

FamilyCalendar – Deploy + EnvironmentSettings

Environments

Som det låter på namnet så handlar detta om olika miljöer. När man sitter och utvecklar en applikation på sin dator så kallas ju detta för en lokal miljö eller utvecklingsmiljö.

Sen har man en produktionsmiljö där applikationen snurrar i produktion. Och i många fall så har man kanske en testmiljö eller en stagemiljö.

Då du är på en upplärningsstig på din karriär så tänker jag att jag svarar på frågor du inte ens ställt, utan jag försöker minnas vad jag undrade över när jag började när det slängs begrepp som stage exempelvis och hur en sån skiljer sig från en testmiljö.

En testmiljö är en ren testapplikation. I detta projekt så skulle en testmiljö av appen vara att man faktiskt testar ny logik och se så det lilar, typ att man gjort en databasförändring och man vill se så det lilar som det ska. Medan en Stage kan vara uppsatt för ett speciellt behov, kanske en mer replika av produktion med lite mer hårdare tester.

Ofast på större bolag, säg Regionen i Nyköping. Dom tömmer allt i sin testmiljö varje natt och kopierar in gårdagen från prod till testmiljön, varje natt, medan en stagemiljö kan förbli med gammal data då man kanske satt upp en viss typ av data som man vill ska vara bestående.

Våra Environments

Jag ser det som att vi har en DEV/Lokal och en Produktion. Just nu kommer produktion vara den vi kallar för calendar-dev.localhosts.se tills vi skaffar ny domän. Medan DEV är mer lokalt när vi sitter och utvecklar.

Då har jag sett att på webbhotellet så kan inte databasen ligga på samma ställe som den gör när vi utvecklare. Och det här är ju något som ett specificerat i AppSettings.json

```
1  {
2    "Logging": {
3      "LogLevel": {
4        "Default": "Information",
5        "Microsoft.AspNetCore": "Warning"
6      }
7    },
8    "AllowedHosts": "*",
9    "ConnectionStrings": {
10     "DefaultConnection": "Data Source=../FamilyCalendar.Infrastructure/FamilyCalendar.db",
11     "ApplicationDbContextConnection": "Server=(localdb)\\mssqllocaldb;Database=FamilyCalendar.Web;Trusted_Connection=True;MultipleActiveResultSets=true"
12   }
13 }
```

Den här sökvägen till databasen behöver vara annorlunda när vi laddar upp till webbhotellet, då behöver den se ut så här:

```
1  {
2    "Logging": {
3      "LogLevel": {
4        "Default": "Information",
5        "Microsoft.AspNetCore": "Warning"
6      }
7    },
8    "AllowedHosts": "*",
9    "ConnectionStrings": {
10     "DefaultConnection": "Data Source=APP_DATA/FamilyCalendar.db",
11     "ApplicationDbContextConnection": "Server=(localdb)\\mssqllocaldb;Database=FamilyCalendar.Web;Trusted_Connection=True;MultipleActiveResultSets=true"
12   }
13 }
```

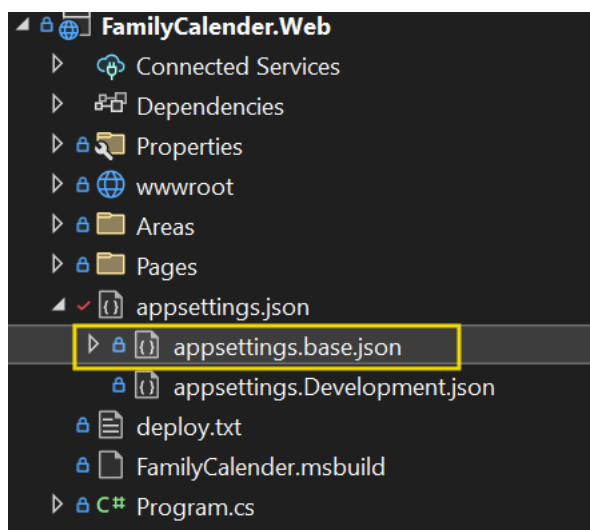
Och för att manuellt ändra detta när man ska utveckla lokalt och när man ska driftsätta så har jag lyft in lite hjälpskript för att kunna ändra dessa. Och det görs med två skript i repositoryt.

