
TP 3 : PRESTATIONS SOINS

CREATION D'UN PACKAGE ET AJOUT DE FONCTIONNALITE

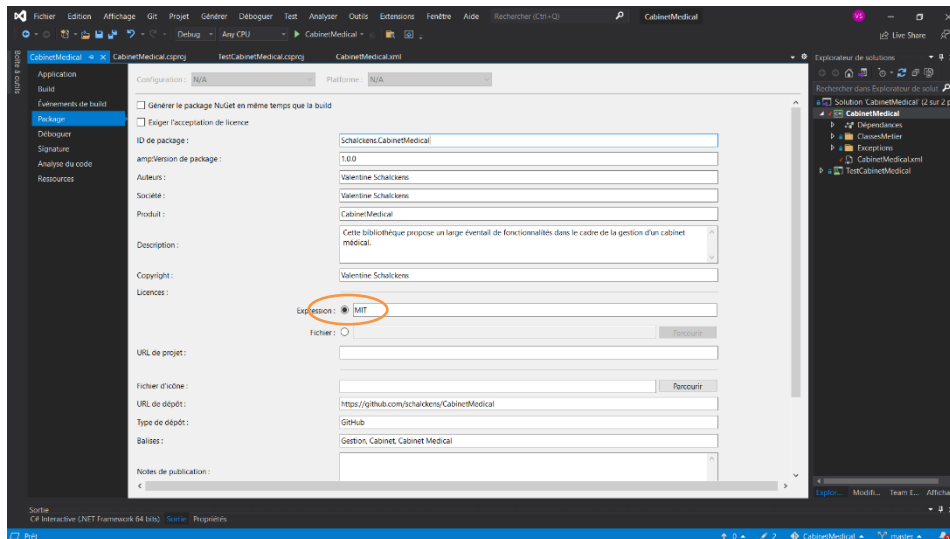
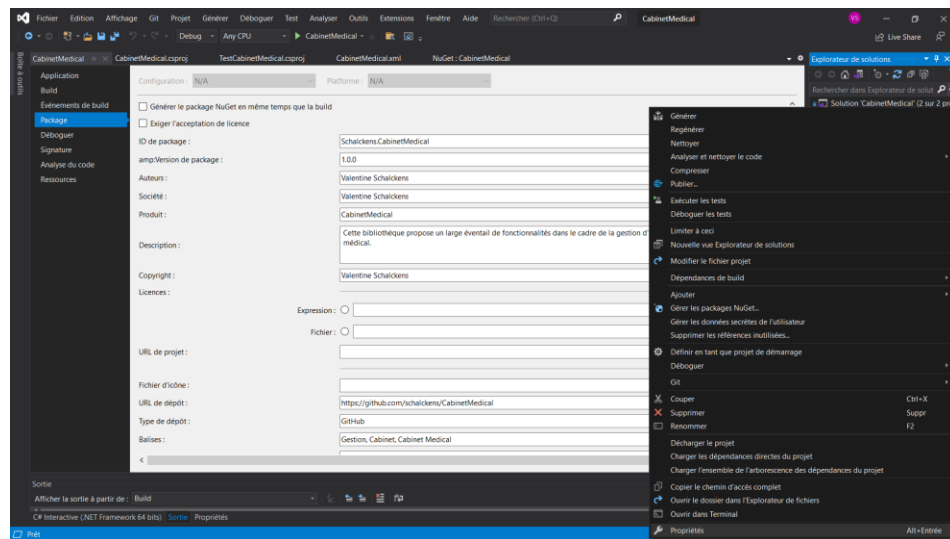


SOMMAIRE

Tutoriel: Création et publication d'un projet en tant que package NuGet.....	2
Partie I: Test du package créer	6
Partie II: Ajout d'une fonctionnalité (+Tutoriel)	8
Partie III: Mise en oeuvre de la nouvelle fonctionnalité.....	11

TUTORIEL: CREATION ET PUBLICATION D'UN PROJET EN TANT QUE PACKAGE NUGET.

1. Création d'un compte sur le site de [NuGet](https://www.nuget.org/).
2. Se rendre dans les propriétés du projet afin de compléter les métadonnées nécessaires à la création d'un package NuGet.

