Atelier 2 Mission 3 – Gestion du suivi de l'état des exemplaires

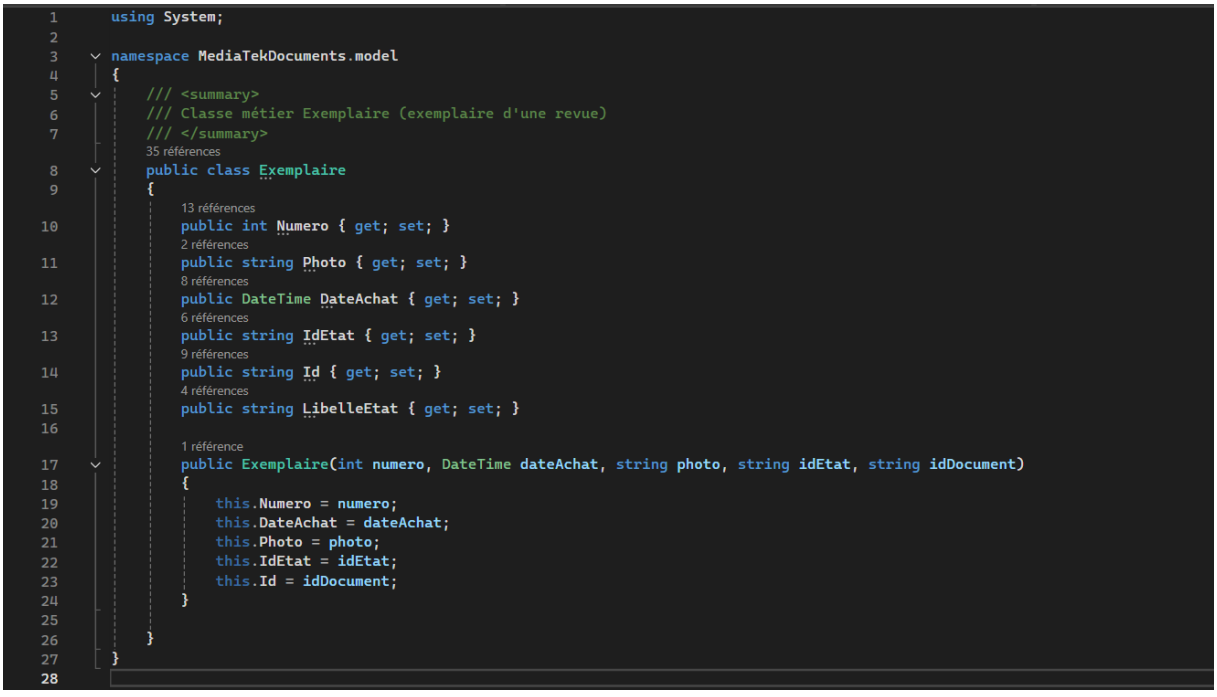
Contexte de la mission

La mission 3 avait pour objectif de mettre en place la gestion de l'état des exemplaires dans l'application MediatekDocuments.
D'après l'énoncé, il devait être possible de suivre l'état des exemplaires de livres, de DVD et de revues/parutions, avec les états prédéfinis « **neuf** », « **usagé** », « **détérioré** » et « **inutilisable** ».
L'application devait également permettre de modifier l'état d'un exemplaire et de supprimer un exemplaire.