Name	Date modified	Type	Size
.git	2025-03-25 20:06	File folder	
.github	2025-03-25 10:42	File folder	
.vs	2025-03-25 11:27	File folder	
dist	2025-03-25 20:01	File folder	
Environments	2025-03-25 11:11	File folder	
FamilyCalender.Core	2025-03-25 10:42	File folder	
FamilyCalender.Infrastructure	2025-03-25 11:07	File folder	
FamilyCalender.Tests	2025-03-21 14:55	File folder	
FamilyCalender.Web	2025-03-26 14:52	File folder	
ThirdParty	2025-03-25 11:12	File folder	
.gitattributes	2025-03-21 14:55	GITATTRIBUTES File	3 KB
.gitignore	2025-03-25 20:06	Git Ignore Source File	7 KB
build_prod.bat	2025-03-25 11:22	Windows Batch File	1 KB
FamilyCalender.sln	2025-03-21 14:55	Visual Studio Solution	3 KB
generate_dev_config.bat	2025-03-25 11:20	Windows Batch File	1 KB
generate_prod_config.bat	2025-03-25 12:56	Windows Batch File	1 KB
README.md	2025-03-21 14:55	MD File	1 KB

Dubbeltklickar man på filen generate_dev_configs.bat så kommer connectionsträngen i appsettings.json att bli som vi vill ha den för utveckling och kör man generate_prod_configs.bat så kommer appsettings.json bli ämnad för hur webbhotellet behöver den.

Så hur funkar det?

Om vi tittar i visual studio så kan vi se att det finns en appsettings.base.json



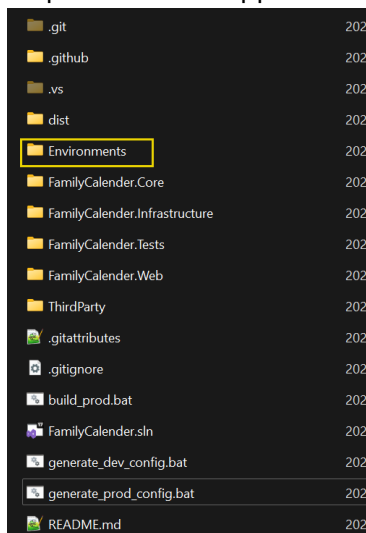
Det här är en fil jag skapat och den ser nästan likadan ut som appsettings.json men med en skillnad att den har en template för connectionsträngen till databasen

```
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*",
  "ConnectionStrings": {
    "DefaultConnection": "${DbConnectionString}",
    "ApplicationDbContextConnection": "Server=(localdb)\\mssqllocaldb;Database=FamilyCalender.Web;Trusted_Connection=True;MultipleActiveResultSets=true"
  }
}
```

Den här strängen "\${DbConnectionString}" byts ut med vad som lämpar sig för dev/lokal och för produktion. Så när vi klickar på generate_dev_configs.bat så kommer "\${DbConnectionString}" att ersättas med hur connectionsträngen ska se ut när vi utvecklar och vice versa för produktion.

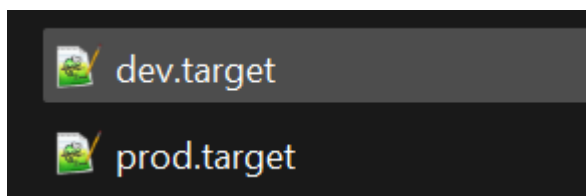
Så vart kommer värdet ifrån dom "\${DbConnectionString}" blir?

