

Documenter son développement.

Il est très important de documenter ces classes afin d'aider les développeurs qui doivent les implémenter et produire une documentation technique de nos développements.

Comme la plupart des développeurs, vous avez la directive de documenter vos codes. Mais vous ne devez pas échapper à la règle : comme la plupart des développeurs, la documentation n'est pas votre tasse de thé.

Le C# va vous aider à faire une bonne documentation, sans pénaliser votre principale fonction : développer. Nous allons nous appuyer ici sur un des principes essentiels de la plateforme .Net : la capacité des objets à se documenter.

Introduire des commentaires

C# propose quatre catégories de commentaires, que vous pouvez utiliser dans vos développements :

- La ligne simple de commentaire, précédée par //
- Les commentaires délimités, encadrés par /* votre commentaire */
- La ligne simple de documentation, précédée par ///
- Et enfin, la documentation délimitée, qui combine les deux possibilités précédentes ;

L'intérêt de la ligne simple de documentation et de la documentation délimitée est qu'au moment de la compilation du code, il est possible de préciser que l'on souhaite l'extraction de la documentation.

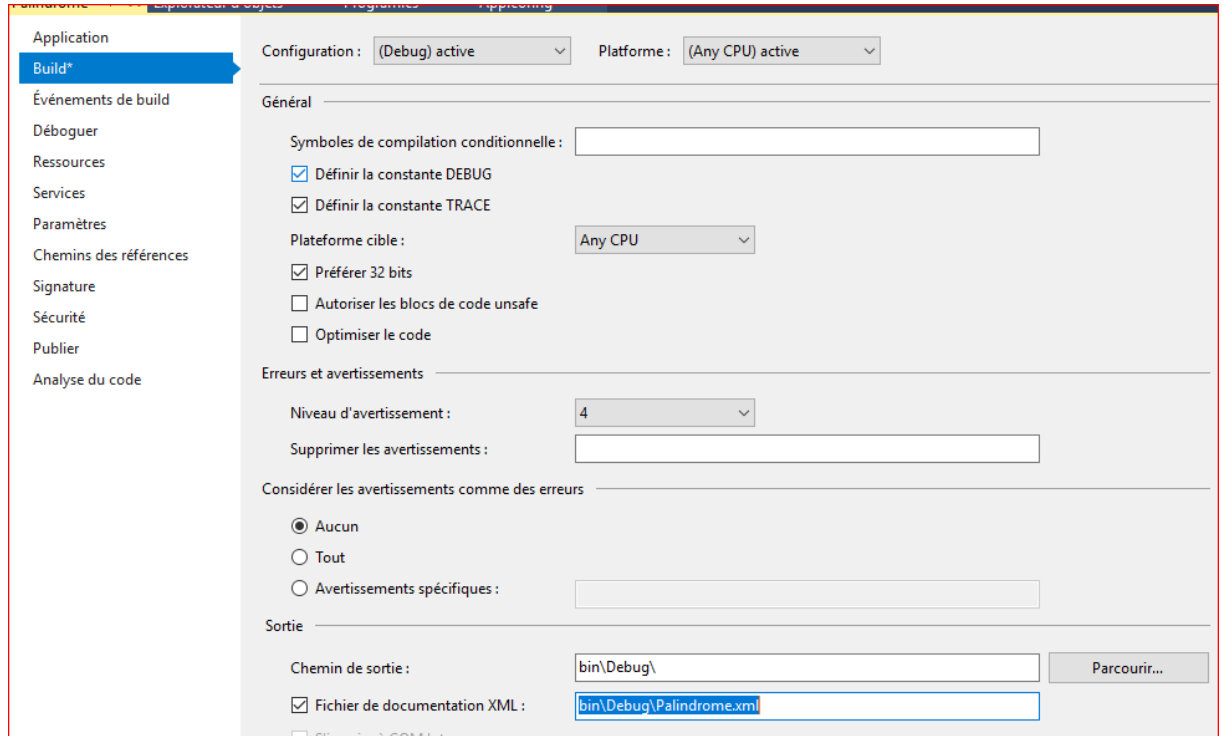
Un fichier XML est alors généré, et il suffit d'y appliquer une feuille de style XSL-T pour obtenir une documentation bien formée de votre code.

```
/// <summary>
/// Nom du salarié
/// Longueur comprise entre 3 et 30 caractères. Ni caractères spéciaux ni chiffres
/// </summary>
public string Nom
{
    get { return (this._nom); }
```

Génération du fichier de documentation

Ouvrez un projet documenté et affichez sa fenêtre propriétés depuis l'option de menu Projet.

