



TP .Net

Bug tracking like

Lemettre Arnaud

Version 1.0

9 Pages

02/04/2009

EPITA-MTI2014-NET-TP-bug-track



Propriétés du document

Auteur	Lemettre Arnaud
Version	1.0
Nombre de pages	9
Références	EPITA-MTI2014-NET-TP-bug-track

Historique du document

Date de vision	Version	Auteur	Changements
27/02/2009	0.1	Lemettre	création

Site de référence

description	url
Site MTI	
Blog MTI	

Sommaire

Introduction	4
Contexte.....	5
Partie 3.....	6
Renseignements	6
Travail à faire	6
Modalité de rendu	9

Introduction

Le but de cette série de TP sera la réalisation d'un système de bug tracking. Ce tp se décompose en plusieurs parties. Toutes les semaines vous aurez une partie à rendre. Cette partie sera évaluée. Toute fois si pour une raison quelconque vous n'avez pas réussi à faire convenablement cette partie une correction sera postée une journée avant le début du prochain TP. Vous aurez à charge de l'intégrer dans votre code si celui ne produit pas le fonctionnement attendu.

Ce travail est à faire individuellement, tout code similaire sur deux personnes sera considéré comme un travail non rendu et non négociable.

Durant cette série de TP les notions suivantes seront abordées :

- modélisation BDD
- création de script SQL (Base, tables, données)
- création procédures stockées
- Entities Framework
- Linq To SQL
- Manipulation Open XML
- Génération de Flux XML
- Dynamic Data Entities
- Membership provider / Role provider
- utilisation des pages aspx et composants
- WCF (Webservice et sécurisation)

Chaque partie à rendre ne demande pas plus de 3h de travail par semaine, Bonne chance ;)

Contexte

Un système de suivi de problèmes (de l'anglais bug tracking system ou système de suivi de bogues) est une application informatique qui permet d'aider développeurs et utilisateurs à améliorer la qualité d'un logiciel. Les utilisateurs soumettent les bogues (défauts de fonctionnement) rencontrés dans le logiciel. Les développeurs sont alors toujours au fait des problèmes rencontrés.

L'ensemble de la solution devra utiliser SQL Server 2012 pour la base de données, et le Framework 4.5 de .Net.

Le rendu de chaque lot, correspond au rendu de chaque partie.

Partie 3

Un tracker de bug c'est très utile pour la résolution de bugs, de plus cela facilite le travail collaboratif. Cependant en réunion nous n'avons toujours pas accès à l'internet. Donc il nous faut un système d'export de données.

Renseignements

Lorsque nous sommes sur un bug nous voudrions pouvoir générer un rapport comprenant le titre du bug, la personne assignée, et son statut. Ainsi que la liste des commentaires, pour un bug.

Le résultat au format docx doit ressembler à cela :

Rapport de bug

Titre : XXXXX

Personne responsable : XXXXX

Statut : XXXXX

commentaires :

XXXX

XXXX

XXXX

Travail à faire

Vous devez implémenter dans la bonne couche la gestion du rapport.

Dans un premier temps il vous faut une nouvelle classe à mettre dans la couche DBO :

```
namespace lemett_aBugTrack.DBO
{
    public class BugReport
    {
        #region variables
        private string _title;
        private string _type;
        private string _responsable;
        private string _statut;
        private List<string> _comments;
        #endregion

        [getter / setter]

        public BugReport()...
```

Figure 1 - Description de la classe

Ensuite dans la couche de dataAccess, vous devez créer une méthode :

```
public static string GenerateReportBug(DBO.BugReport bugReport)
```

Cette méthode permettra de générer le fichier docx dans le répertoire « download » la chaine de caractère en retour permettra de connaitre le nom du fichier s'il y a des erreurs la chaine sera vide. Servez vous de la classe Guid pour générer un nom de fichier unique.

Afin de rendre cette fonctionnalité disponible depuis l'interface vous devez créer la classe dans la couche BusinessManagement. Cette classe devra se nommer : GenerateDocx.cs

La méthode que vous devez implémenter sera la suivante :

```
public static bool GenerateReportBug(long idBug)
```

Elle prend en paramètre l'id d'un bug dont vous vous servirez pour aller chercher en base de données les commentaires associés à ce bug.