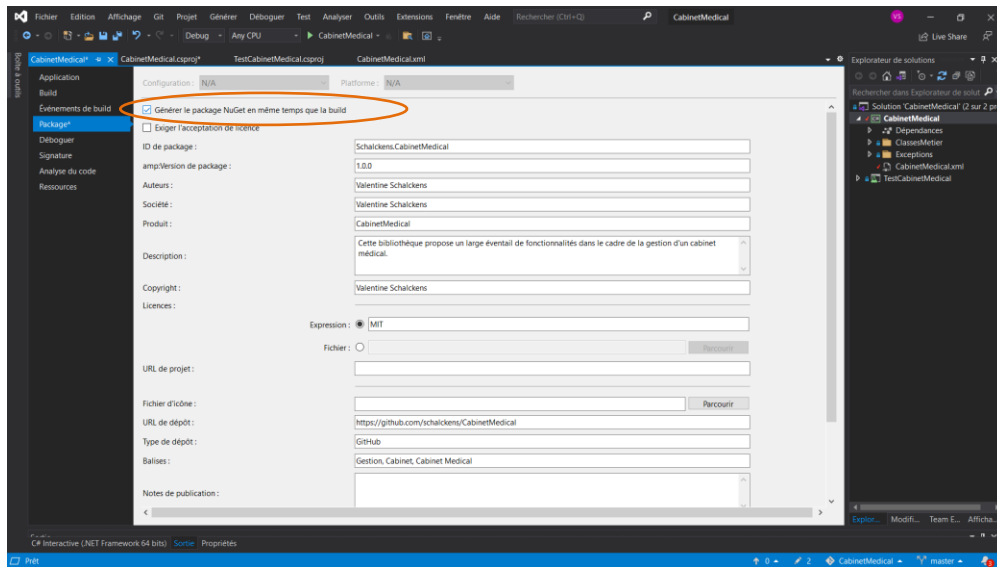


Ici on ajoute une licence afin de ne pas avoir d'erreur lors de l'ajout sur le site NuGet.

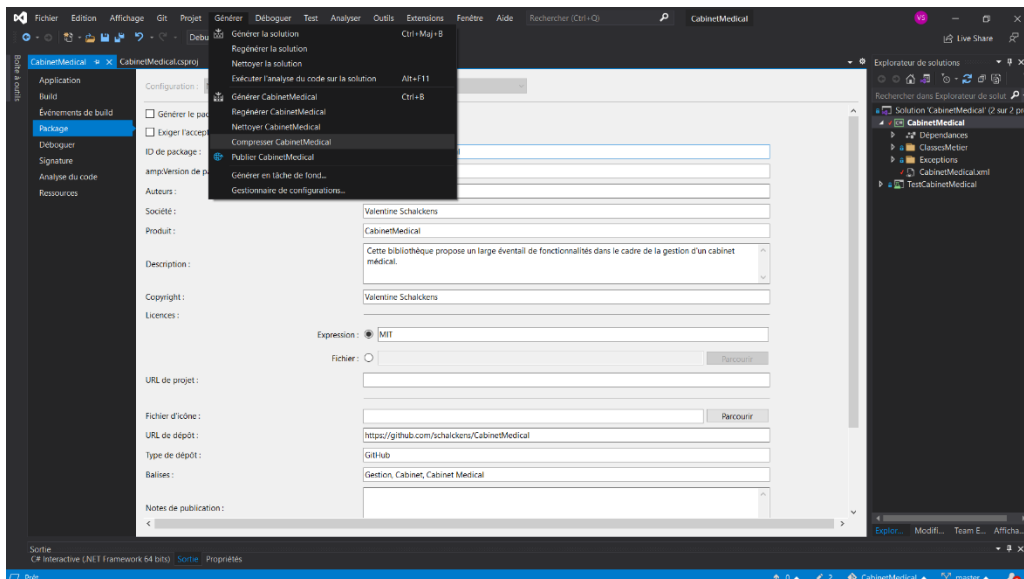
J'ai choisi la licence MIT car elle est très peu contraignante ce qui est parfait pour tester la mécanique de publication d'un package NuGet et me permettra de le réutiliser par la suite facilement.

3. Compresser le projet, pour cela deux manières :

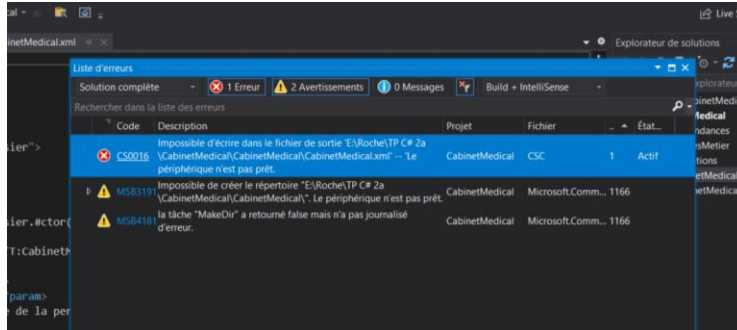
- a. Automatique à chaque génération du projet : cocher la case « Générer le package NuGet en même temps que la build » .



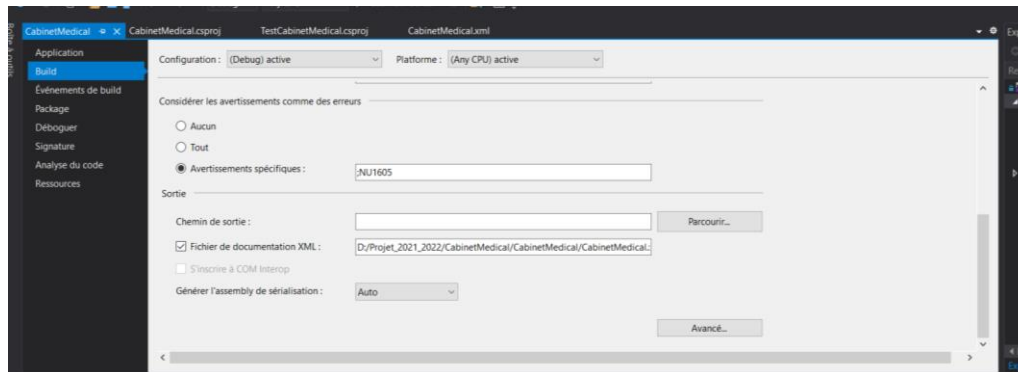
- b. De manière manuelle : aller dans « Générer → Compresser <nom du projet> ».



