

Atelier 2 Mission 6 – Tester et documenter

Objectifs de la mission

La mission 6 avait pour objectif de :

- Mettre en place des **tests unitaires** sur les classes métier (Model)
- Réaliser des **tests fonctionnels** de l'API via Postman
- Produire la **documentation technique** de l'application et de l'API
- Créer une **documentation utilisateur sous forme de vidéo**

Cette mission vise à garantir la **fiabilité, la maintenabilité et l'utilisabilité** de l'application MediatekDocuments.

Tests unitaires (Model)

Mise en place

Des tests unitaires ont été réalisés sur les classes du package **Model** afin de vérifier :

- La bonne initialisation des objets
- La cohérence des propriétés
- Le respect des règles métier

Exemples de classes testées :

- Document (Livre, DVD, Revue)
- Exemple
- Commande

Résultats

Les tests ont permis de :

- Valider le bon fonctionnement des modèles
- Détecter d'éventuelles incohérences dans les données
- Fiabiliser la couche métier

Test	Durée	Caractéris...	N
MediatekDocuments.Tests (60)	123 ms		
MediaTekDocuments.Tests (60)	123 ms		
AbonnementHelperTests (2)	11 ms		
CategorieTests (3)	15 ms		
CommandeDocumentTests (4)	22 ms		
CommandeRevueTests (4)	23 ms		
DocumentTests (3)	15 ms		
DvdTests (4)	17 ms		
EtatTests (4)	< 1 ms		
ExemplaireTests (4)	< 1 ms		
GenreTests (3)	< 1 ms		
LivreTests (5)	< 1 ms		
PublicTests (3)	20 ms		
RayonTests (3)	< 1 ms		
RevueTests (4)	< 1 ms		
ServiceTests (3)	< 1 ms		
SuiviTests (4)	< 1 ms		
UtilisateurTests (7)	< 1 ms		

Exemple de code des tests unitaires :

```

1  using Microsoft.VisualStudio.TestTools.UnitTesting;
2  using System;
3  using MediatekDocuments.model;
4
5  namespace MediatekDocuments.Tests
6  {
7      // =====
8      // Tests unitaires - AbonnementHelper
9      // =====
10
11     [TestClass]
12     public class AbonnementHelperTests
13     {
14         [TestMethod]
15         public void ParutionDansAbonnement_Vrai_DansBornes()
16         {
17             DateTime d1 = new DateTime(2026, 1, 1);
18             DateTime d2 = new DateTime(2026, 12, 31);
19
20             Assert.IsTrue(AbonnementHelper.ParutionDansAbonnement(d1, d2, new DateTime(2026, 6, 1)));
21             Assert.IsTrue(AbonnementHelper.ParutionDansAbonnement(d1, d2, new DateTime(2026, 1, 1))); // borne min incluse
22             Assert.IsTrue(AbonnementHelper.ParutionDansAbonnement(d1, d2, new DateTime(2026, 12, 31))); // borne max incluse
23         }
24
25         [TestMethod]
26         public void ParutionDansAbonnement_Faux_HorsBornes()
27         {
28             DateTime d1 = new DateTime(2026, 1, 1);
29             DateTime d2 = new DateTime(2026, 12, 31);
30
31             Assert.IsFalse(AbonnementHelper.ParutionDansAbonnement(d1, d2, new DateTime(2025, 12, 31)));
32             Assert.IsFalse(AbonnementHelper.ParutionDansAbonnement(d1, d2, new DateTime(2027, 1, 1)));
33         }
34     }
35

