



La Documentation

Julien CONDOMINES | StJoSup | 25 avril 2022

Table des matières

I-Introduction.....	2
II-Doxygen présentation	2
III-Mise en place.....	4
IV-Programmes utilisés.....	9
V-Résultats	10
VI-Conclusion.....	12
VII-Annexes	12
Liste de Classe	12
<i>Classe Collection</i>	12
<i>Classe Mission</i>	13
<i>Classe PdtAch</i>	16
<i>Classe PdtFab</i>	18
<i>Classe Produit</i>	19

I-Introduction

Nous allons ici découvrir les différentes manières de documenter un programme que l'on réalise, il sera ici question de produire un compte rendu de nos fichiers sous format de page HTML à l'aide d'un programme appelé Doxygen.

Cela va permettre de pouvoir présenter l'entièreté de notre projet sous un format facilement appréhendable pour les personnes les moins expertes en programmation mais aussi pour rendre cela plus claire.

II-Doxygen présentation

Doxygen est un logiciel informatique libre permettant de créer de la documentation à partir du code source d'un programme. Pour cela, il tient compte de la grammaire du langage dans lequel est écrit le code source, ainsi que des commentaires s'ils sont écrits dans un format particulier.

En permettant l'intégration de la documentation directement dans le code sources, *Doxygen* permet de favoriser la cohérence entre la documentation et le code et de systématiser le comportement des développeurs afin qu'ils documentent le code qu'ils produisent.

Il est également possible d'extraire de la documentation à partir d'un code source non documenté au préalable, ce qui peut faciliter la compréhension d'un programme dont le code est compliqué.

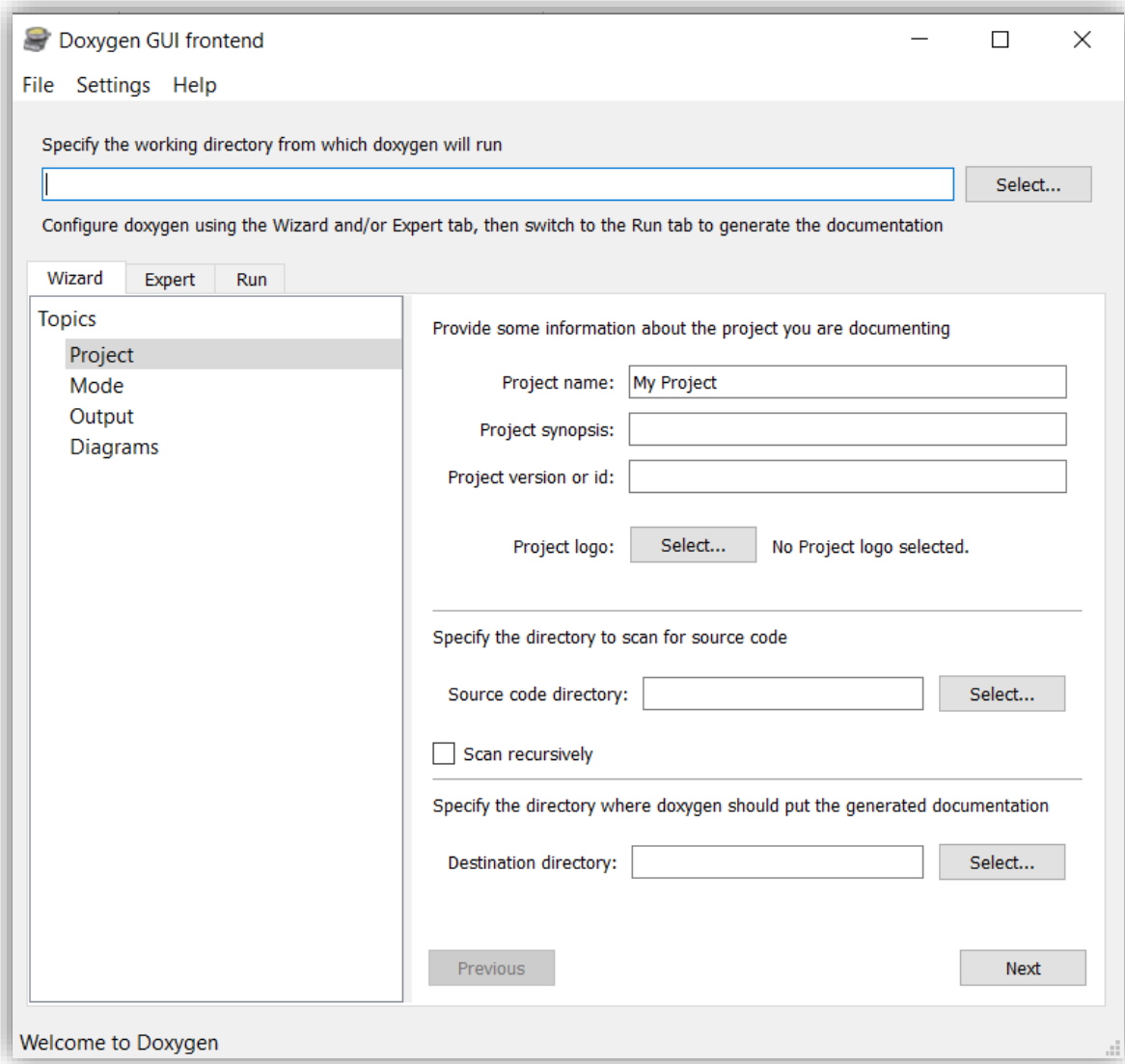
De nombreux projets, tels que KDE, utilisent Doxygen pour créer la documentation de leur API. KDevelop intègre le support de Doxygen. De nombreux éditeurs de texte proposent des modes ou des scripts pour faciliter l'écriture des commentaires Doxygen et la génération de la documentation.