L'architecture finale devrait ressembler à cela :

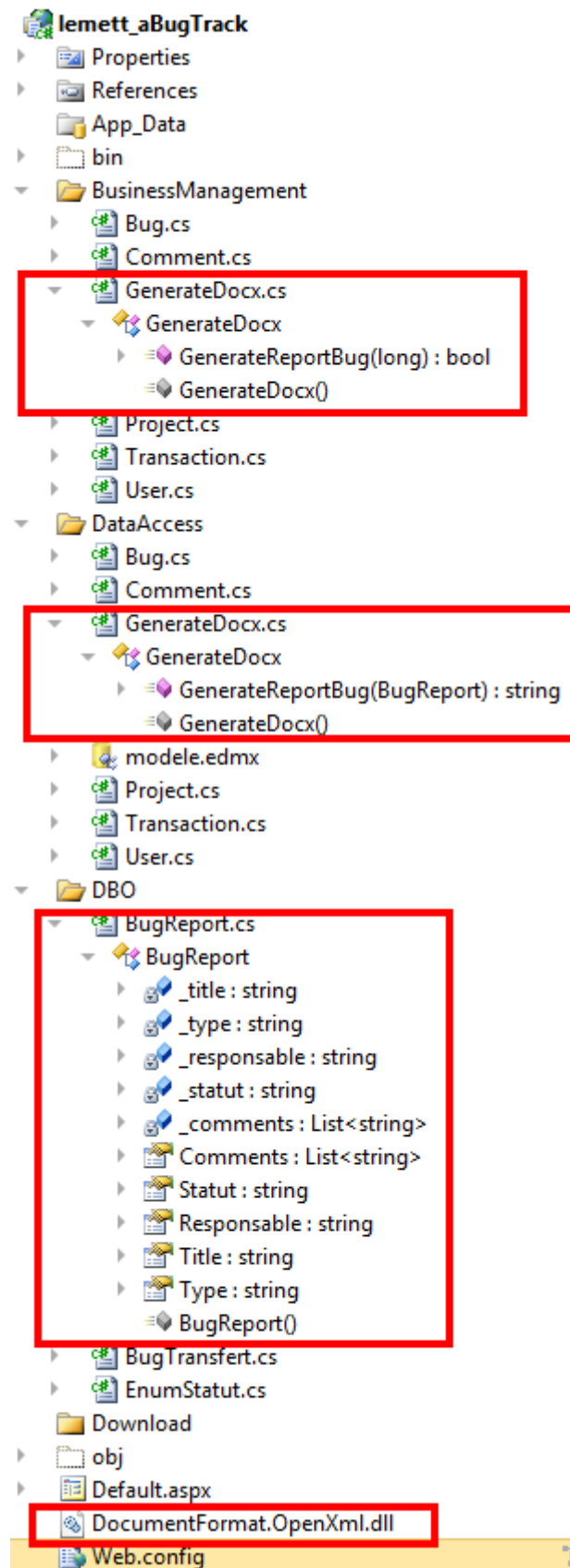


Figure 2 - Architecture de la solution

Bonus :

Faire un fichier qui ressemble à ça en utilisant les styles :

Rapport de bug

Titre : XXXXX

Personne responsable : XXXXX

Statut : XXXXX

commentaires :

XXXX

XXXX

XXXX

Modalité de rendu

Les fichiers seront à rendre dans une tarball ayant pour nom :

login_l.zip

Cette tarball devra comprendre :

- un script de création de votre base ainsi que des tables.
Nom : CreateBugTrack.sql
- Un script pour remplir votre base avec des données
Nom : DataBugTrack.sql
- Un dossier contenant la solution Visual Studio qui devra compiler.
Nom : login_lBugTrack

Le tout à envoyer sur l'adresse mti.rendu.dotnet@gmail.com avec les balises suivantes :

[MTI2014][NET][login_l][BugTrack] partie 3