



Appsettings Json

ARCTECH INFO PRIVATE
LIMITED

appsettings.json

appsettings.json file is an application configuration file used to store

- configuration settings such as database connections strings,
- any application scope global variables, etc.

See sample appsettings.json below

```
1 {
2   "Logging": {
3     "LogLevel": {
4       "Default": "Information",
5       "Microsoft": "Warning",
6       "Microsoft.Hosting.Lifetime": "Information"
7     }
8   },
9   "AllowedHosts": "*",
10  "ConnectionStrings": {
11    "DefaultConnectionString": "Data Source=.;...",
12    "MyOtherconnectionString": "Data Source=192.168.122.1;..."
13  },
14  "MyOtherSettings": {
15    "MinAgeAllowed": 21,
16    "MaxAgeAllowed": 60
17  }
18 }
```

IConfiguration

The **IConfiguration** service

- The **IConfiguration** service provides us the access to the appsettings.json data
- It is automatically available for DI in an asp.net core project
 - `_configuration.GetConnectionString("DefaultConnectionString")`
 - `_configuration["MyOtherSettings:MinAgeAllowed"];`

IOptions

The **IOptions** service

- The **IOptions** service is used to bind strongly typed options class to configuration section within appsettings.json
- It can be registered with the Asp.Net Core Service Container using `services.Configure<...>(...)`
- E.g., `services.Configure<MySettings>(Configuration.GetSection("MySettings"));`
 - Where MySettings is the class mapping to a section in appSettings.config called MySettings
 - E.g.,

```
"MySettings": {
  MinValue = 10,
  MaxValue = 100
}
```
 - ```
class MySettings
{
 public int MinValue {get; set;}
 public int MaxValue {get; set;}
}
```
- Inject the setting into any controller or service as
  - `IOptions<MySettings> mySettingsAccessor`
  - `_mySettings = mySettingsAccessor.Value;`