Les informations suivantes peuvent être extraites des sources :

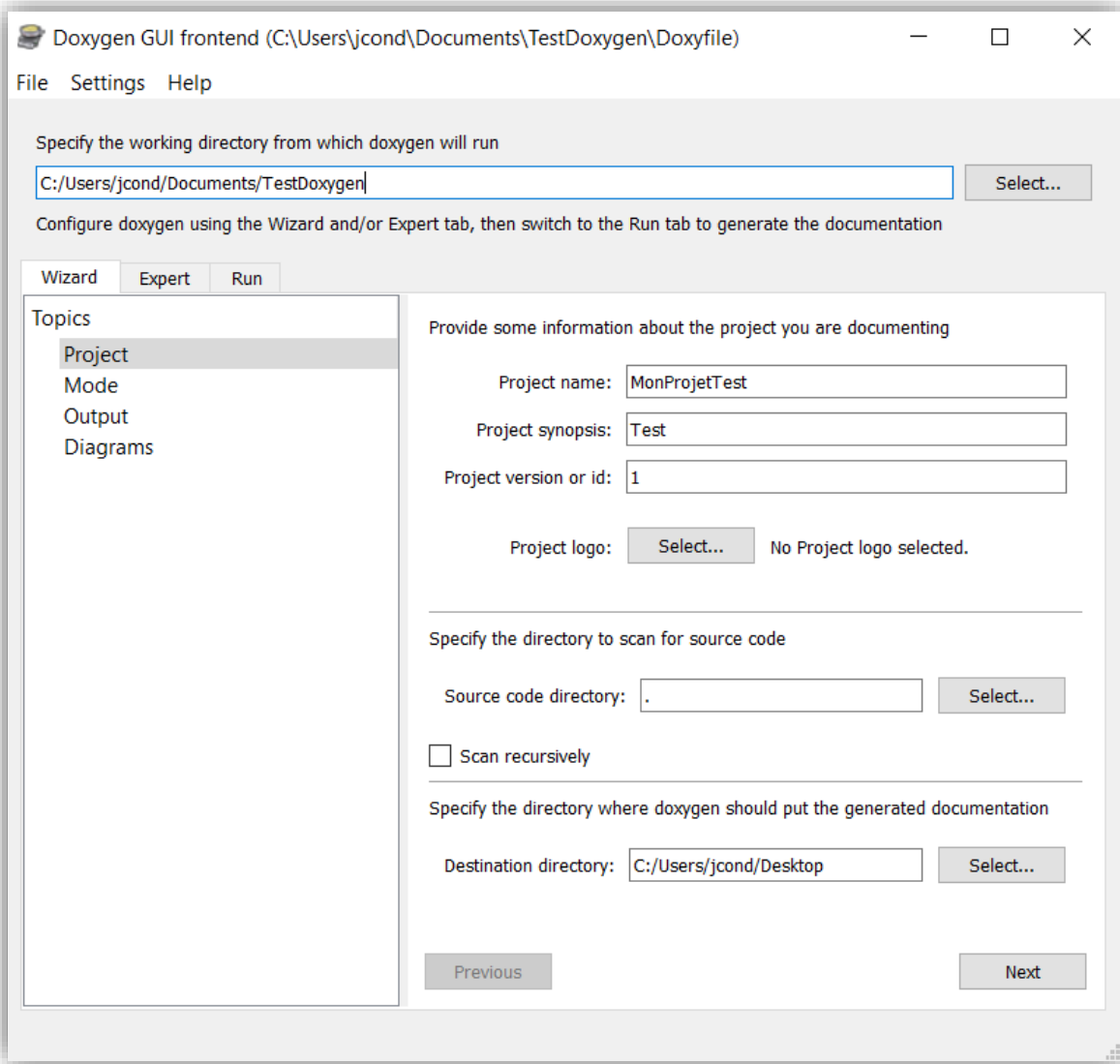
- Prototype et documentation des fonctions, qu'elles soient locales, privées ou publiques, etc.
- Liste des fichiers inclus
- Documentation des structures de données
- Prototype et documentation des classes et leur hiérarchie
- Différents types de graphes : diagrammes de classe, de collaboration, d'appels, d'inclusion, etc. Pour générer certains de ces diagrammes, l'outil gratuit *Dot* est nécessaire.
- Un index de tous les identifiants
- Des fichiers sources annotés (par exemple avec les numéro de lignes) et navigables (par exemple avec HTML, avec lequel les identifiants renvoient vers la documentation associée)