```

```

36 // =====
37 // Tests unitaires - Document
38 // =====
39
40 [TestClass]
41 0 références
42 public class DocumentTests
43 {
44     3 références | 3/3 passing
45     private Document CreerDocument()
46     {
47         return new Document("D001", "Titre test", "image.jpg",
48             "G1", "Roman", "P1", "Adulte", "R1", "Rayon A");
49     }
50
51     [TestMethod]
52     0 références
53     public void Document_Constructeur_ProprietesBienInitialisees()
54     {
55         Document doc = CreerDocument();
56
57         Assert.AreEqual("D001", doc.Id);
58         Assert.AreEqual("Titre test", doc.Titre);
59         Assert.AreEqual("image.jpg", doc.Image);
60         Assert.AreEqual("G1", doc.IdGenre);
61         Assert.AreEqual("Roman", doc.Genre);
62         Assert.AreEqual("P1", doc.IdPublic);
63         Assert.AreEqual("Adulte", doc.Public);
64         Assert.AreEqual("R1", doc.IdRayon);
65         Assert.AreEqual("Rayon A", doc.Rayon);
66     }
67 }

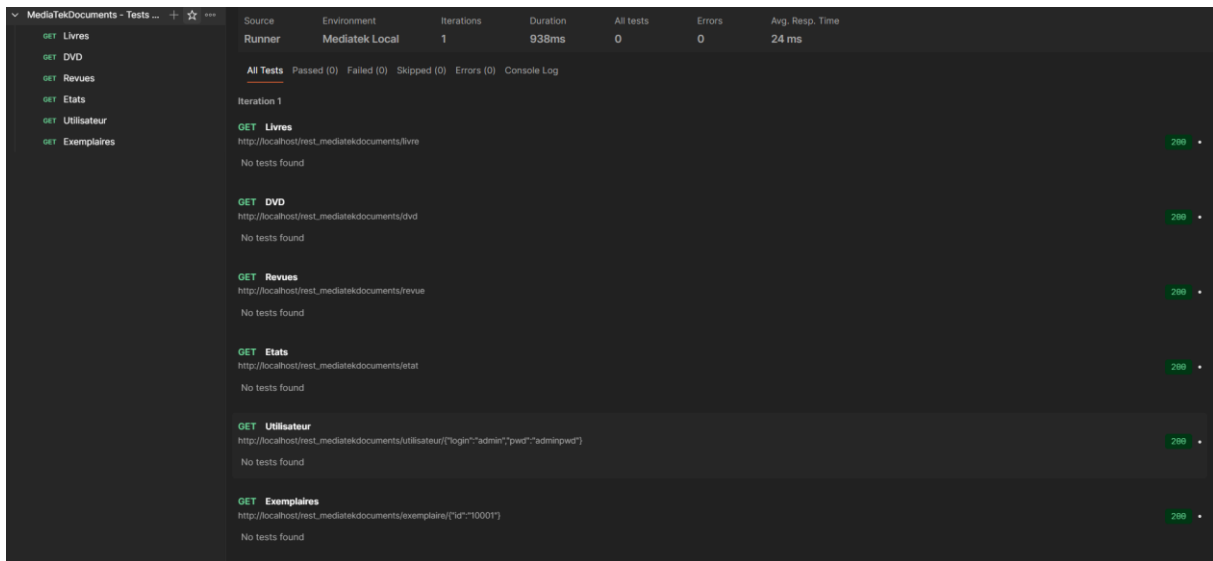
```

```

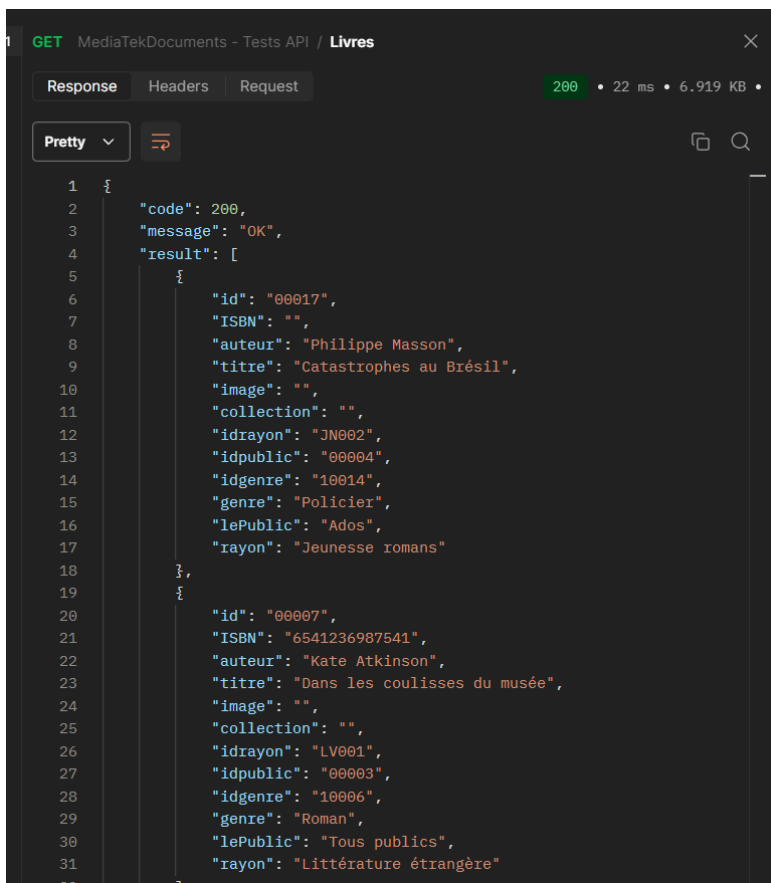
64
65     [TestMethod]
66     0 références
67     public void Document_Id_NonNul()
68     {
69         Document doc = CreerDocument();
70         Assert.IsNotNull(doc.Id);
71     }
72
73     [TestMethod]
74     public void Document_Titre_NonVide()
75     {
76         Document doc = CreerDocument();
77         Assert.IsFalse(string.IsNullOrEmpty(doc.Titre));
78     }
79 }