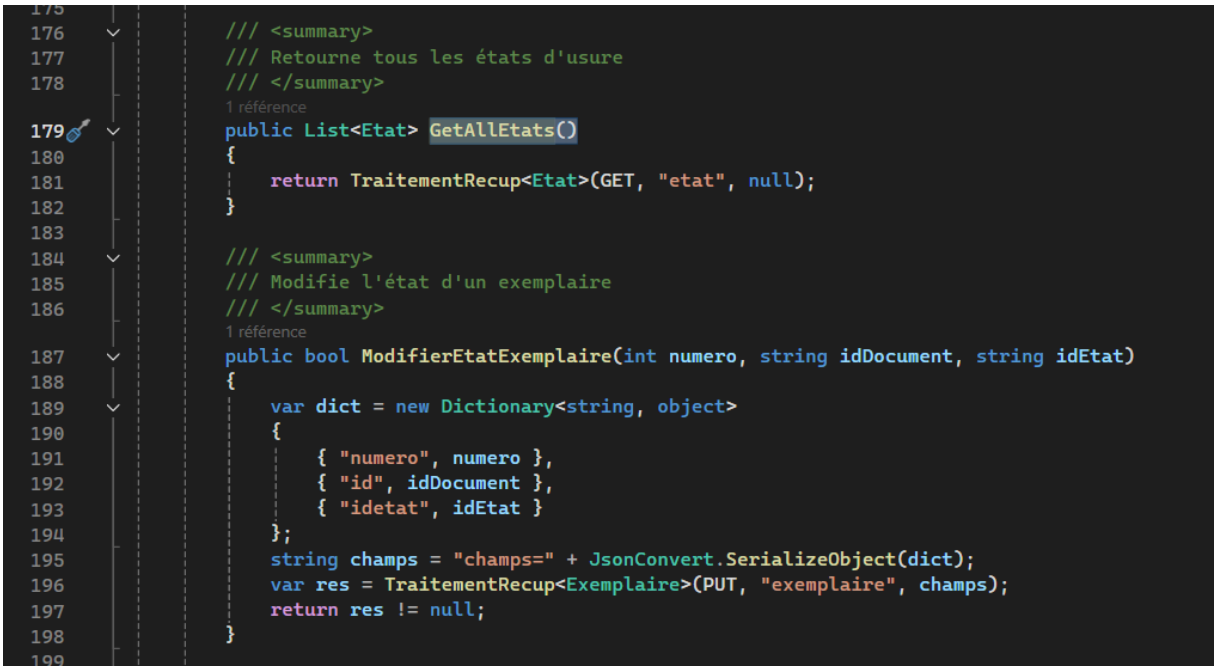
Ajout de la capacité de chercher, modifier et supprimé un exemplaire

Dans la couche **Model**, le fichier model/Exemplaire.cs a été complété par l'ajout de la propriété **LibelleEtat**, afin de pouvoir récupérer et afficher directement le libellé de l'état d'usure de chaque exemplaire.

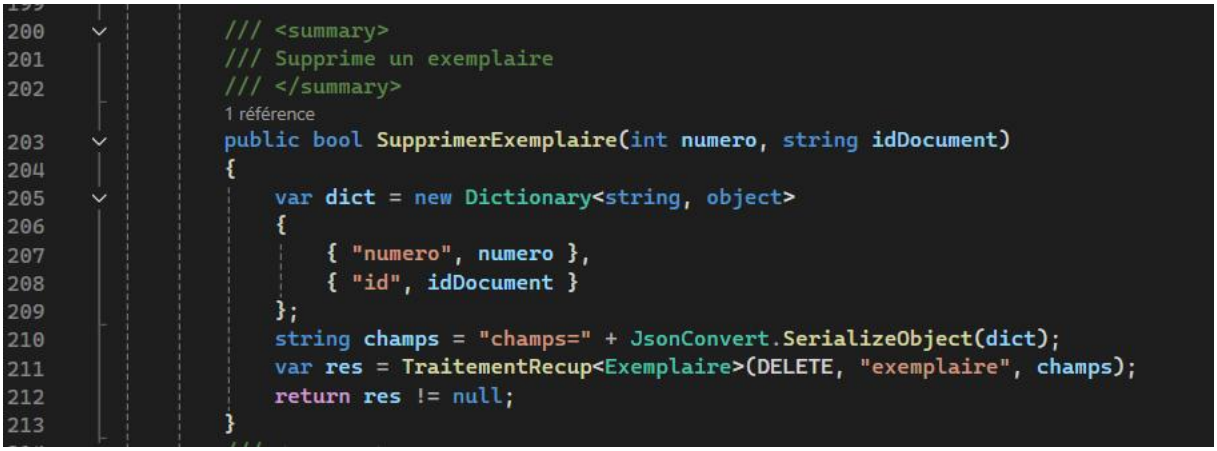
Dans la couche **DAL**, le fichier dal/Access.cs a été enrichi avec plusieurs méthodes permettant de communiquer avec l'API :



```
using System;

namespace MediaTekDocuments.model
{
    /// <summary>
    /// Classe métier Exemplaire (exemplaire d'une revue)
    /// </summary>
    35 références
    public class Exemplaire
    {
        13 références
        public int Numero { get; set; }
        2 références
        public string Photo { get; set; }
        8 références
        public DateTime DateAchat { get; set; }
        6 références
        public string IdEtat { get; set; }
        9 références
        public string Id { get; set; }
        4 références
        public string LibelleEtat { get; set; }

        1 référence
        public Exemplaire(int numero, DateTime dateAchat, string photo, string idEtat, string idDocument)
        {
            this.Numero = numero;
            this.DateAchat = dateAchat;
            this.Photo = photo;
            this.IdEtat = idEtat;
            this.Id = idDocument;
        }

    }
}
```