III-Mise en place

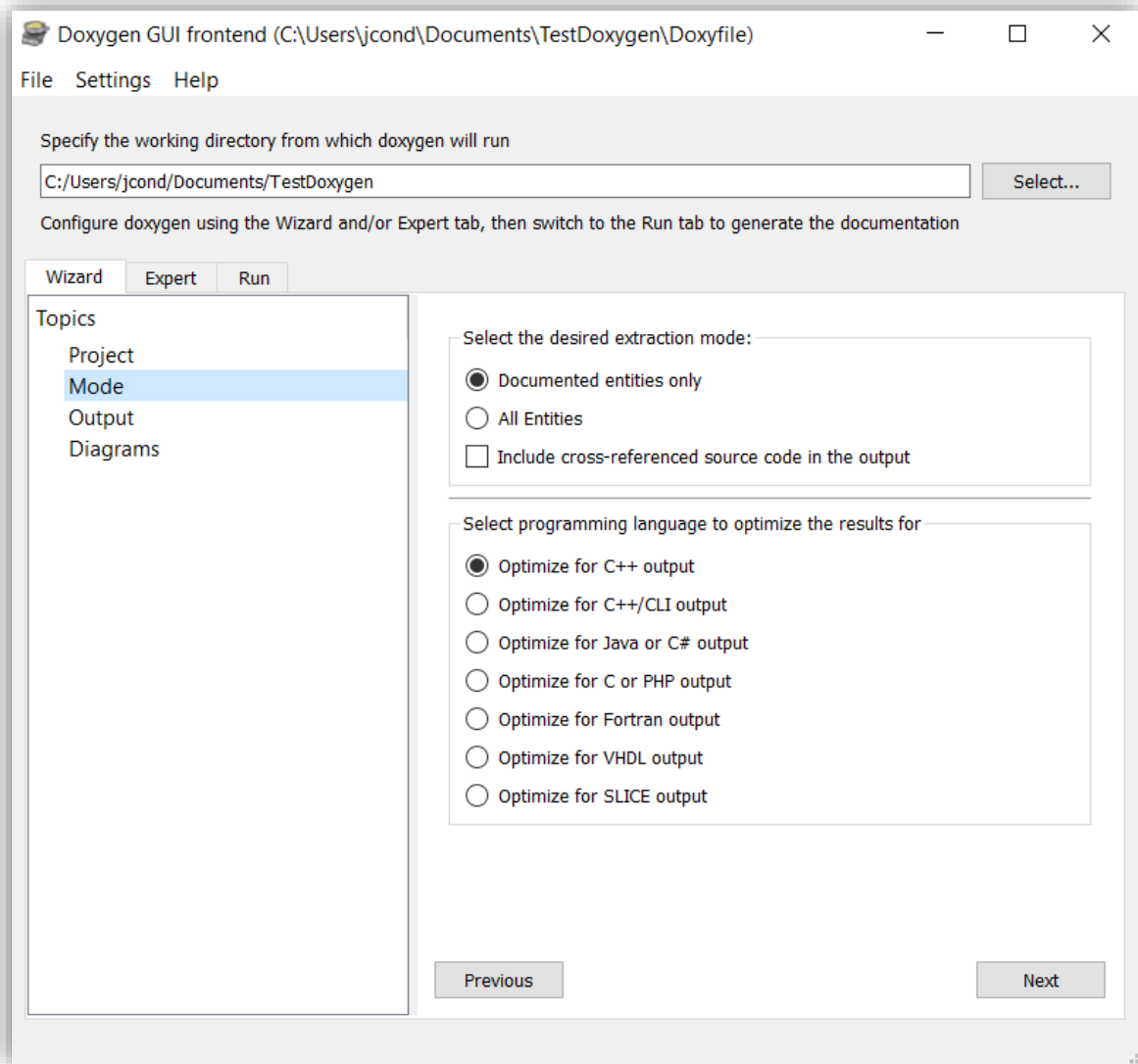
Après l'installation, nous nous retrouvons avec l'écran de base de Doxygen, il va donc être question de fournir les différentes données nécessaire au bon fonctionnement ainsi que de définir l'endroit dans lequel on souhaite créer la documentation.



