

EF Core 反向工程自訂義T4 Template Model 轉換Enum

🕒 Created	@2024年6月2日 上午1:15
🏷️ Tags	

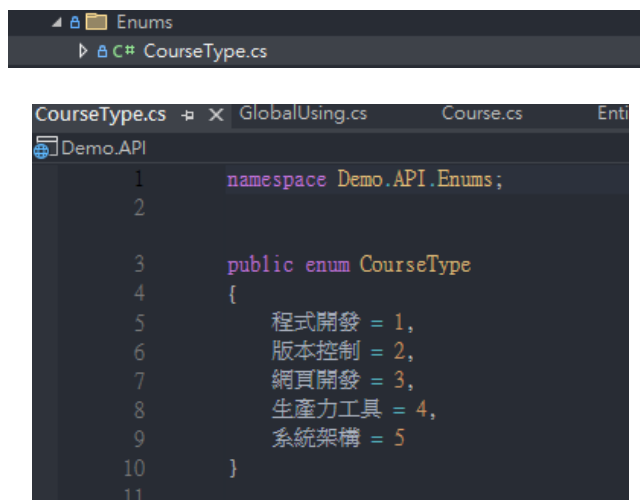
1. 我先在保哥的範本資料庫Course新增欄位CourseType

	資料行名稱	資料類型	允許 Null
🔑	CourseID	int	<input type="checkbox"/>
	Title	nvarchar(50)	<input checked="" type="checkbox"/>
	Credits	int	<input type="checkbox"/>
	DepartmentID	int	<input type="checkbox"/>
▶	CourseType	int	<input type="checkbox"/>

資料變更為

	CourseID	Title	Credits	DepartmentID	CourseType
1	1	Entity Framework 6 開發實戰	1	5	1
2	2	Git新手入門	1	5	2
3	3	Git進階版控流程	2	5	2
4	4	ASP.NET MVC 5 開發實戰	5	1	1
5	5	Javascript 新手入門	1	1	3
6	6	ASP.NET Core 3 開發實戰：從入門到進階	5	1	1
7	7	LINQ 語法入門	3	1	1
8	8	C# 開發實戰：非同步程式開發技巧	3	5	1
9	9	NET / C# 開發實戰：掌握相依性注入的觀念與開發技巧	4	5	1
10	10	Angular 新手開發練功坊	5	5	3
11	12	十項超實用 Excel 技能 — 節省 50% 的工作效率	2	2	4
12	13	Windows Containers 微服務架構實戰	4	13	5

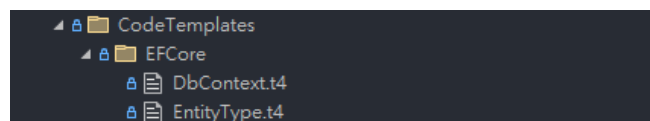
2. 在專案新增Enums 資料夾，並新增CourseType.cs



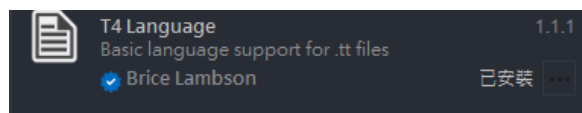
3. 利用EF Core Power Tools，進行反向代理，並且點選使用模板自訂程式碼



3. 會產生以下資料夾與檔案，附檔名為T4



4. 安裝延伸模組 T4，這樣T4 語法Code才會變色，不然很難Debug



裝完會變成這樣

```

<#if hostSpecific == true >
<#include name="Microsoft.EntityFrameworkCore" >
<#include name="Microsoft.EntityFrameworkCore.Design" >
<#include name="Microsoft.EntityFrameworkCore.Relational" >
<#include name="Microsoft.Extensions.DependencyInjection.Abstractions" >
<#include name="EntityType" type="Microsoft.EntityFrameworkCore.Metadata.IEntityType" >
<#include name="Options" type="Microsoft.EntityFrameworkCore.Scaffolding.ModelCodeGenerationOptions" >
<#include name="NamespaceInt" type="System.String" >
<#include namespace="System.Collections.Generic" >
<#include namespace="System.ComponentModel.DataAnnotations" >
<#include namespace="System.Linq" >
<#include namespace="System.Text" >
<#include namespace="Microsoft.EntityFrameworkCore" >
<#include namespace="Microsoft.EntityFrameworkCore.Design" >
<#include namespace="Microsoft.Extensions.DependencyInjection" >
</if>

// Template version: 800 - please do NOT remove this line
if (EntityType.IsSimpleManyToManyJoinEntityType())
{
    // Don't scaffold these
    return "";
}

var renames = new List<(string EntityTypeName, string PropertyName, string NewPropertyName)>
{
    ("PlatformIdentifierMapping", "PlatformPlatformType", "DestinationPlatform"),
    ("PlatformIdentifierMapping", "PlatformPlatformTypeNavigation", "SourcePlatform"),

