Documentation PersistentDataObject

La classe PersistentDataObject enregistre toutes les données dans un dictionnaire, de la classe .NET Dictionary<string,dynamic>.

Les propriétés et les méthodes de cette classe permet d’ajouter ou de retirer des données du dictionnaire.

Il n’y a qu’un seul et unique champ : le dictionnaire de données.

Il y a un unique constructeur pour créer un objet. Deux propriétés permettent d’obtenir les éléments un par un :

* Data
* Keys

Data donne l’intégralité des données et Keys la liste des noms dans le dictionnaire.

Ensuite viennent les méthodes. Certaines fonctions sont virtuelles pour les classes dérivées MarshallingValue, MarshallingList et MarshallingHash.

Deux fonctions sont très utilisées :

* GetProperties()
* GetProperty()

Ces fonctions sont inutiles dans MarshallingValue mais elles le sont pour MarshallingHash et MarshallingList.

La fonction GetProperty() retourne le type de donnée d’interface IMarshalling car GetProperties() retourne uniquement des données de type IMarshalling.

La fonction GetProperty() prend en paramètre un nom du dictionnaire qui est censé être de type IMarshalling.

Les données de type IMarshalling sont créées à partir des objets de classe MarshallingHash et MarshallingList.

* Exists()
* Get()
* Set()
* Remove()

Les fonctions ci-dessus permettent de prendre en compte des données de type de données hors de IMarshalling.

En particulier

/// <summary>

/// Set a value if exists

/// </summary>

/// <param name="name">value name</param>

/// <param name="a">function</param>

/// <returns>true if succeeded</returns>

public bool Get(string name, Action<string, IMarshalling> a)

{

if (this.Exists(name))

{

a(name, this.Get(name));

return true;

}

else

return false;

}

Cette fonction permet de tester l’existence d’une donnée dans l’objet et de modifier une variable externe sauf si la donnée n’existe pas.

Utilisation de la classe PersistentDataObject

1. Récupérer une donnée   
   La donnée est un couple nom/valeur sans définir le type de donnée de cette valeur  
   La fonction Get() récupère une donnée s’il on a son nom
2. Modifier une donnée  
   La fonction Set()est mutualisée avec l’ajout ou la modification
3. Supprimer une donnée  
   La fonction Remove(name) permet de supprimer une donnée par son nom
4. Itération de toutes les données  
   Une boucle sur les clés permet de lire toutes les données

Transfert de données dans un autre objet

Deux formes de transfert de données dans un autre objet

1. L’objet est uniquement recopié (pas intégralement) des données du dictionnaire
2. L’objet est copié mais chaque donnée est renommée

Easy Web For Developers

# La génération du site web

Générer le site web correspond à générer

* Une page
* Dans cette page, il y a une Master Page
* La master page contient les entêtes du fichier HTML (les premières balises)
* La page contient des objets qui sont dynamiquement insérés dans le fichier HTML
* Selon les objets qui sont enregistrés dans la page, ce sont des instances de master objet ou d’outil web
* Un master objet est un tableau au sens HTML. Il est réparti par ligne et par colonne pour chaque ligne : Horizontally est la propriété contenant une liste d’éléments contenant une sous-liste d’éléments Vertically. Enfin, Vertically contient une liste d’éléments avec une propriété Container.
* Le container est l’élément qui peut contenir un autre objet de type master objet ou un outil.
* Un outil est une séquence codée d’un élément HTML pouvant contenir des propriétés CSS et du code JavaScript ainsi que le texte HTML à insérer.
* L’outil et le master objet est copié dans l’objet de classe HTMLObject et cet objet a la propriété HookContainer qui permet de lier un container nommé avec un HTMLObject.

Un container ne contient qu’un seul contenu mais un HookContainer peut être reproduit plusieurs fois dans une page, notamment pour présenter des listes.

La génération de la page est un formatage des données contenues dans les objets. Ces données appartiennent à la classe PersistentDataObject. Les données proviennent de UXFramework qui permet de composer un système de rendu web à partir d’une définition précise donnée par les objets de UXFramework.

## UXFramework

UXFramework est une bibliothèque de classes permettant de stocker sous la forme d’objet un ensemble de propriétés sous la forme d’un arbre de données ; la propriété Children de UXControl contient la séquence des objets de type UX.

Une fois le UXFramework provisionné avec ses données, il s’agit alors d’un transfert direct ou indirect des données du UXFramework vers la génération du site web.

## Composition du site web de création

**Easy WEB For Developers** est un site web de création de site web. Le site web généré est composé de pages HTML pouvant contenir du code côté serveur (PHP, Java, ASP.NET).

Le site web permet de créer les pages, les master pages, les master objet et les outils qui composent votre site web.

Une fois ces informations précisées, votre site web est généré directement sous la forme de pages HTML rigoureusement recopiées et formattées.

## UXFramework : framework générique

Le UXFramework est une bibliothèque générique de composition d’objets d’interface UI. Elle est générique parce qu’elle ne dépend d’aucune spécialisation.

C’est le moteur de rendu des interfaces UI qui construit minutieusement le rendu. En particulier, Easy WEB For Developers est un moteur de rendu en HTML. Et, pour cela, il existe une classe WebRenderer.cs qui implémente les interfaces UI du framework en page web.