I repot finns en mapp som heter environments



A screenshot of a file explorer window showing a project directory. The 'environments' folder is highlighted with a yellow box. Other files and folders visible include .git, .github, .vs, dist, FamilyCalender.Core, FamilyCalender.Infrastructure, FamilyCalender.Tests, FamilyCalender.Web, ThirdParty, .gitattributes, .gitignore, build_prod.bat, FamilyCalender.sln, generate_dev_config.bat, generate_prod_config.bat, and README.md.

Och i denna finns det två filer, dev.target och prod.target



A screenshot showing two file icons with a notepad icon. The top icon is labeled 'dev.target' and the bottom icon is labeled 'prod.target'.

Tittar man i dev.targets med notepad++ så ser den ut enligt följande

```
<?xml version="1.0" encoding="utf-8"?>
<Project xmlns="http://schemas.microsoft.com/developer/msbuild/2003">
  <PropertyGroup>
    <DbConnectionString>Data Source=../FamilyCalender.Infrastructure/FamilyCalender.db</DbConnectionString>
  </PropertyGroup>
</Project>
```

Det är en xml baserad fil som har ett fält som heter precis som appsettings.base.json med värdet <DbConnectionString> förutom det här med dollartecknet. Men det som står i denna tag, det kommer ersätta "\${DbConnectionString}" i appsettings.base.json och i sin tur bli en

appsetting.json. Generate skripten ersätter alltid befintliga appsettings.json när man kör dom. Det betyder att om min appsettings.json ser ut så här

```
appsettings.json*  X
Schema: https://json.schemastore.org/appsettings.json
1 {
2   "Logging": {
3     "LogLevel": {
4       "Default": "Information",
5       "Microsoft.AspNetCore": "Warning"
6     }
7   },
8   "AllowedHosts": "*",
9   "ConnectionStrings": {
10
11   }
12 }
13 }
```

Och så kör jag generate_dev_configs.bat så kommer den efter att skriptet är klart, se ut så här igen

```
appsettings.json  X
Schema: https://json.schemastore.org/appsettings.json
1 {
2   "Logging": {
3     "LogLevel": {
4       "Default": "Information",
5       "Microsoft.AspNetCore": "Warning"
6     }
7   },
8   "AllowedHosts": "*",
9   "ConnectionStrings": {
10     "DefaultConnection": "Data Source=../FamilyCalender.Infrastructure/FamilyCalender.db",
11     "ApplicationDbContextConnection": "Server=(localdb)\\mssqllocaldb;Database=FamilyCalender.Web;Trusted_Connection=True;MultipleActiveResultSets=true"
12   }
13 }
```

För att den tar allt som står i appsettings.base.json, ersätter eventuella värden som den kan ersätta med data ifrån dev.targets i detta fall.

Tittar man i prod.targets i environment mappen så ser den ut så här

```
dev.target  X  prod.target  X
1 <?xml version="1.0" encoding="utf-8"?>
2 <Project xmlns="http://schemas.microsoft.com/developer/msbuild/2003">
3   <PropertyGroup>
4
5     <DbConnectionString>Data Source=APP_DATA/FamilyCalender.db</DbConnectionString>
6   </PropertyGroup>
7 </Project>
```

Den tycker alltså att DbConnectionString ska vara APP_DATA/... istället. Och det är för att det numera finns en mapp på FTP servern som heter APP_DATA som innehåller databasen i produktion

Tittar man på skriptet generate_dev_configs.bat eller generate_prod_configs.bat i notepad++ så ser det ut så här

```
dev.target  X  prod.target  X  generate_dev_config.bat  X
1 @echo off
2 call ThirdParty\VoidPointer.Build\hMSBuild_minified.bat FamilyCalender.Web\FamilyCalender.msbuild /p:Env=dev /t:GenerateConfig
3 pause
4
```

Scriptet anropar en bat fil i mappen ThirdParty/VoidPointer.Build som heter hMSBuild_minified.Bat. Som säger åt det skriptet att använda en MSBuild fil som ligger i mappen FamilyCalender.Web (som jag skapat). Säger åt att Environment (p:ENV=Dev eller prod) som säger åt ett jobb att Generera configs (t:GenerateConfig)