```

5. 在作者GitHub的 Issue 有新增新的T4 模板 <https://github.com/ErikEJ/EFCorePowerTools/issues/1499>



ErikEJ commented on Jan 8

Owner

Author

...

I have uploaded the start of a "Power" template, with the following features:

- Enum support
- Property renaming
- File header

<https://github.com/ErikEJ/EFCorePowerTools/tree/master/samples/CodeTemplates/EFCore>



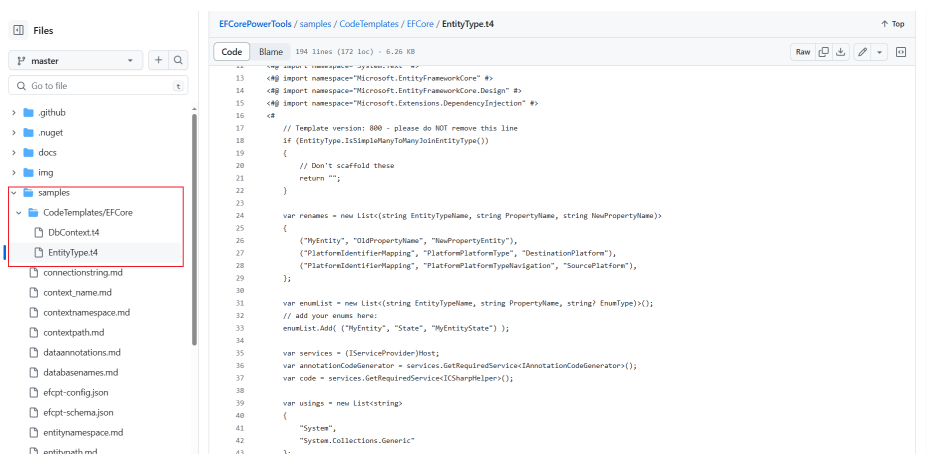
1

在這個連結

<https://github.com/ErikEJ/EFCorePowerTools/blob/master/samples/CodeTemplates/EFCore/EntityType.t4>

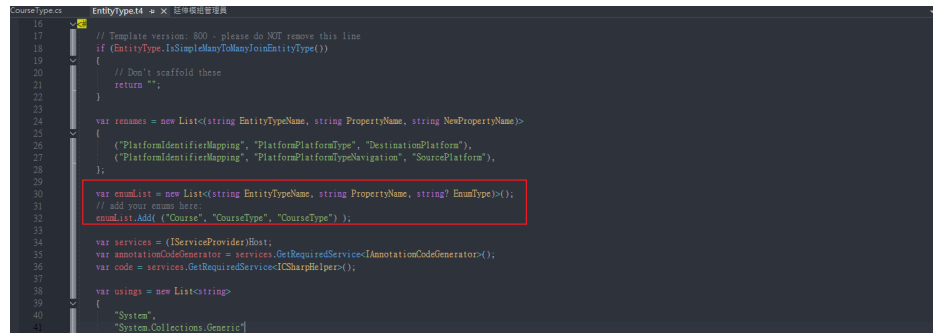
我們直接去複製這兩份檔案 EntityType.t4 跟 DbContext.t4 並覆蓋剛剛產生的 EntityType.t4 跟 DbContext.t4

差異就是作者新增的這三個功能我有請AI解釋一下，<https://chatgpt.com/share/46710a0e-5e10-476b-8e06-6d4bad01f6e5>



