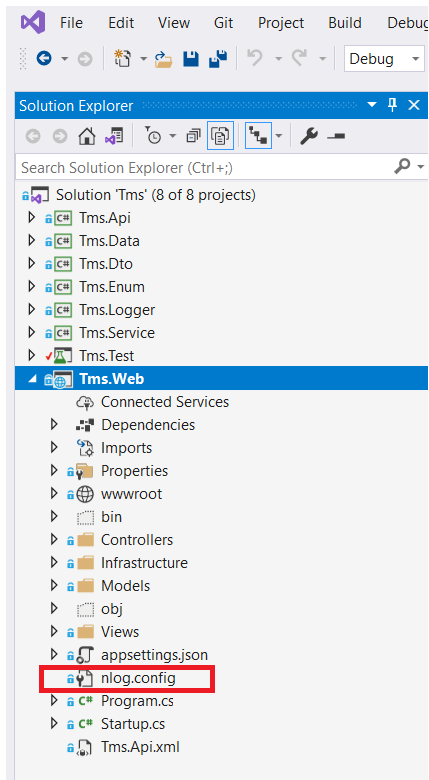
"InMemoryDatabase"



## Logging configuration

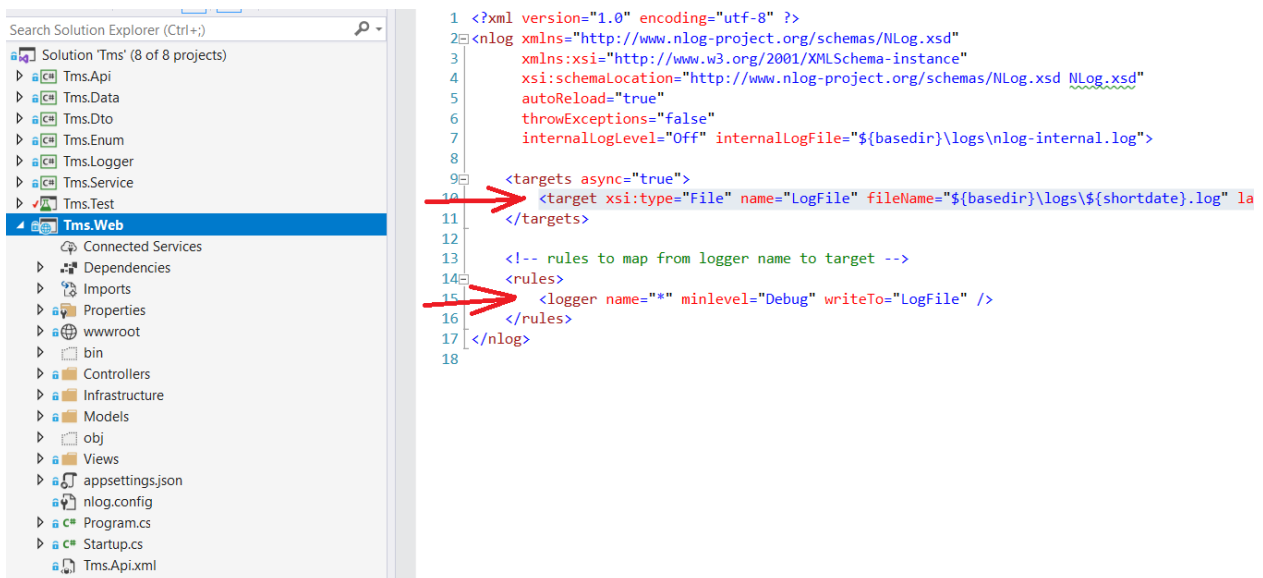
### Configuration file

Tms.Web project -> nlog.config



## File logging rules

1. Section "targets":  
`<target xsi:type="File" name="LogFile" fileName="${basedir}\logs\${shortdate}.log" layout="${longdate:universalTime=true} UTC|${level:uppercase=true}|${message}"/>`
2. Section "rules":  
`<logger name="*" minlevel="Debug" writeTo="LogFile" />`