```

Test dans Postman pour l'accès à la BDD.



Test GET Livres



Test GET DVD

```
1 GET MediaTekDocuments - Tests API / DVD
Response Headers Request 200 • 43 ms • 1.756 KB
Pretty
1 {
2   "code": 200,
3   "message": "OK",
4   "result": [
5     {
6       "id": "20003",
7       "duree": 128,
8       "realisateur": "Steven Spielberg",
9       "titre": "Jurassic Park",
10      "image": "",
11      "synopsis": "Un milliardaire et des g n ticiens cr ent des
          dinosaures   partir de clonage.",
12      "idrayon": "DF001",
13      "idpublic": "00003",
14      "idgenre": "10002",
15      "genre": "Science Fiction",
16      "lePublic": "Tous publics",
17      "rayon": "DVD films"
18    },
19    {
20      "id": "20002",
21      "duree": 228,
22      "realisateur": "Peter Jackson",
23      "titre": "Le seigneur des anneaux : la communaut  de
          l'anneau",
24      "image": "",
25      "synopsis": "L'anneau unique, forg  par Sauron, est port 
          par Fraudon qui l'am ne   Foncombe. De l , des
          repr sentants de peuples diff rents vont s'unir pour
          aider Fraudon   amener l'anneau   la montagne du
          Destin.",
26      "idrayon": "DF001",
27      "idpublic": "00003",
28      "idgenre": "10019",
29      "genre": "Fantazy",
30      "lePublic": "Tous publics",
31      "rayon": "DVD films"
32    }
33  ]
34 }
```

Test GET Revues

view Summary

1 GET MediaTekDocuments - Tests API / Revues

Response Headers Request

200 • 15 ms • 2.627 KB

Pretty

```
1  {}
2  "code": 200,
3  "message": "OK",
4  "result": [
5    {
6      "id": "10002",
7      "periodicite": "MS",
8      "titre": "Alternatives Economiques",
9      "image": "",
10     "delaiMiseADispo": 52,
11     "idrayon": "PR002",
12     "idpublic": "00002",
13     "idgenre": "10015",
14     "genre": "Presse Economique",
15     "lePublic": "Adultes",
16     "rayon": "Magazines"
17   },
18   {
19     "id": "10001",
20     "periodicite": "MS",
21     "titre": "Arts Magazine",
22     "image": "",
23     "delaiMiseADispo": 52,
24     "idrayon": "PR002",
25     "idpublic": "00002",
26     "idgenre": "10016",
27     "genre": "Presse Culturelle",
28     "lePublic": "Adultes",
29     "rayon": "Magazines"
30   },
31 ]
```