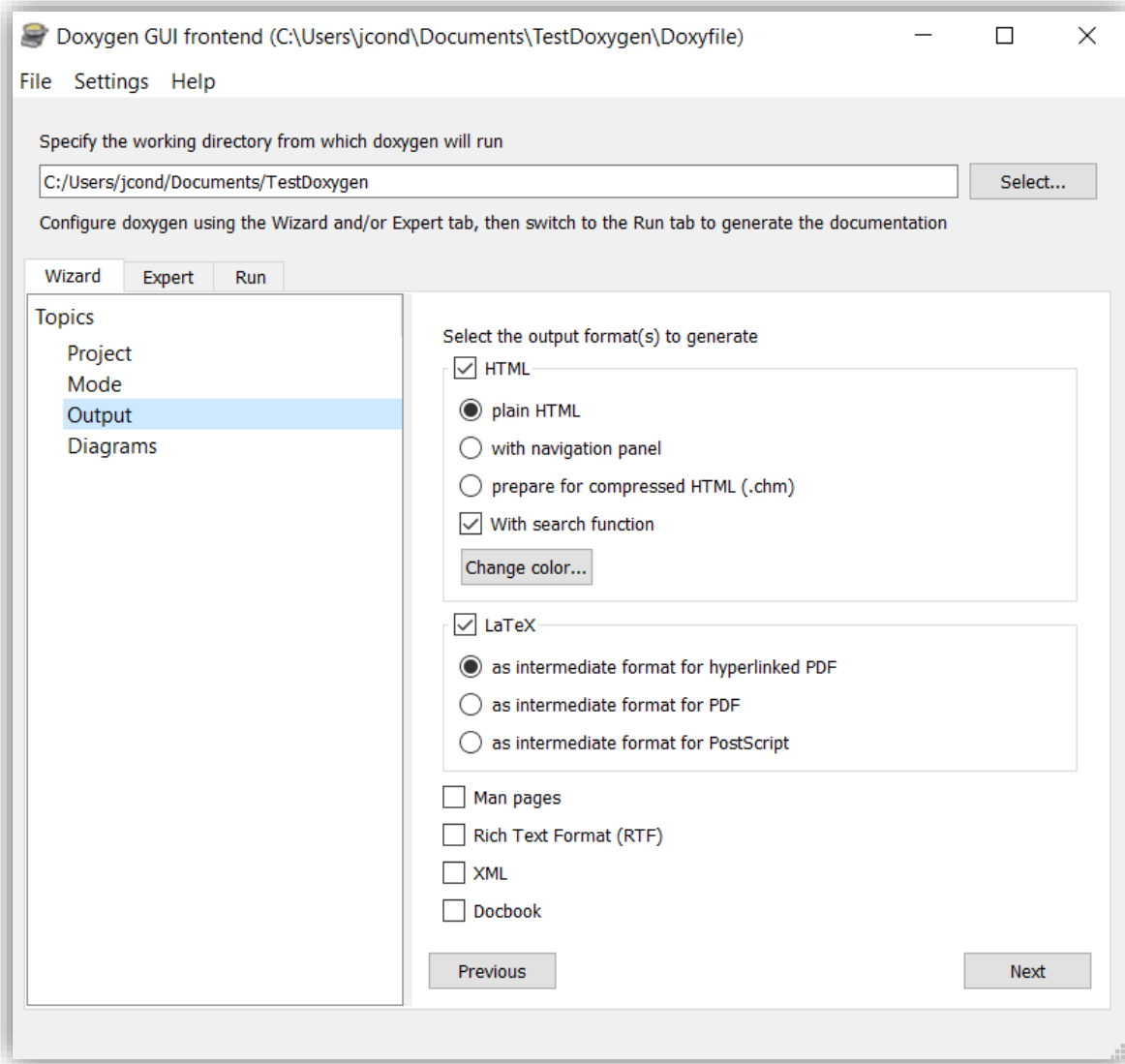
J'ai donc ici choisis le fichier que je souhaitais documenter ainsi que l'endroit dans lequel je souhaite faire apparaitre la documentation ainsi que le nom de mon projet et sa version.