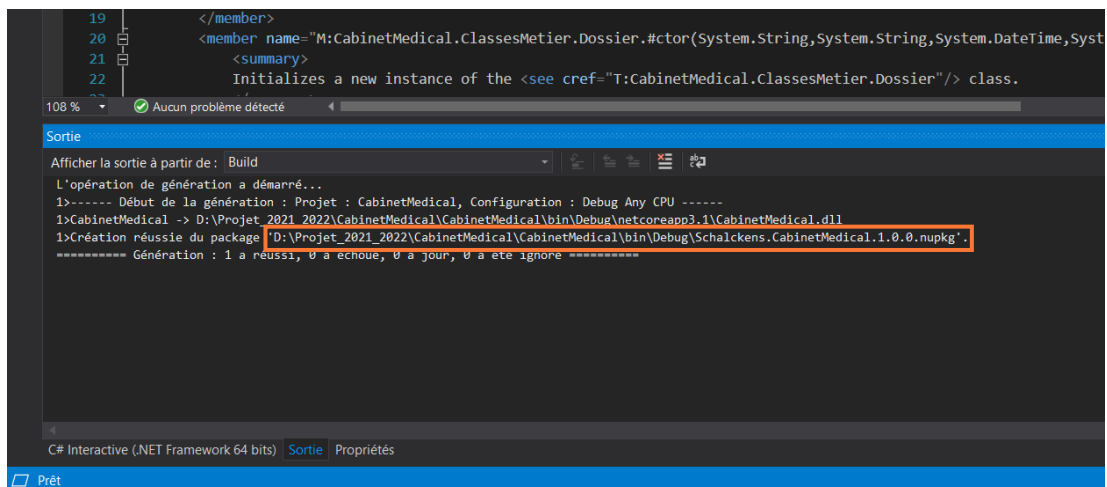
Problème rencontré et résolu :



A cause de la perte de ma clé USB, j'ai dû faire un clone de mon projet depuis GitHub pour récupérer mon projet. L'URL du projet a donc changé et cela m'a soulevé une erreur. Pour la résoudre je suis allé dans les propriétés du projet dans « Build » et j'ai modifié le chemin d'accès au fichier XML.



4. Si réussite, elle s'affiche dans l'onglet « Output » de Visual Studio :



5. Se rendre sur NuGet dans la rubrique « Upload », nous pouvons maintenant lui donner le chemin d'accès au package que nous venons de créer. Après vérification des informations nous pouvons cliquer sur « Submit » en bas de la page :

The screenshot shows the NuGet.org 'Upload' page. The browser tabs include 'Soms_Nuget_2021_V...', 'NuGet Gallery Upload...', 'Create and publish a...', 'NuGet pack and rest...', 'nupkg File Referen...', 'NuGet.Client/nupkg...', and 'Output Window - Vi...'. The page header has 'nuget' and 'schalckens'.

Search for packages...

⌕ Packages > Upload

Your package file will be uploaded and on the NuGet Gallery server (<https://www.nuget.org>).

To learn more about authoring great packages, view our [Best Practices](#) page.

Upload

Schalckens.CabinetMedical.1.0.0.nupkg [Browse...](#)

Verify

Package ID
Schalckens.CabinetMedical

Version
1.0.0

Owner
schalckens

Minimum NuGet Client Version
(none specified)

Add Markdown Documentation here...

[Preview](#)

Submit

[Cancel](#)

Submit

Cancel

Contact
Got questions about NuGet or the NuGet Gallery?

Status
Find out the service status of NuGet.org and its related services.

FAQ
Read the Frequently Asked Questions about NuGet and see if your question made the list.

Félicitation ! Le Package est maintenant créé et publié sur NuGet !

The screenshot shows the NuGet.org package page for 'Schalckens.CabinetMedical' version 1.0.0. The browser tabs are the same as in the previous screenshot. The page header has 'nuget' and 'schalckens'.

Search for packages...

ⓘ You successfully uploaded Schalckens.CabinetMedical 1.0.0.

Schalckens.CabinetMedical 1.0.0

This package has not been published yet. It will appear in search results and will be available for install/restore after both validation and indexing are complete. Package validation and indexing may take up to an hour. [Read more.](#)

[README](#) [Dependencies](#) [Used By](#) [Versions](#)

The package description is shown below. Please update your package to include a README.

Cette bibliothèque propose un large éventail de fonctionnalités dans le cadre de la gestion d'un cabinet médical.