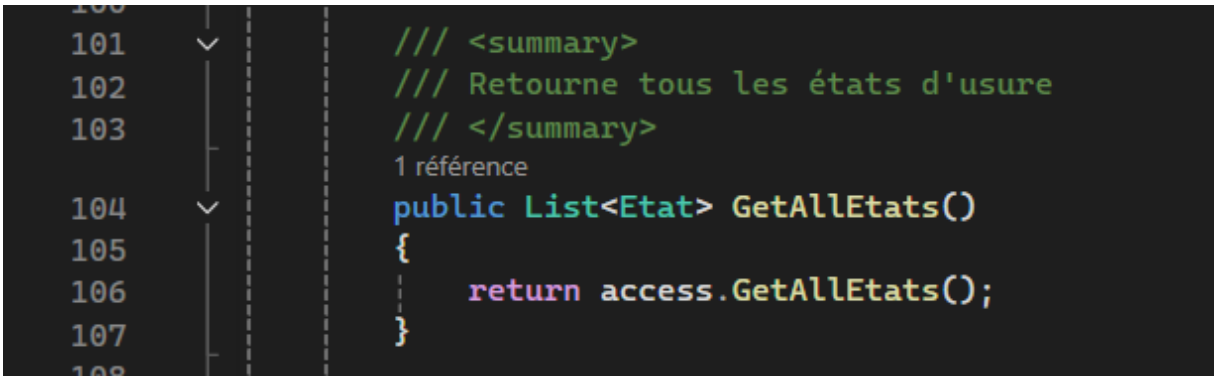
```
175
176     /// <summary>
177     /// Retourne tous les états d'usure
178     /// </summary>
        1 référence
179     public List<Etat> GetAllEtats()
180     {
181         return TraitementRecup<Etat>(GET, "etat", null);
182     }
183
184     /// <summary>
185     /// Modifie l'état d'un exemplaire
186     /// </summary>
        1 référence
187     public bool ModifierEtatExemplaire(int numero, string idDocument, string idEtat)
188     {
189         var dict = new Dictionary<string, object>
190         {
191             { "numero", numero },
192             { "id", idDocument },
193             { "idetat", idEtat }
194         };
195         string champs = "champs=" + JsonConvert.SerializeObject(dict);
196         var res = TraitementRecup<Exemplaire>(PUT, "exemplaire", champs);
197         return res != null;
198     }
199
```