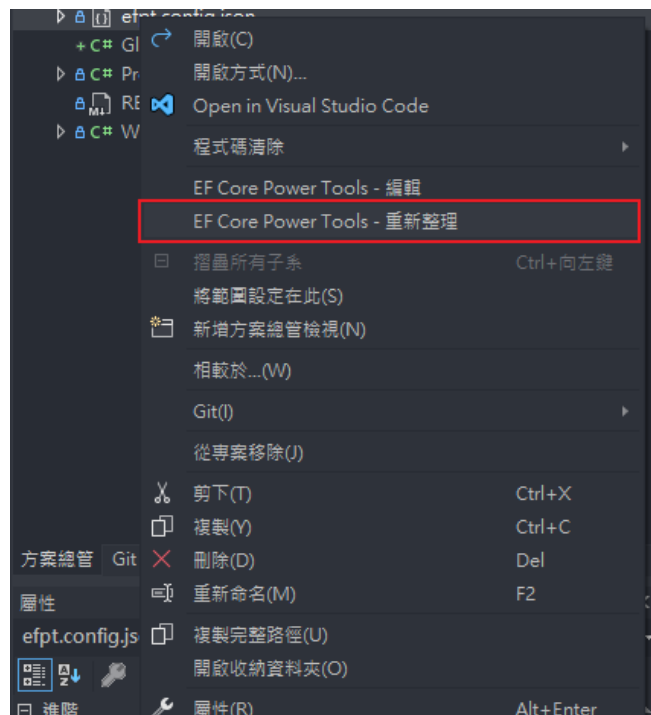
6. 我這次需求是型別要轉換EnumType，作者很貼心幫我們註解區塊在哪🤔

```
// (Entity名稱, Entity 內屬性名稱, Enum 名稱)
enumList.Add( ("Course", "CourseType", "CourseType") );
```



7. 最後我們在efpt.config.json 右鍵點選 EF Core Power Tools - 重新整理即可，AI一直跟我說儲存 EntityType.t4就可以自動更新🤔，答案不是，我後來在YT找到作者有介紹怎麼更新

<https://www.youtube.com/watch?v=PUexkGWErNk>



8. 最後有更新完成

```
4 [using System.Collections.Generic;
5
6 namespace Demo.API.Models;
7
8 0 個參考
9 public partial class Course
10 {
11     1 個參考
12     public int CourseID { get; set; }
13
14     1 個參考
15     public string Title { get; set; }
16
17     0 個參考
18     public int Credits { get; set; }
19
20     3 個參考
21     public int DepartmentID { get; set; }
22
23     0 個參考
24     public CourseType CourseType { get; set; }
25
26     1 個參考
27     public virtual Department Department { get; set; }
28
29     1 個參考
30     public virtual ICollection<Enrollment> Enrollment { get; set; } = new List<Enrollment>();
31
32     1 個參考
33     public virtual ICollection<Person> Instructor { get; set; } = new List<Person>();
34 }
```

測試一下代碼如何，確實已經轉換成enum

```
[HttpGet("{id}")]
0 個參考
public string Get(int id)
{
    var course = _context.Course.Find(id);
    if (course.CourseType == CourseType.程式開發)
    {
        return "value";
    }
}
```

並確認一下API有無問題

Status: 200 OK Time: 2181.12 ms Size: 1869 bytes

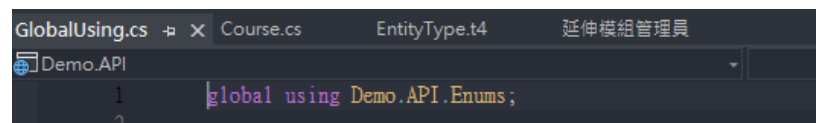
Formatted Raw Headers Request

Body

application/json; charset=utf-8, 1869 bytes

```
[
  {
    "courseID": 1,
    "title": "Entity Framework 6 開發實戰",
    "credits": 1,
    "departmentID": 5,
    "courseType": 1,
    "department": null,
    "enrollment": [],
    "instructor": []
  },
  {
    "courseID": 2,
    "title": "Git新手入門",
    "credits": 1,
    "departmentID": 5,
    "courseType": 2,
    "department": null,
    "enrollment": [],
    "instructor": []
  }
]
```

貼心跟大家說一下，可以加命名空間儲存在GlobalUsing檔案中，這樣就不用每個檔還要using



The screenshot shows the Visual Studio IDE with several tabs open: GlobalUsing.cs, Course.cs, EntityType.t4, and 延伸模組管理員. The GlobalUsing.cs file is selected, and its content is visible. It contains a single line of code: `global using Demo.API.Enums;`. The line is numbered 1 in the left margin.