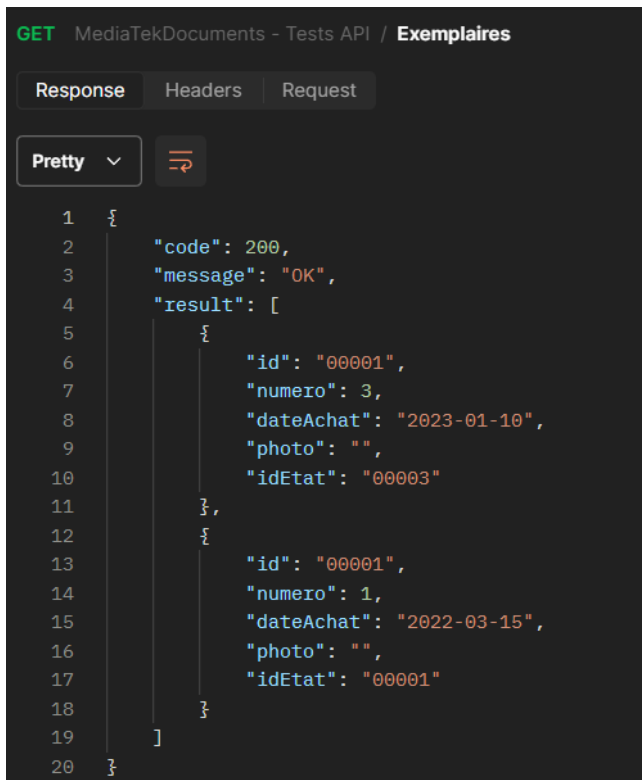
Test GET Etats

```
GET MediaTekDocuments - Tests API / Etats
Response Headers Request
Pretty
1 {
2   "code": 200,
3   "message": "OK",
4   "result": [
5     {
6       "id": "00003",
7       "libelle": "détérioré"
8     },
9     {
10      "id": "00004",
11      "libelle": "inutilisable"
12    },
13    {
14      "id": "00001",
15      "libelle": "neuf"
16    },
17    {
18      "id": "00002",
19      "libelle": "usagé"
20    }
21  ]
22 }
```

Test GET Utilisateur

```
GET MediaTekDocuments - Tests API / Utilisateur
Response Headers Request
Pretty
1 {
2   "code": 200,
3   "message": "OK",
4   "result": [
5     {
6       "login": "admin",
7       "pwd": "admin",
8       "idservice": 1,
9       "LibelleService": "Administrateur"
10    }
11  ]
12 }
```

Test GET Exemples

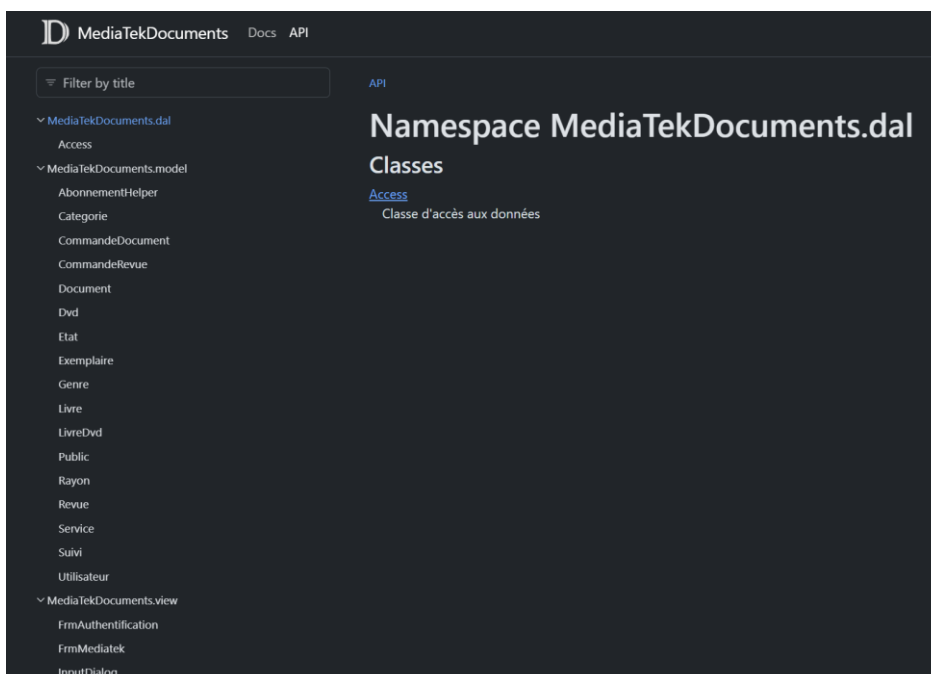


The screenshot shows a REST client interface with a dark theme. At the top, it displays 'GET MediaTekDocuments - Tests API / Exemples'. Below this are tabs for 'Response', 'Headers', and 'Request', with 'Response' selected. A 'Pretty' dropdown menu is set to 'Pretty', and a refresh icon is visible. The response body is a JSON object with the following structure:

```
1 {
2   "code": 200,
3   "message": "OK",
4   "result": [
5     {
6       "id": "00001",
7       "numero": 3,
8       "dateAchat": "2023-01-10",
9       "photo": "",
10      "idEtat": "00003"
11    },
12    {
13      "id": "00001",
14      "numero": 1,
15      "dateAchat": "2022-03-15",
16      "photo": "",
17      "idEtat": "00001"
18    }
19  ]
20 }
```