Downloads [Full stats](#)

Total 0

Current version 0

Per day average 0

About

Last updated a minute ago

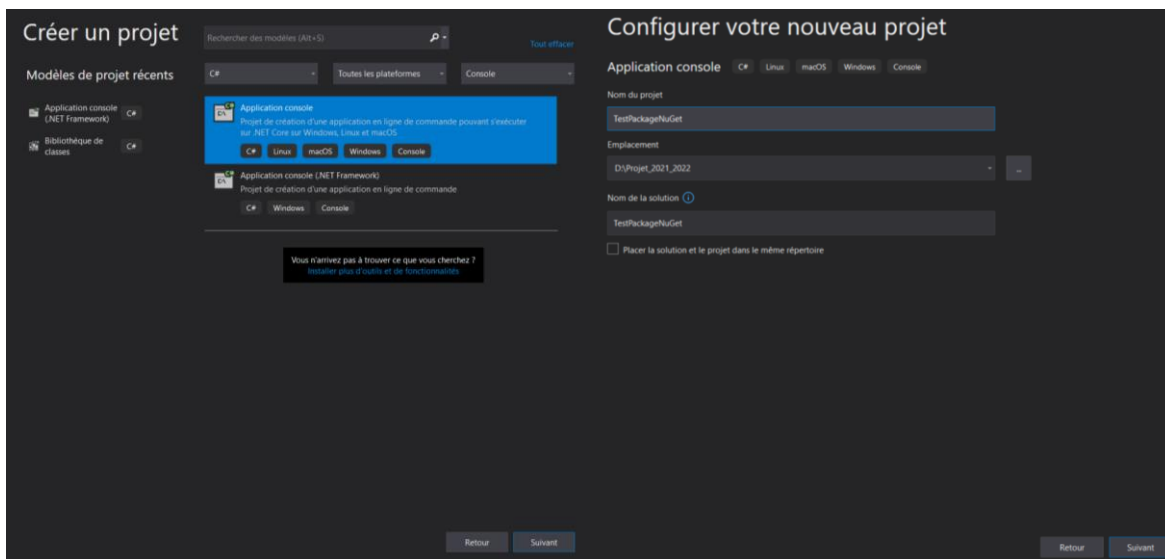
Source repository

MIT license

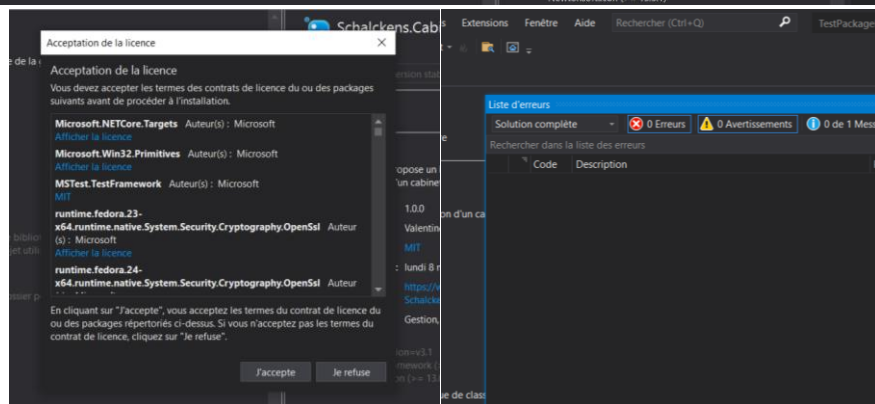
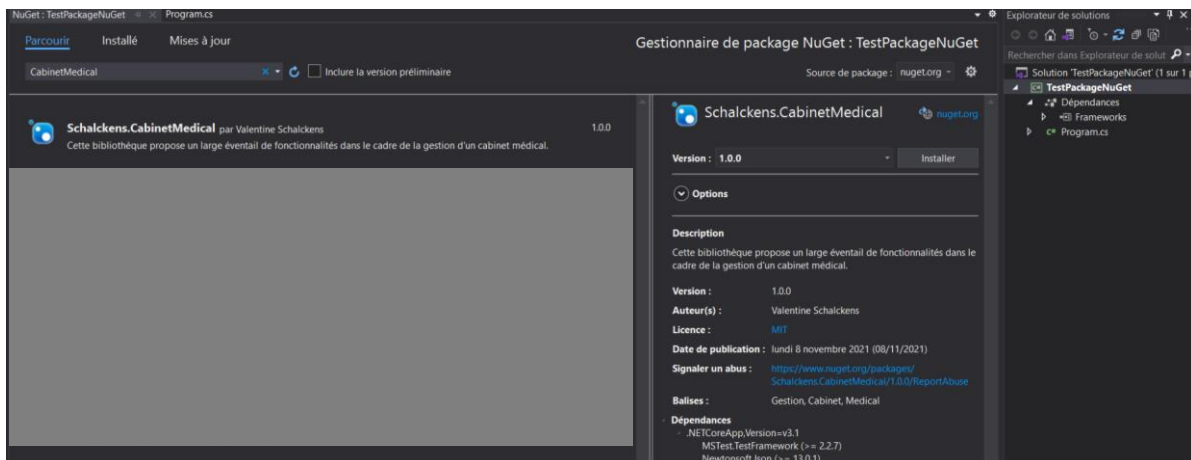
Il ne reste plus qu'à attendre sa validation qui se fait en quelques minutes !