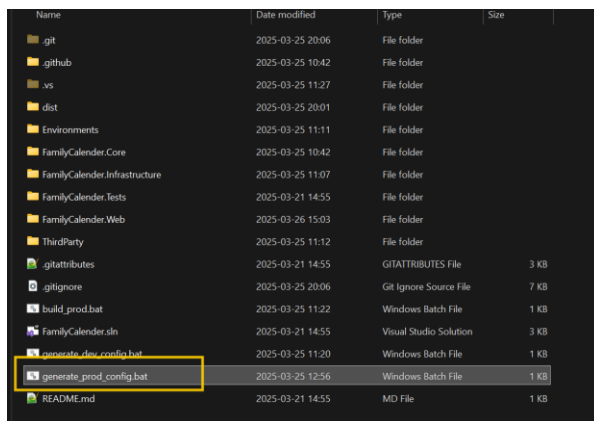
Det här ett verktyg som vi använder som är utvecklat av ett företag som heter VoidPointer där jag tidigare jobbat och det är för att enkelt kunna ha olika inställningar beroende på om man ska utveckla lokalt, eller driftsätta en testmiljö, eller produktion etc.

Här kan man ju ha hur många inställningar som helst. Men det är alltså inställningar som är miljöspecifika och som applikationen behöver ha vetskap om och som kan skilja sig beroende på om man sitter lokalt och kanske har sin egna databas eller en databas någon annanstans.

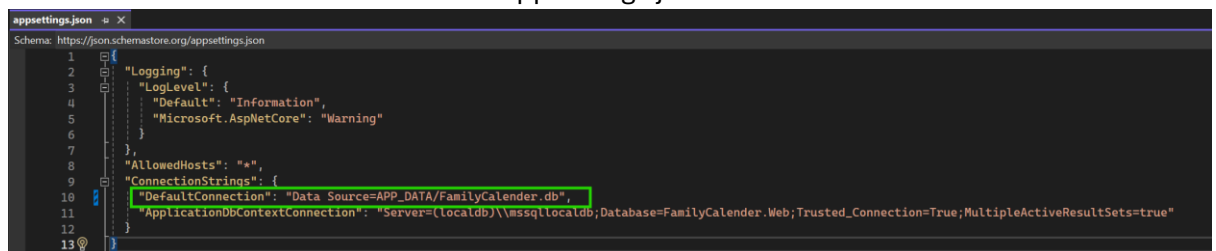
Det här känns normalt sett "wooo wtf" så no preasure, låt det sjunka in och att läsa sig till hur det funkar är lika svårt som att skriva hur det funkar så det här får vi gå igenom någon dag.

Så hur deployar man

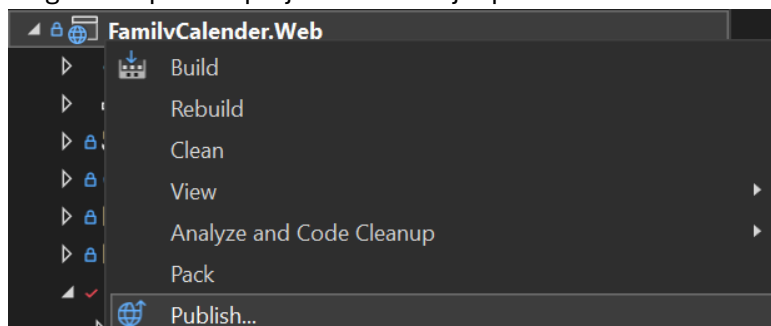
1. Först är det viktigt att man genererar rätt configuration så som webshotellet vill ha det så det första man gör, det är att dubbelklicka på generate_prod_configs.bat