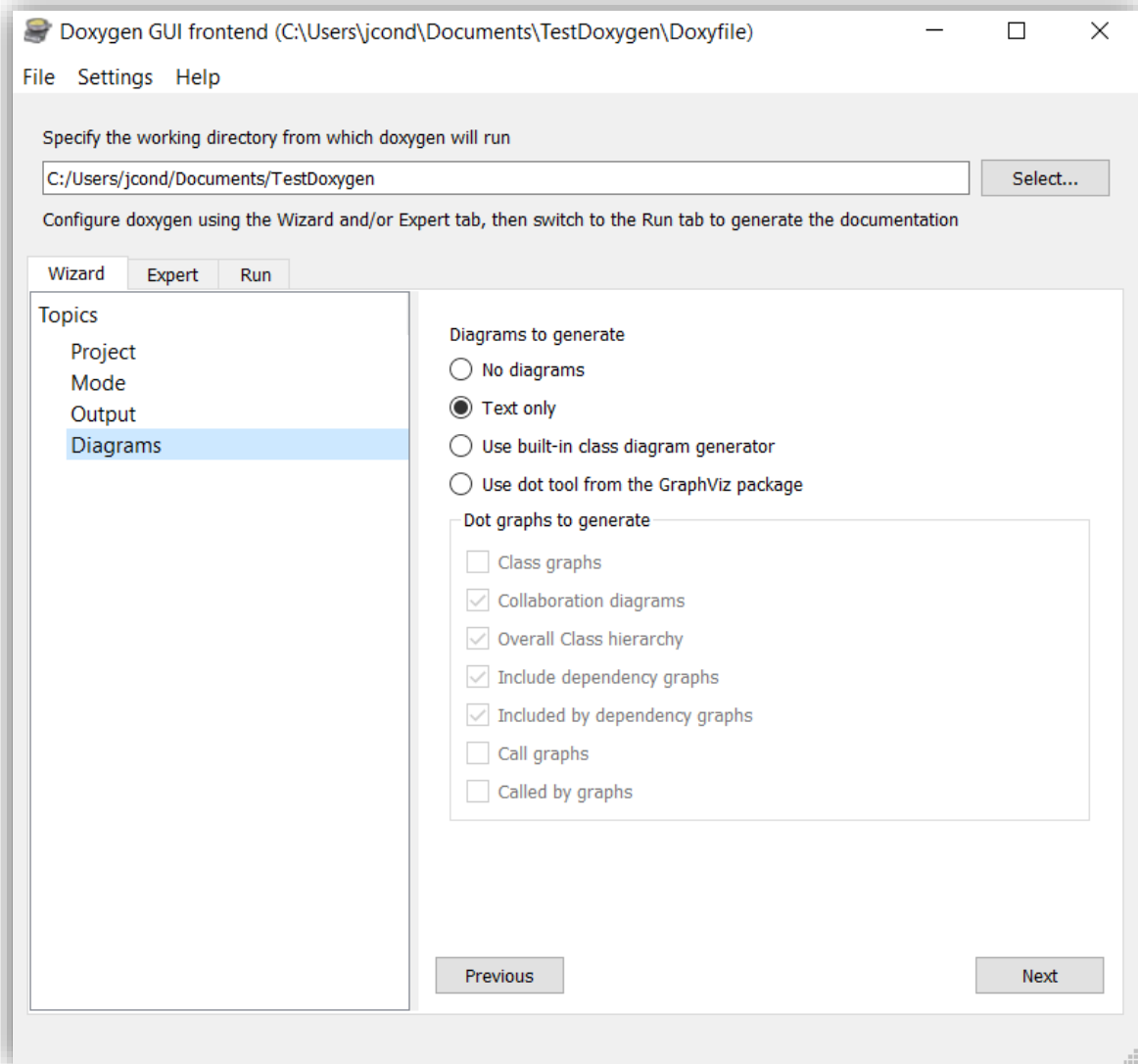
Il faut par la suite décider de ce que l'on souhaite extraire de nos fichiers dans notre documentation ainsi que définir le langage utilisé afin d'optimiser le processus.



Il faut ensuite définir la forme qu'aura notre documentation à la sortie, si l'on souhaite que celle-ci prenne la forme d'une page HTML ou d'un fichier LaTeX ou encore Un RTF ou un fichier XML.



Pour finir, on va définir si l'on souhaite voir apparaitre des diagrammes dans notre documentation.



IV-Programmes utilisés

Il va être ici question d'utilisés différents programmes, afin de visualiser au mieux l'étendue des capacités de Doxygen. On va tout d'abord prendre une classe contenant une collection, celle-ci sera appelé Collection, nous allons ensuite prendre des classes traduisant un héritage appelé Produit, et ses classes hérités PdtFab et PdtAch et enfin une classe Mission contenant un dictionnaire.