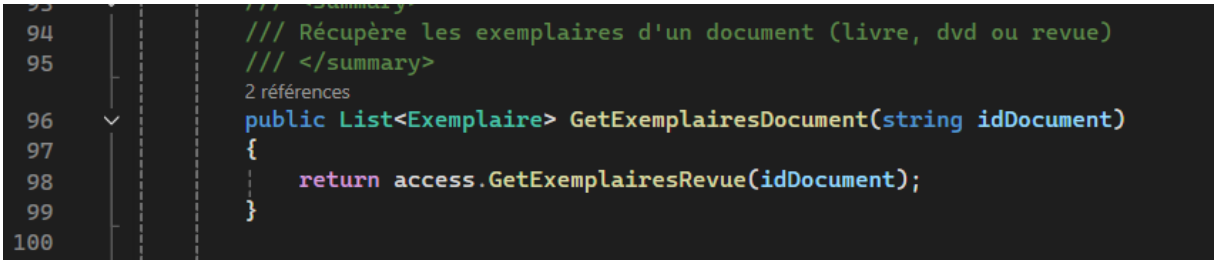
```
199
200     /// <summary>
201     /// Supprime un exemplaire
202     /// </summary>
        1 référence
203     public bool SupprimerExemplaire(int numero, string idDocument)
204     {
205         var dict = new Dictionary<string, object>
206         {
207             { "numero", numero },
208             { "id", idDocument }
209         };
210         string champs = "champs=" + JsonConvert.SerializeObject(dict);
211         var res = TraitementRecup<Exemplaire>(DELETE, "exemplaire", champs);
212         return res != null;
213     }
```

- GetAllEtats() pour récupérer la liste des états disponibles ;
- ModifierEtatExemplaire(numero, idDocument, idEtat) pour envoyer une requête PUT vers l'API afin de modifier l'état d'un exemplaire ;
- SupprimerExemplaire(numero, idDocument) pour envoyer une requête DELETE et supprimer un exemplaire.

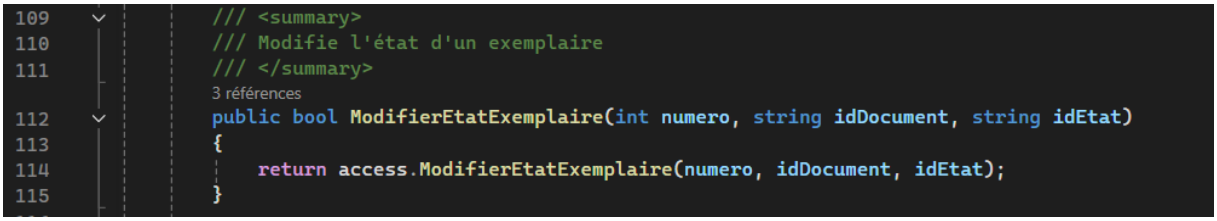
Ces méthodes permettent d'assurer le lien entre l'interface et les données distantes. Elles rendent possible la consultation de la liste des états existants, la mise à jour d'un état d'usure et la suppression d'un exemplaire devenu inutilisable ou erroné.
Le passage par la DAL permet également de conserver une architecture claire, en séparant les traitements liés à l'accès aux données des traitements liés à l'interface.