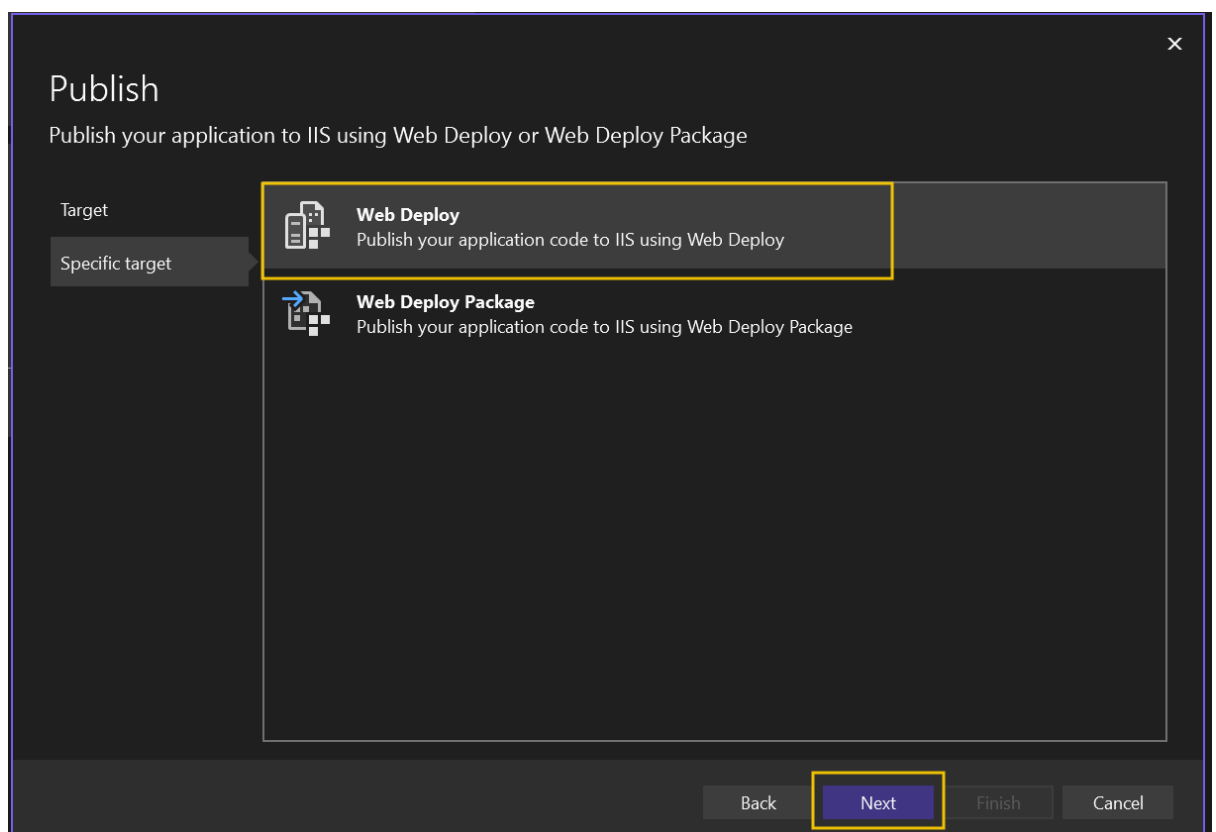
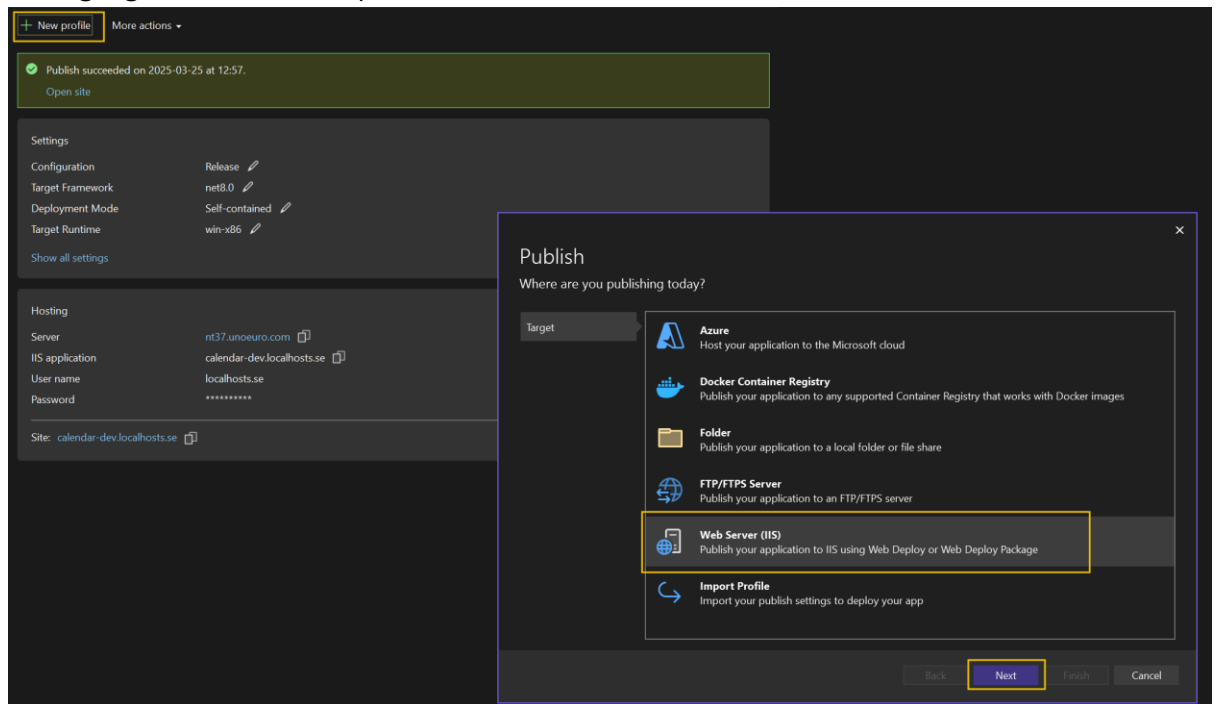
2. Därefter är det alltid bra att verifiera att appsettings.json ser korrekt ut