## Logging level

Every appsettings.config and the same files contains the section "Logging" -> "LogLevel", for changing logging level. Preferred values:

“Logging”

“LogLevel”

“Default” = “Debug”

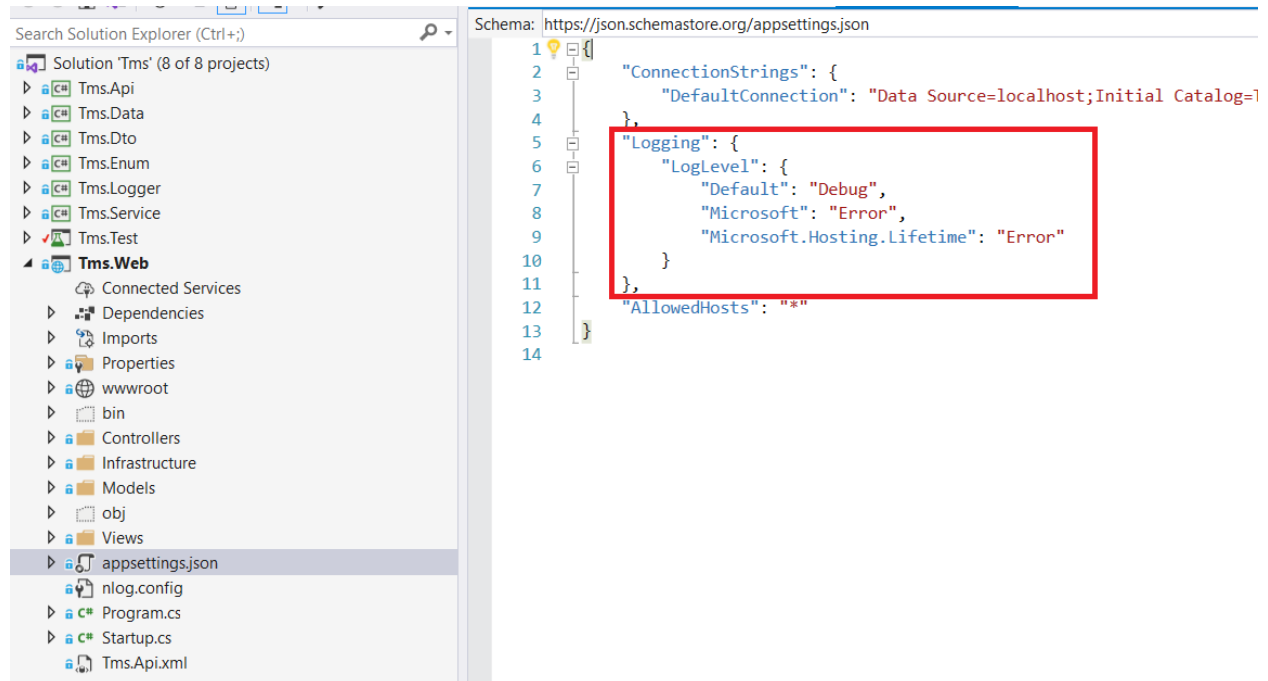
“Microsoft” = “Error”

“Microsoft.Hosting.Lifetime” = “Error”

“Hangfire” = “Error” (optional, if used)

“Microsoft.AspNetCore.SignalR” = “Error” (optional, if used)

“Microsoft.AspNetCore.Http.Connections” = “Error” (optional, if used)



## Migrations

Solution has the “migrations” feature. It is used for keeping the database in sync with the data model. Migrations performed while the Tms.Web project started (development or published). The mechanism is started in the classes:

Tms.Web -> Startup.cs -> void Configure -> app.UseMigration()

Tms.Data -> Context -> ApplicationDbContextUtils.cs -> void UseMigration

The “Seed data” process is used for populating a database with an initial set of demo data:

Tms.Data -> Context -> ApplicationDbContext.cs -> this.SeedData(modelBuilder)

Tms.Data -> Context -> ApplicationDbContext.cs -> void SeedData