PARTIE I: TEST DU PACKAGE CREER

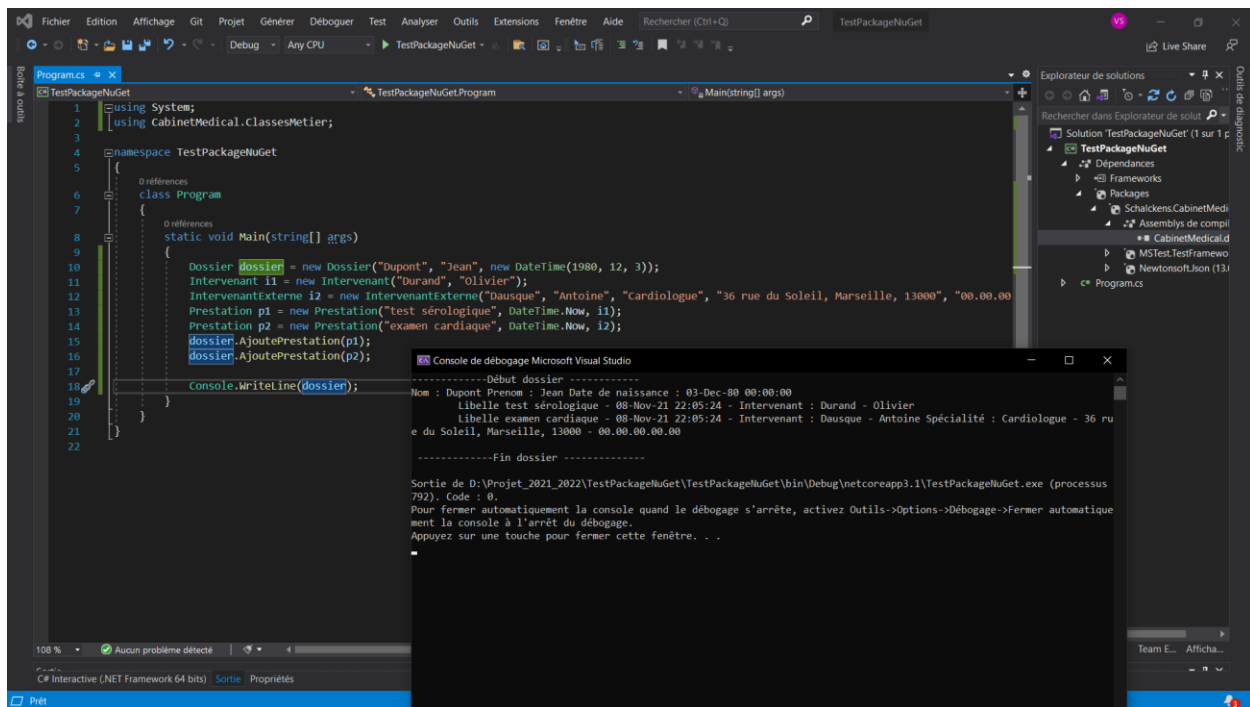
1. Création d'un nouveau projet :



2. Installation du package NuGet « Schalckens.CabinetMedical » : Il faudra accepter les conditions de la licence et si tout s'est bien passé on ne doit pas avoir d'erreur à la fin de l'installation.



3. Test du package : On devrait maintenant avoir accès aux fonctionnalités du package !



PARTIE II: AJOUT D'UNE FONCTIONALITÉ (+TUTORIEL)