Afin d'être reconnu par le programme, il va falloir ajouter des commentaires spéciaux aux services de documentation à l'intérieur de nos classe afin de différencier les différentes composantes de celle-ci.

```

/**
 * @class      Mission
 * @brief      La déclaration de la classe Mission de l'application
Synapsinfo
 * @author     Julien CONDOMINES <julien.condomines@stjosup.com>
 * @version    0.1
 * @date       2022
 * @note       Une note à l'attention de ceux qui lisent les notes
 * @pre        Initialisez d'abord le système
 * @post       L'objet est initialisé ou pas
 * @bug        La copie est impossible ou illégale
 * @warning    Une mauvaise utilisation peut faire planter votre
application (c'est votre faute)
 * @attention  Il faut toujours faire attention
 * @remark     Une remarque à faire ?
 * @copyright  GNU Public License.
 */





















```

Comme on peut le voir ici, il y a différentes marques indiquant chacune une information différentes sur le résultat produit.

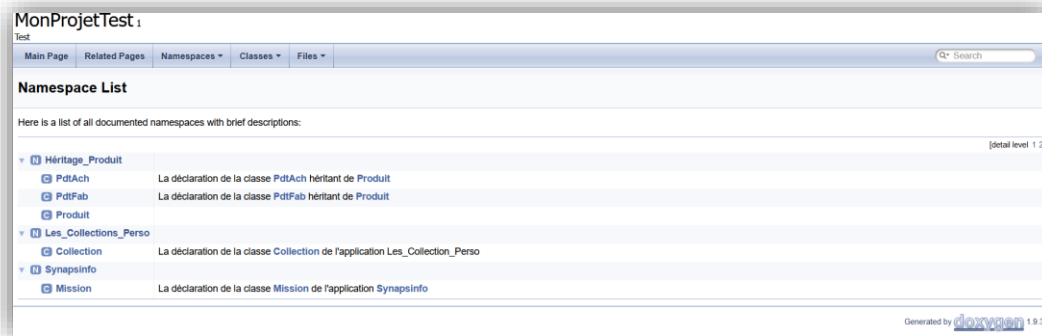
V-Résultats

On va maintenant analyser le résultat obtenu suite au passage des programmes sous Doxygen.

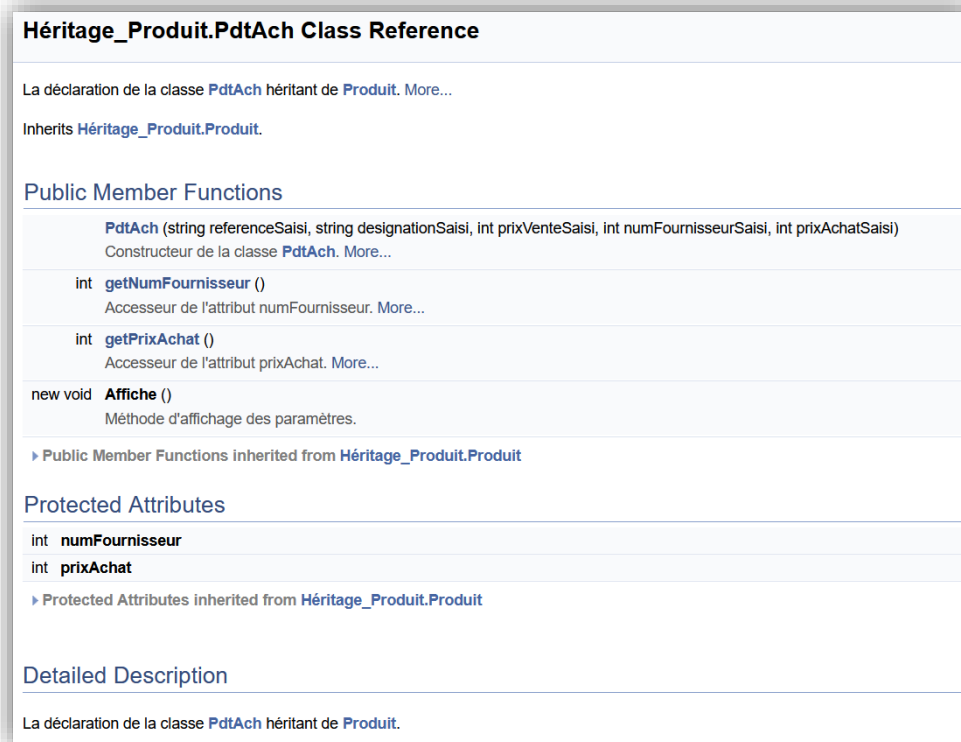
On obtient tout d'abord un fichier regroupant l'ensemble des données selon le format sélectionner précédemment, il est ici question de HTML.