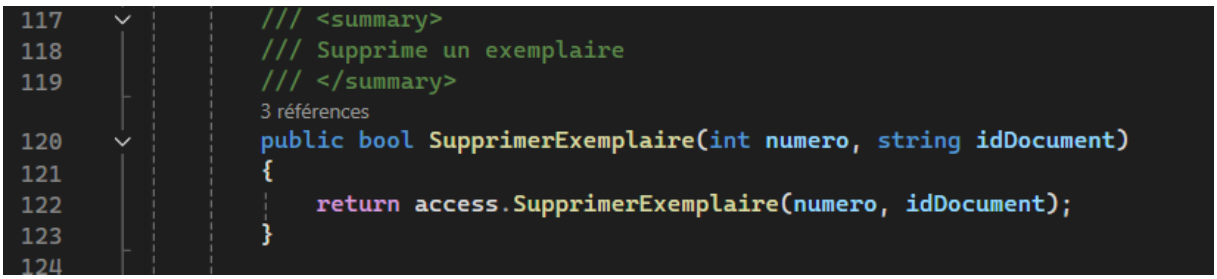
```
101     /// <summary>
102     /// Retourne tous les états d'usure
103     /// </summary>
        1 référence
104     public List<Etat> GetAllEtats()
105     {
106         return access.GetAllEtats();
107     }
```



```
94      /// Récupère les exemplaires d'un document (livre, dvd ou revue)
95      /// </summary>
        2 références
96      public List<Exemplaire> GetExemplairesDocument(string idDocument)
97      {
98          return access.GetExemplairesRevue(idDocument);
99      }
100
```



```
109     /// <summary>
110     /// Modifie l'état d'un exemplaire
111     /// </summary>
        3 références
112     public bool ModifierEtatExemplaire(int numero, string idDocument, string idEtat)
113     {
114         return access.ModifierEtatExemplaire(numero, idDocument, idEtat);
115     }
```



```
117     /// <summary>
118     /// Supprime un exemplaire
119     /// </summary>
        3 références
120     public bool SupprimerExemplaire(int numero, string idDocument)
121     {
122         return access.SupprimerExemplaire(numero, idDocument);
123     }
124
```

- GetAllEtats() ;
- ModifierEtatExemplaire() ;
- SupprimerExemplaire() ;

Une méthode générique GetExemplairesDocument() a également été mise en place afin de récupérer les exemplaires d'un document, qu'il s'agisse d'un livre, d'un DVD ou d'une revue. L'utilisation d'une méthode générique constitue ici un choix pertinent, car elle évite de dupliquer le même traitement pour chaque type de document. Cela permet de mutualiser la logique de récupération des exemplaires et de rendre le code plus lisible, plus maintenable et plus facile à faire évoluer.

Ajout de l'interface graphique

Au niveau de l'interface graphique, la fenêtre principale a d'abord été agrandie dans **view/FrmMediatek.Designer.cs**, afin de disposer de plus d'espace pour intégrer la gestion des exemplaires.