1. Ajout de la classe Cotation, modification des autres classes et implémentation de test :

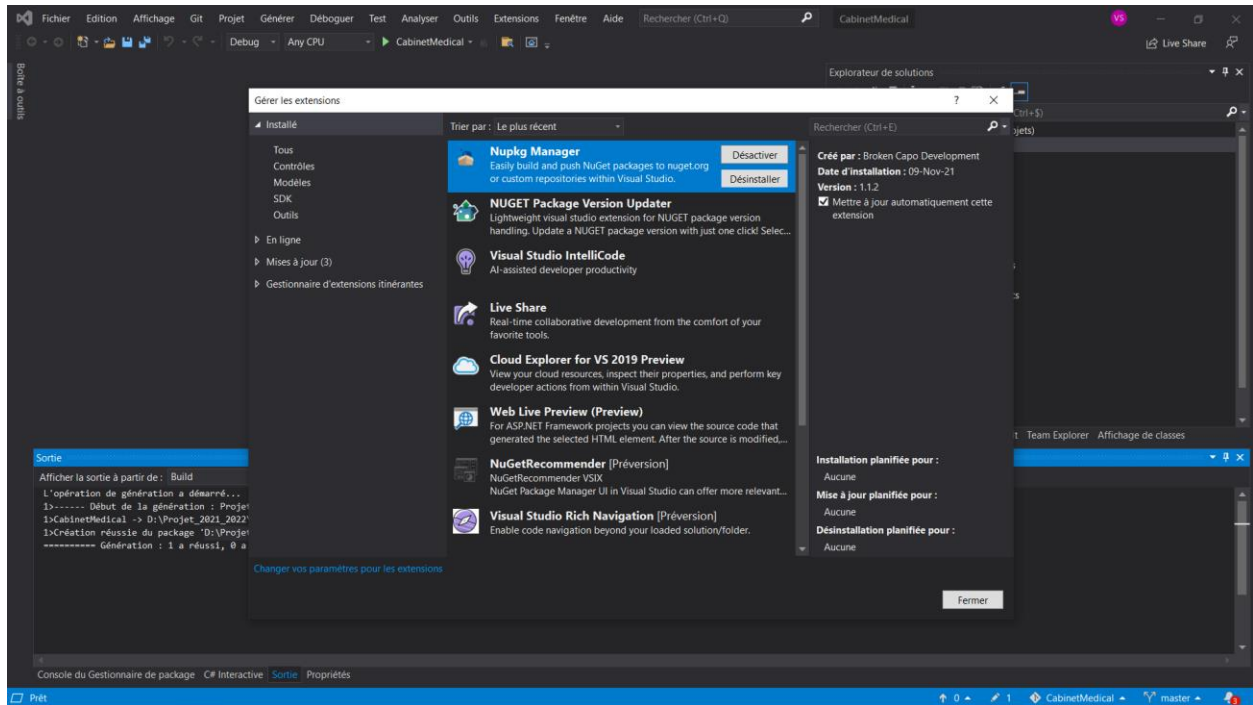
```
4 namespace CabinetMedical.ClassesMetier
5 {
6     using CabinetMedical.Exceptions;
7
8     /// <summary>
9     /// </summary>
10    /// <summary>
11    /// </summary>
12    public class Cotation
13    {
14        private string id;
15        private string libelle;
16        private int indice;
17
18        /// <summary>
19        /// Initializes a new instance of the <see cref="Cotation"/> class.
20        /// Constructeur de la Classe Cotation.
21        /// </summary>
22        /// <param name="id">Id de la cotation. </param>
23        /// <param name="libelle">Libelle de la cotation. </param>
24        /// <param name="indice">Indice de cotation. </param>
25        public Cotation(string id, string libelle, int indice)
26        {
27            this.id = id;
28            this.libelle = libelle;
29            if (indice > 0)
30            {
31                this.indice = indice;
32            }
33            else
34            {
35                throw new CabinetMedicalException("Indice négatif ou égale à 0.");
36            }
37        }
38
39        /// <summary>
40        /// Gets id.
41        /// </summary>
42        public string Id { get => this.id; }
43
44        /// <summary>
45        /// Gets or Sets libelle.
46        /// </summary>
47        public string Libelle { get => this.libelle; set => this.libelle = value; }
48
49        /// <summary>
50        /// Gets or Sets indice.
51        /// </summary>
52        public int Indice { get => this.indice; set => this.indice = value; }
53    }
54 }
55
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Text;
4 using Microsoft.VisualStudio.TestTools.UnitTesting;
5 using CabinetMedical.ClassesMetier;
6 using CabinetMedical.Exceptions;
7
8 namespace TestCabinetMedical
9 {
10    [TestClass]
11    class CotationTest
12    {
13        [TestMethod]
14        public void TestInstanceCotationOK()
15        {
16            Cotation cotation = new Cotation("3e", "cotation 1", 2);
17            Assert.IsInstanceOfType(cotation, typeof(Cotation));
18        }
19
20        [TestMethod]
21        [ExpectedException(typeof(CabinetMedicalException))]
22        public void TestInstanceCotationKO()
23        {
24            Cotation cotation = new Cotation("3g", "cotation 2", -1);
25        }
26    }
27 }
28
29
```

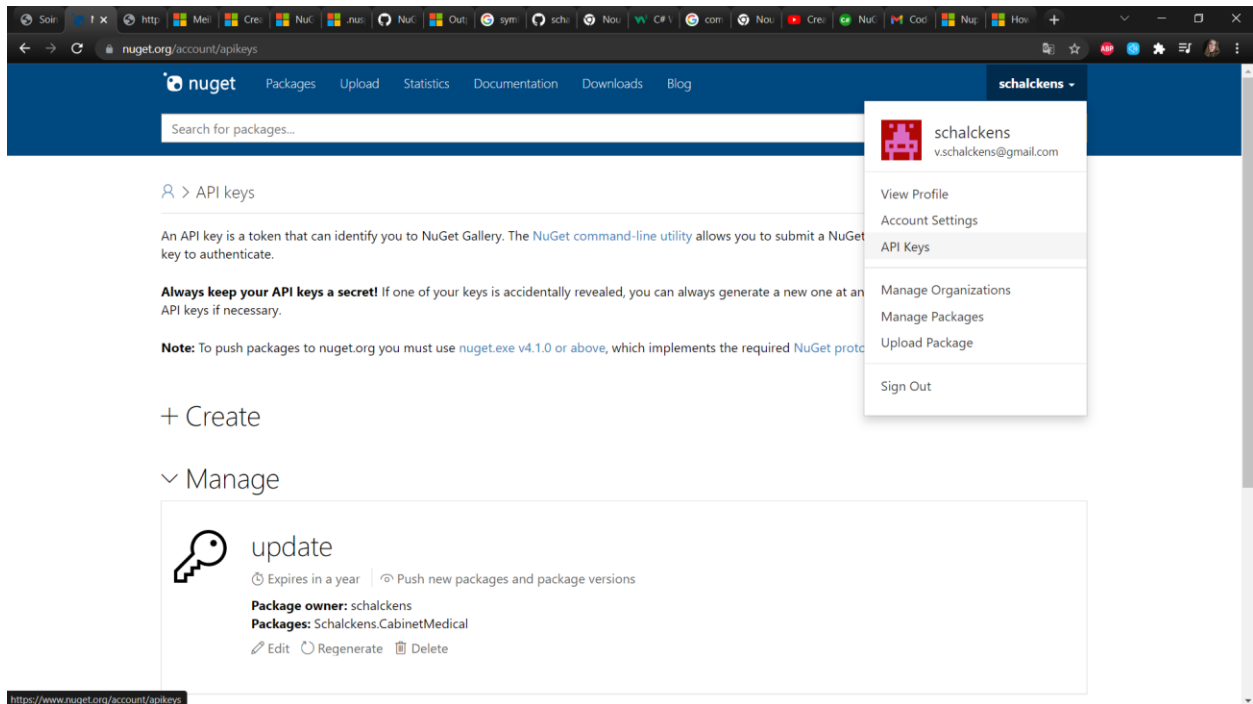
```
13 public class Prestation
14 {
15     private int prixFixe = 100;
16     private int prixCourant;
17     private Cotation cotation;
18
19     // Méthodes
20
21     /// <summary>
22     /// Initializes a new instance of the <see cref="Prestation"/> class.
23     /// </summary>
24     /// <param name="libelle">Libellé. </param>
25     /// <param name="dateHeureSoin">Date et heure de la prestation. </param>
26     /// <param name="intervenant">Objet de la classe Intervenant. </param>
27     /// <param name="cotation">Objet de la classe Cotation. </param>
28     public Prestation(string libelle, DateTime dateHeureSoin, Intervenant intervenant, Cotation cotation)
29     {
30         this.Libelle = libelle;
31         if (DateTime.Compare(DateTime.Now.Date, dateHeureSoin.Date) >= 0)
32         {
33             this.DateHeureSoin = dateHeureSoin;
34         }
35         else
36         {
37             throw new CabinetMedicalException("Date non conforme ");
38         }
39
40         this.Intervenant = intervenant;
41         this.cotation = cotation;
42         this.prixCourant = cotation.Indice * this.prixFixe;
43     }
44
45     /// <summary>
46     /// Gets or Sets prixFixe.
47     /// </summary>
48     public int PrixFixe { get => this.prixFixe; set => this.prixFixe = value; }
49
50     /// <summary>
51     /// Gets or Sets prixCourant.
52     /// </summary>
53     public int PrixCourant { get => this.prixCourant; set => this.prixCourant = value; }
54
55     /// <summary>
56     /// Gets or Sets cotation.
57     /// </summary>
58     public Cotation Cotation { get => this.cotation; set => this.cotation = value; }
59 }
60
```