	search	22/03/2022 11:28	Dossier de fichiers	
	_collection_8cs	22/03/2022 11:39	Firefox HTML Doc...	5 Ko
	_mission_8cs	22/03/2022 11:39	Firefox HTML Doc...	5 Ko
	_pdt_ach_8cs	22/03/2022 11:39	Firefox HTML Doc...	5 Ko
	_pdt_fab_8cs	22/03/2022 11:39	Firefox HTML Doc...	5 Ko
	_projet_8cs	22/03/2022 10:50	Firefox HTML Doc...	5 Ko
	annotated	22/03/2022 11:39	Firefox HTML Doc...	7 Ko
	bc_s	22/03/2022 11:39	Fichier PNG	1 Ko
	bdwn	22/03/2022 11:39	Fichier PNG	1 Ko
	bug	22/03/2022 11:39	Firefox HTML Doc...	4 Ko
	class_h_xC3_xA9ritage__produit_1_1_pdt...	22/03/2022 11:39	Firefox HTML Doc...	19 Ko
	class_h_xC3_xA9ritage__produit_1_1_pdt...	22/03/2022 11:39	Firefox HTML Doc...	9 Ko
	class_h_xC3_xA9ritage__produit_1_1_pdt...	22/03/2022 11:39	Firefox HTML Doc...	22 Ko
	class_h_xC3_xA9ritage__produit_1_1_pdt...	22/03/2022 11:39	Firefox HTML Doc...	10 Ko
	class_h_xC3_xA9ritage__produit_1_1_pro...	22/03/2022 11:39	Firefox HTML Doc...	8 Ko
	class_h_xC3_xA9ritage__produit_1_1_pro...	22/03/2022 11:39	Firefox HTML Doc...	7 Ko
	class_h_xC3_xA9ritage__produit_1_1_pro...	22/03/2022 10:50	Firefox HTML Doc...	4 Ko
	class_h_xC3_xA9ritage__produit_1_1_pro...	22/03/2022 10:50	Firefox HTML Doc...	4 Ko
	class_les__collections__perso_1_1_collec...	22/03/2022 11:39	Firefox HTML Doc...	12 Ko
	class_les__collections__perso_1_1_collec...	22/03/2022 11:39	Firefox HTML Doc...	5 Ko

On va par la suite pouvoir observer le rendu que cela a dans une page web :



On peut ici voir l'ensemble des classes déclarer dans le projet, redirigeant chacune plus précisément vers leurs définition :



VI-Conclusion

Nous avons donc vu ici la documentation et plus particulièrement sous Doxygen, comment celui-ci se met en place, s'initialise, les différentes manières qui lui permet d'analyser nos fichiers et enfin le rendu final produit.

VII-Annexes

LISTE DE CLASSE

Classe Collection

```
using System;
using System.Collections.Generic;
using System.Text;

namespace JukeBox
{
    class Biblio
    {
        List<CD> lesCD = new List<CD>();
        public Biblio()
        {
        }

        public void ajouteCD(CD unCD)
        {
            lesCD.Add(new
CD(unCD.getTitre(),unCD.getNom(),unCD.getNbPiste(),unCD.getDuree(),unCD.ge
tStock(),unCD.getCommentaire()));
        }

        public void afficheCD()
        {
        }
    }
}
```

```

        foreach (CD unCd in lesCD)
        {
            Console.WriteLine(unCd);
        }

    }

    List<Videos> lesVideos = new List<Videos>();
    public void ajouteFilm(Videos unFilm)
    {

        lesVideos.Add(new Videos(unFilm.getTitre(),
unFilm.getNomReal(), unFilm.getNomActeur(), unFilm.getDuree(),
unFilm.getStock(), unFilm.getCommentaire()));

    }

    public void afficheFilm()
    {

        foreach (CD unCd in lesCD)
        {
            Console.WriteLine(unCd);
        }

    }

    }
}

```

Classe Mission

```

using System;
using System.Collections.Generic;
using System.Text;

namespace JukeBox
{
    class CD
    {

```

```

private string titre;
private string nom;
private int nbPiste;
private int duree;
private int stock;
private string commentaire;

public CD (string titreSaisi, string nomSaisi, int nbPisteSaisi,
int dureeSaisi, int stockSaisi, string commentaireSaisi)
{
    titre = titreSaisi;

    nom = nomSaisi;

    nbPiste = nbPisteSaisi;

    duree = dureeSaisi;

    stock = stockSaisi;

    commentaire = commentaireSaisi;
}
public void Affiche()
{
    Console.WriteLine("le titre du CD est : " + this.getTitre());
    Console.WriteLine("le nom de l'artiste est : " +
this.getNom());
    Console.WriteLine("le nombre de pistes est : " +
this.getNbPiste());
    Console.WriteLine("le CD dure : " + this.getDuree() + "
secondes ");
    Console.WriteLine("il reste : " + this.getStock() + " Cd ");
    Console.WriteLine("commentaire : " + this.getCommentaire());

}
public string getTitre()
{
    return titre;
}