Documentation technique

Pour l'application C#



The screenshot shows a technical documentation page for a C# application. The page title is 'Namespace MediaTekDocuments.dal' and it lists 'Classes'. The classes listed are: AbonnementHelper, Categorie, CommandeDocument, CommandeRevue, Document, Dvd, Etat, Exemple, Genre, Livre, LivreDvd, Public, Rayon, Revue, Service, Suivi, and Utilisateur. The page also includes a search bar and a filter by title option.

J'ai utilisé DocFX pour la documentation technique.

Exemple Class FrmAuthentification

The screenshot shows the API documentation for the `FrmAuthentification` class. The page title is "Class FrmAuthentification". The namespace is `MediaTekDocuments.view` and the assembly is `MediaTekDocuments.dll`. The description states: "Fenêtre d'authentification : demande login et mot de passe à l'utilisateur. Retourne DialogResult.OK si l'authentification réussit." The class is defined as `public class FrmAuthentification : Form, IDropTarget, ISynchronizeInvoke, IWin32Window, IBindableComponent, IComponent, IDisposable, IContainer`. It inherits from `Form` and implements `IDropTarget`, `ISynchronizeInvoke`, `IWin32Window`, `IBindableComponent`, `IComponent`, `IDisposable`, and `IContainerControl`. The page lists numerous inherited members and implemented methods, including `Form.SetVisibleCore`, `Form.Activate`, `Form.AddOwnedForm`, `Form.AdjustFormScrollbars`, `Form.Close`, `Form.CreateControlsInstance`, `Form.CreateHandle`, `Form.DefWndProc`, `Form.Dispose`, `Form.ProcessMnemonic`, `Form.CenterToParent`, `Form.CenterToScreen`, `Form.LayoutMdi`, `Form.OnActivated`, `Form.OnBackgroundImageChanged`, `Form.OnBackgroundImageLayoutChanged`, `Form.OnClosing`, `Form.OnClosed`, `Form.OnFormClosing`, `Form.OnFormClosed`, `Form.OnCreateControl`, `Form.OnDeactivate`, `Form.OnEnabledChanged`, `Form.OnEnter`, `Form.OnFontChanged`, `Form.OnHandleCreated`, `Form.OnHandleDestroyed`, `Form.OnHelpButtonClicked`, `Form.OnLayout`, `Form.OnLoad`, `Form.OnMaximizedBoundsChanged`, `Form.OnMaximumSizeChanged`, `Form.OnMinimumSizeChanged`, `Form.OnInputLanguageChanged`, `Form.OnInputLanguageChanging`, `Form.OnVisibleChanged`, `Form.OnMdiChildActivate`, `Form.OnMenuStart`, `Form.OnMenuComplete`, `Form.OnPaint`, `Form.OnResize`, `Form.OnDpiChanged`, and `Form.OnGetDpiScaledSize`.

Et pour l'API Rest

Documentation

Namespaces

- Composer
- Autoload
- GrahamCampbell
 - ResultType
- PhpOption
- Dotenv
 - Exception
- Loader
- Parser
- Repository
- Store

Packages

- Application

Reports

- Deprecated
- Errors
- Markers

Indices

- Files

Search (Press "/" to focus)

Documentation

Table of Contents

Packages

- Application

Namespaces

- Composer
- GrahamCampbell
- PhpOption
- Dotenv

Interfaces

- Stringable

Classes

- AccessBDD

Classe qui sollicite ConnexionBDD pour l'accès à la BDD MySQL Elle contient les méthodes appelées par Controle et les méthodes abstraites que MyAccessBDD doit redéfinir pour construire les requêtes

- Connexion

Classe de connexion à la BDD MySQL (singleton) et d'exécution des requêtes en retournant : - pour les requêtes LID : contenu du curseur au format tableau associatif - pour les requêtes LMD : nbre d'enregistrements impactés Dans tous les cs, 'null' est renvoyé si la requête échpie.