2. Mise à jour du package NuGet une fois le projet re-compressé et la version changée : **TUTORIEL**

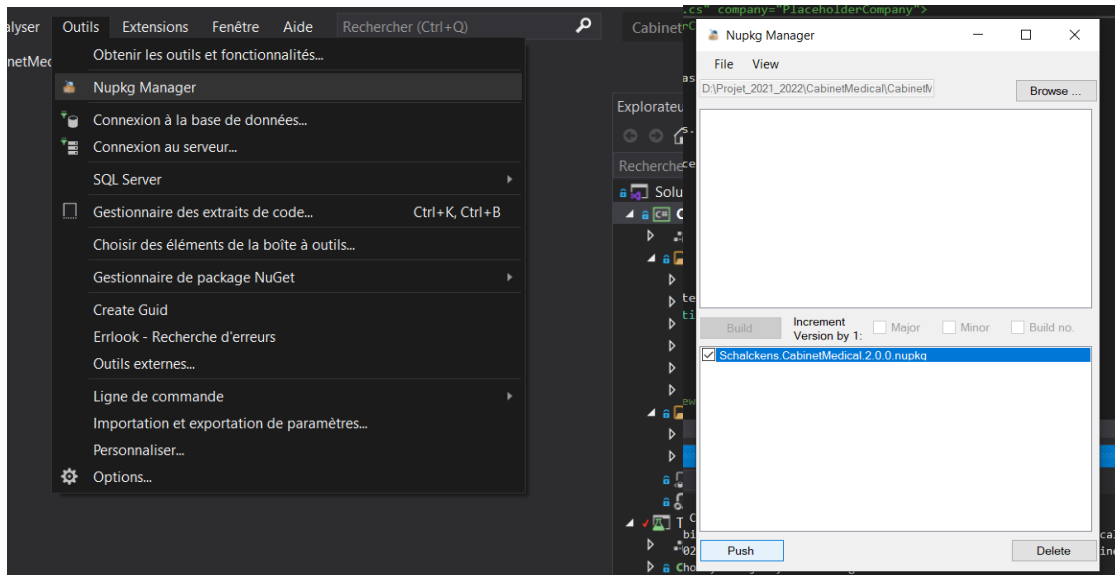
a. Installation de de l'extension Nupkg Manager : Extension → Gérer les extensions...



b. Création d'une API keys sur NuGet.org.