1. Choisissez l'onglet Build et cochez la case Fichier de documentation XML. Par défaut le fichier se place dans le répertoire où se trouve l'assembly du projet.



2. Générez l'application.

Vous obtenez des avertissements dans la fenêtre de sortie vous indiquant les omissions de commentaires **des membres publics exposés** ou les commentaires mal rédigés.

	Description
1	Commentaire XML absent pour un type ou membre visible publiquement 'Poo3.Commercial'
2	Commentaire XML absent pour un type ou membre visible publiquement 'Poo3.Salarie.operator ==(Poo3.Salarie, Poo3.Salarie)'
3	Commentaire XML absent pour un type ou membre visible publiquement 'Poo3.Salarie.operator !=(Poo3.Salarie, Poo3.Salarie)'

3. Ajoutez les commentaires qui manquent pour vous familiariser avec ces techniques si nécessaire.

Regénérez l'assembly.

4. Rendez-vous dans le répertoire où votre document XML a été généré.
5. Ouvrez le document XML dans votre navigateur ou dans visual Studio

Documenter son code

Vous pouvez consulter le document XML.

Essayez d'analyser la structure de ce dernier.

Vous retrouvez les éléments et attributs XML qui décrivent les membres de vos types.

Un préfixe indique la nature de l'élément. Sans entrer dans les détails, et à titre d'exemple, nous avons :

- Un T pour le type
- Un M pour une méthode (+suffixe #cor pour les constructeurs)
- Un P pour les propriétés

```
<?xml version="1.0"?>
- <doc>
  - <assembly>
    <name>Poo3</name>
  </assembly>
  - <members>
    - <member name="T:Poo3.Salarie">
      <summary> Type Salarie </summary>
    </member>
    - <member name="M:Poo3.Salarie.IsNomPrenomValide(System.String)">
      <summary> Verification de la conformite de la saisie pour les noms et prenom. Longueur comprise entre
        3 et 30 caractères Uniquement composé de lettres </summary>
      <param name="value"/>
      <returns/>
    </member>
```

Ce document est lisible mais sa mise en forme peut être améliorée.

Nous allons lui associer un document XSLT. XSLT (eXtensible Stylesheet Language Transformations), défini au sein de la recommandation XSL du W3C, est un langage de transformation XML.

Il permet notamment de transformer un document XML dans un autre format, tel PDF ou encore HTML pour être affiché comme une page web.

Téléchargez le document XSLT mis à votre disposition sur le site. Stockez ce dernier dans le même répertoire que votre document XML (pour simplification de l'expression de la référence hypertexte).

- Ouvrez votre document XML dans Notepad++ ou Visual Studio et ajoutez la ligne de référencement de la feuille de style de transformation juste en dessous de la balise de définition :

<?xml-stylesheet type="text/xsl" href="documentation.xsl"?>

Résultat :

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="documentation.xsl"?>
<doc>
```

Enregistrez le document et ouvrez-le dans votre navigateur à nouveau. Le document est mis en forme. Vous pouvez bien entendu modifier la mise en forme codée dans le document XSLT.

Extrait :

Assemblage : Palindrome	
Classe : Program	
Ce programme permet de déterminer si une chaîne entrée au clavier est un palindrome.	
Méthodes :	
DialoguePalindrome	
Acquisition d'une chaîne pour traitement palindrome	
EstPalindrome(System.String)	
Détermine si la chaîne est un palindrome	
Valeur de retour :	
<i>Paramètres :</i>	
messageRetraite:	
EstPalindrome2(System.String)	
Déterminer palindrome Autre version plus concise mais moins claire	
Valeur de retour :	
<i>Paramètres :</i>	
messageRetraite:	
EstPalindrome3(System.String)	