```

```
public void setTitre(string nouveauTitre)
{
    titre = nouveauTitre;
}
public string getNom()
{
    return nom;
}
public void setNom(string nouveauNom)
{
    nom = nouveauNom;
}
public int getNbPiste()
{
    return nbPiste;
}
public void setNbPiste(int nouveauNbPiste)
{
    nbPiste = nouveauNbPiste;
}
public int getDuree()
{
    return duree;
}
public void setDuree(int nouveauDuree)
{
    duree = nouveauDuree;
}
public int getStock()
{
    return stock;
}
public void setStock(int nouveauStock)
{
    stock = nouveauStock;
}
public string getCommentaire()
{
    return commentaire;
}
```



```
    public void setCommentaire(string nouveauCommentaire)
    {
        commentaire = nouveauCommentaire;
    }
}
```

Classe PdtAch

```
using System;
using System.Collections.Generic;
using System.Text;

namespace JukeBox
{
    class Videos
    {
        private string titre;
        private string nomReal;
        private string nomActeur;
        private int duree;
        private int stock;
        private string commentaire;

        public Videos(string titreSaisi, string nomRealSaisi, string
nomActeurSaisi, int dureeSaisi, int stockSaisi, string commentaireSaisi)
        {
            titre = titreSaisi;

            nomReal = nomRealSaisi;

            nomActeur = nomActeurSaisi;

            duree = dureeSaisi;

            stock = stockSaisi;

            commentaire = commentaireSaisi;
        }
    }
}
```

```

    public void Affiche()
    {
        Console.WriteLine("le titre du film est : " +
this.getTitre());
        Console.WriteLine("le nom du réalisateur est : " +
this.getNomReal());
        Console.WriteLine("le nom de l'acteur principal est : " +
this.getNomActeur());
        Console.WriteLine("le film dure : " + this.getDuree() + "
secondes ");
        Console.WriteLine("il reste : " + this.getStock() + " films en
vente ");
        Console.WriteLine("commentaire critique : " +
this.getCommentaire());
    }
    public string getTitre()
    {
        return titre;
    }
    public void setTitre(string nouveauTitre)
    {
        titre = nouveauTitre;
    }
    public string getNomReal()
    {
        return nomReal;
    }
    public void setNomReal(string nouveauNomReal)
    {
        nomReal = nouveauNomReal;
    }
    public string getNomActeur()
    {
        return nomActeur;
    }
    public void setNom(string nouveauNomActeur)
    {
        nomActeur = nouveauNomActeur;
    }

```

```

    public int getDuree()
    {
        return duree;
    }
    public void setDuree(int nouveauDuree)
    {
        duree = nouveauDuree;
    }
    public int getStock()
    {
        return stock;
    }
    public void setStock(int nouveauStock)
    {
        stock = nouveauStock;
    }
    public string getCommentaire()
    {
        return commentaire;
    }
    public void setCommentaire(string nouveauCommentaire)
    {
        commentaire = nouveauCommentaire;
    }
}
}

```

Classe PdtFab

```

using System;
using System.Collections.Generic;
using System.Text;

namespace JukeBox
{
    class Program
    {
        static void Main(string[] args)
        {
            Biblio uneBibli = new Biblio();

```

```

        CD unCD1 = new CD("Destin", "Ninho", 15, 30000, 2500, "top
album spotify première semaine");
        uneBibli.ajouteCD(unCD1);
        unCD1.Affiche();

        Console.WriteLine("");

        Videos uneVideo1 = new Videos("Shutter Island", "Scorsese",
"Leonardo DiCaprio ", 138, 2500, "DiCaprio aurait dû recevoir un oscar
pour ce film...");
        uneBibli.ajouteFilm(uneVideo1);
        uneVideo1.Affiche();
    }
}
}

```

Classe Produit

```

using System;
using System.Collections.Generic;
using System.Text;

namespace Les_Collections_Perso
{
    class Collection
    {
        public Collection()
        {

        }

        List<Joueur> lesJoueurs = new List<Joueur>();

        public void ajouteJoueur(Joueur unJoueur)
        {

            lesJoueurs.Add(new Joueur(unJoueur.getNomJ(),
unJoueur.getPrenomJ(), unJoueur.getPoints(), unJoueur.getCommentaire()));
        }
    }
}

```

```
}  
  
public void afficheJoueurs()  
{  
  
    foreach (Joueur unJoueur in lesJoueurs)  
    {  
        Console.WriteLine(unJoueur);  
    }  
  
}  
}
```