- Controle

Contrôleur : reçoit et traite les demandes du point d'entrée

J'ai utilisé phpDocumentor

Exemple Connexion.php

Documentation

Namespaces

- Composer
 - Autoload
- GrahamCampbell
 - ResultType
- PhpOption
- Dotenv
 - Exception
 - Loader
 - Parser
 - Repository
 - Store

Packages

- Application

Reports

- Deprecated
- Errors
- Markers

Indices

- Files

Connexion

[Connexion.php](#) : 9

in package Application

Classe de connexion à la BDD MySQL (singleton) et d'exécution des requêtes en retournant : - pour les requêtes LID : contenu du curseur au format tableau associatif - pour les requêtes LMD : nbre d'enregistrements impactés Dans tous les cs, 'null' est renvoyé si la requête échpie.

Table of Contents

Properties

- [\\$conn](#) : PDO
- [\\$instance](#) : Connexion

Methods

- [beginTransaction\(\)](#) : bool
Démarre une transaction SQL.
- [commit\(\)](#) : bool
Valide définitivement la transaction en cours.
- [getInstance\(\)](#) : Connexion
méthode statique de création de l'instance unique
- [queryBDD\(\)](#) : array<string|int, mixed>|null
exécution une requête select retournant 0 à plusieurs lignes
- [rollback\(\)](#) : bool
Annule la transaction en cours.

🔍 Search (Press "/" to focus)

On this page

Table Of Contents

[Properties](#)

[Methods](#)

Properties

[\\$conn](#)

[\\$instance](#)

Methods

[beginTransaction\(\)](#)

[commit\(\)](#)

[getInstance\(\)](#)

[queryBDD\(\)](#)

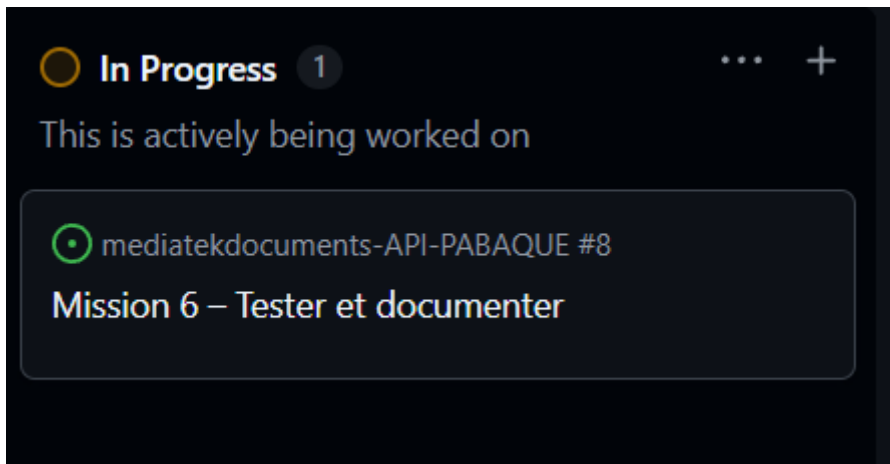
[rollback\(\)](#)

[updateBDD\(\)](#)

[__construct\(\)](#)

[prepareRequest\(\)](#)

KanBan :



Mission 6 – Tester et documenter #8

Open pab1307/mediatekdocuments-API-PABAQUE Public

pab1307 opened on Jan 26 · edited by pab1307 Edits Owner

Tâche 1 – Gérer les tests

Écrire les tests unitaires sur les classes du package Model (en plus du test unitaire écrit précédemment).

Construire une collection de tests dans Postman pour contrôler les fonctionnalités de l'API d'accès à la BDD.

Consultez l'article "[Créer des tests avec Postman](#)" dans le wiki du dépôt distant.

Tâche 2 – Créer les documentations techniques

Contrôler, dans chaque application, que les commentaires normalisés sont bien tous ajoutés et corrects.

Générer la documentation technique de l'application C#, en suivant les explications données dans l'article "[Documentation technique sous Visual Studio](#)".

Générer la documentation technique de l'API REST, en suivant les explications données dans l'article "[Génération de la documentation technique sous Netbeans](#)".

Transférer les documentations dans les dépôts.

Tâche 3 : créer la documentation utilisateur en vidéo

Créer une vidéo de 10mn maximum qui présente l'ensemble des fonctionnalités de l'application C#.

Temps estimé 8h
Temps réel 7h15

Create sub-issue