3. Högerklick på Webprojektet och väljer publish



4. Först gången får man klicka på New Profile



5.

6. Fyll i uppgifter, tänk på stavning

Publish
Configure web server (IIS) connection

Target: Server
nt37.unoeuro.com

Specific target: Site name
calendar-dev.localhosts.se

IIS Connection: Destination URL
calendar-dev.localhosts.se

User name
localhosts.se

Password
••••••••

☐ Save password

Validate Connection

Back Next Finish Cancel

Password: 30391288

7. Klicka gärna på validate Connection så du ser att uppgifterna ovan är rätt
8. Klicka på Finish
9. Klicka på Pennan

IISProfile1.pubxml
Web Deploy Publish

+ New profile More actions

Ready to publish.

Settings

Configuration Release

Deployment Mode Framework-dependent

Target Runtime Portable

Show all settings

Hosting

Server nt37.unoeuro.com

IIS application calendar-dev.localhosts.se

User name localhosts.se

Password *****

Site: calendar-dev.localhosts.se

10.

11.

Publish

Connection

IISProfile1 *

Configuration: Release

Target Framework: net8.0

Deployment Mode: Self-contained

[Learn about deployment modes](#)

Target Runtime: win-x86

File Publish Options

- ☐ Remove additional files at destination
- ☐ Produce single file
- ☐ Enable ReadyToRun compilation
- ☐ Trim unused code

Databases

Discovering Data Contexts...

< Prev Next > **Save** Cancel

12. Spara och klicka på Publish

IISProfile1.pubxml

Web Deploy

Publish

+ New profile More actions

Ready to publish.

Settings

Configuration	Release
Target Framework	net8.0
Deployment Mode	Self-contained
Target Runtime	win-x86

Show all settings

Hosting

Server	nt37.unoeuro.com
IIS application	calendar-dev.localhosts.se
User name	localhosts.se
Password	*****

Site: calendar-dev.localhosts.se