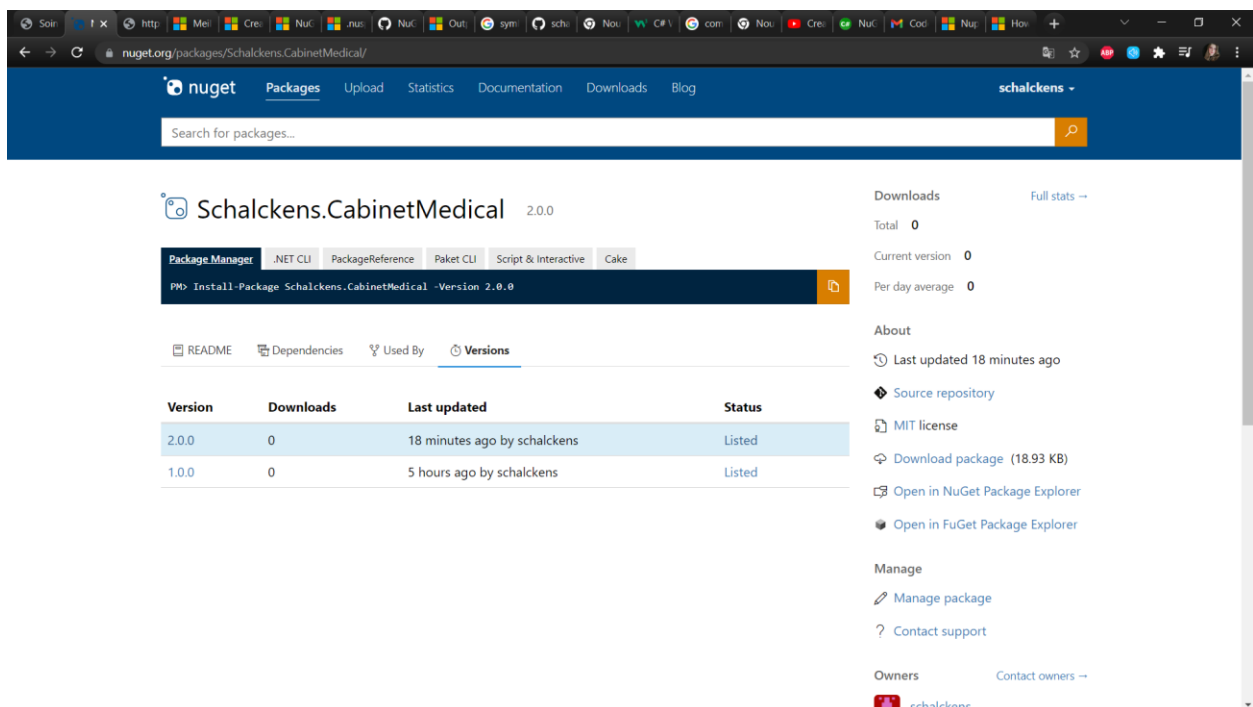


c. Utilisation de Nupkg Manager pour push la nouvelle version sur NuGet.org.



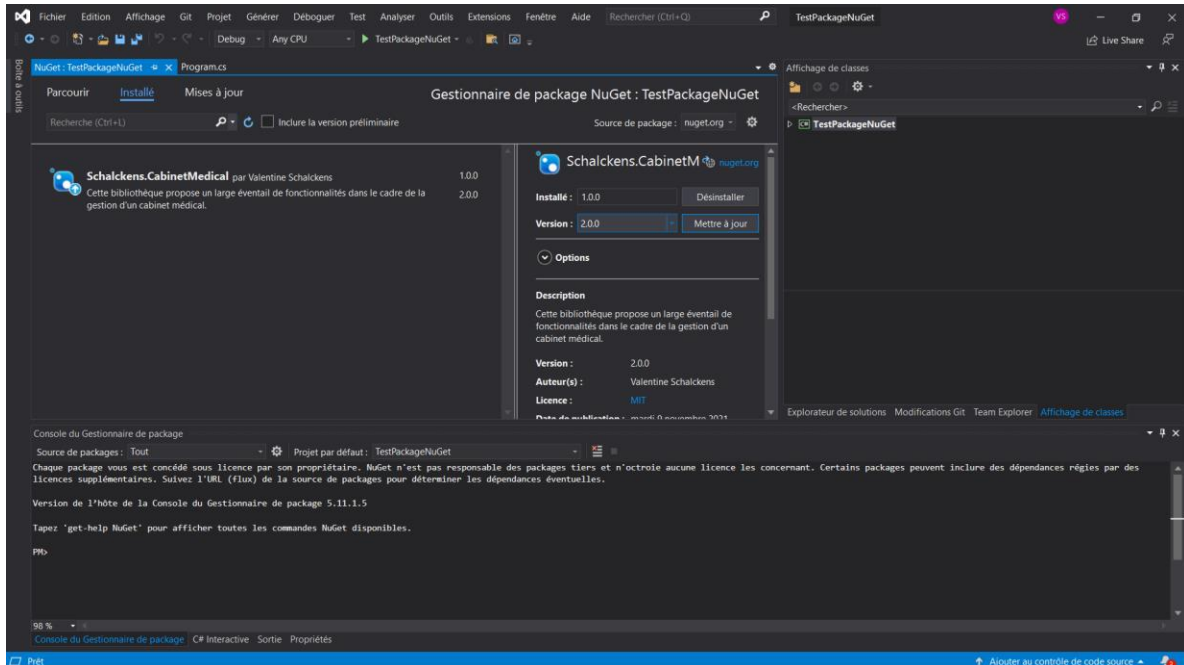
En appuyant sur push la première fois il va vous demander de renseigner l'API keys que vous avez créé au préalable et le serveur NuGet : <https://api.nuget.org/v3/index.json>.

Et voilà ! Après validation votre package NuGet est passé à la version supérieure !



PARTIE III: MISE EN OEUVRE DE LA NOUVELLE FONCTIONNALITÉ

1. Mettre à jour le package Schalckens.CabinetMédical :



2. Et voici son utilisation :