Dans **view/FrmMediatek.cs**, plusieurs aménagements ont ensuite été réalisés :

```
288     /// <summary>
289     /// Constructeur : création du contrôleur lié à ce formulaire
290     /// </summary>
        1 référence
291     internal FrmMediatek()
292     {
293         InitializeComponent();
294         this.controller = new FrmMediatekController();
295         lesSuivis = controller.GetAllSuivis();
296         lesEtats = controller.GetAllEtats() ?? new List<Etat>();
297         InitialiserControlesExemplaires();
298         AlerterRevuesFinAbonnement();
299     }
```

```
2344    private void InitialiserControlesExemplaires()
2345    {
2346        InitLivresExemplairesControles();
2347        InitDvdExemplairesControles();
2348        InitParutionsExemplairesControles();
2349    }
2350
        1 référence
2351    private void InitLivresExemplairesControles()
2352    {
2353        var grp = new GroupBox
2354        {
2355            Text = "Exemplaires du livre sélectionné",
2356            Location = new Point(11, 779),
2357            Size = new Size(1145, 210),
2358            TabIndex = 100
2359        };
```

```
1 référence
private void InitParutionsExemplairesControles()
{
    var grp = new GroupBox
    {
        Text = "Gestion état / Suppression de la parution sélectionnée",
        Location = new Point(11, 775),
        Size = new Size(1145, 70),
        TabIndex = 50
    };

    var lblEtat = new Label { Text = "État :", Location = new Point(12, 30), AutoSize = true };

    cbxParutionsEtat = new ComboBox
    {
        Location = new Point(55, 26),
        Size = new Size(250, 25),
        DropDownStyle = ComboBoxStyle.DropDownList,
        TabIndex = 0
    };
    cbxParutionsEtat.DataSource = lesEtats;
    cbxParutionsEtat.DisplayMember = "Libelle";
    cbxParutionsEtat.ValueMember = "Id";

    btnParutionsChgtEtat = new Button
    {
        Text = "Changer état",
        Location = new Point(320, 24),
        Size = new Size(140, 30),
        Enabled = false,
        TabIndex = 1
    };
    btnParutionsChgtEtat.Click += BtnParutionsChgtEtat_Click;

    btnParutionsSupprExemplaire = new Button
    {
        Text = "Supprimer parution",
        Location = new Point(475, 24),
        Size = new Size(160, 30),
        Enabled = false,
        TabIndex = 2
    };
```

```
    btnParutionsSupprExemplaire.Click += BtnParutionsSupprExemplaire_Click;

    grp.Controls.Add(lblEtat);
    grp.Controls.Add(cbxParutionsEtat);
    grp.Controls.Add(btnParutionsChgtEtat);
    grp.Controls.Add(btnParutionsSupprExemplaire);

    tabReceptionRevue.Controls.Add(grp);
}
```

```
5 références
private void ChargerExemplairesLivre(string idLivre)
{
    livreExemplaireCourant = null;
    btnLivresChgtEtat.Enabled = false;
    btnLivresSupprExemplaire.Enabled = false;

    if (string.IsNullOrEmpty(idLivre))
    {
        bdgLivresExemplaires.DataSource = null;
        return;
    }

    lesExemplairesLivre = controller.GetExemplairesDocument(idLivre) ?? new List<Exemplaire>();
    AppliquerLibelleEtat(lesExemplairesLivre);
    lesExemplairesLivre = lesExemplairesLivre.OrderByDescending(e => e.DateAchat).ToList();

    bdgLivresExemplaires.DataSource = null;
    bdgLivresExemplaires.DataSource = lesExemplairesLivre;
    dgvLivresExemplaires.DataSource = bdgLivresExemplaires;
    MasquerColonnesExemplaire(dgvLivresExemplaires);
}
```

```
2624        private void ChargerExemplairesDvd(string idDvd)
2625        {
2626            dvdExemplaireCourant = null;
2627            btnDvdChgtEtat.Enabled = false;
2628            btnDvdSupprExemplaire.Enabled = false;
2629
2630            if (string.IsNullOrEmpty(idDvd))
2631            {
2632                bdgDvdExemplaires.DataSource = null;
2633                return;
2634            }
2635
2636            lesExemplairesDvd = controller.GetExemplairesDocument(idDvd) ?? new List<Exemplaire>();
2637            AppliquerLibelleEtat(lesExemplairesDvd);
2638            lesExemplairesDvd = lesExemplairesDvd.OrderByDescending(e => e.DateAchat).ToList();
2639
2640            bdgDvdExemplaires.DataSource = null;
2641            bdgDvdExemplaires.DataSource = lesExemplairesDvd;
2642            dgvDvdExemplaires.DataSource = bdgDvdExemplaires;
2643            MasquerColonnesExemplaire(dgvDvdExemplaires);
2644        }
```

```
                3 références
2738        private void AppliquerLibelleEtat(List<Exemplaire> exemplaires)
2739        {
2740            foreach (var ex in exemplaires)
2741            {
2742                var etat = lesEtats.FirstOrDefault(et => et.Id == ex.IdEtat);
2743                ex.LibelleEtat = etat?.Libelle ?? ex.IdEtat;
2744            }
2745        }
2746
                2 références
2747        private void MasquerColonnesExemplaire(DataGridView dgv)
2748        {
2749            if (dgv.Columns["Photo"] != null) dgv.Columns["Photo"].Visible = false;
2750            if (dgv.Columns["photo"] != null) dgv.Columns["photo"].Visible = false;
2751            if (dgv.Columns["IdEtat"] != null) dgv.Columns["IdEtat"].Visible = false;
2752            if (dgv.Columns["idEtat"] != null) dgv.Columns["idEtat"].Visible = false;
2753            if (dgv.Columns["Id"] != null) dgv.Columns["Id"].Visible = false;
2754            if (dgv.Columns["id"] != null) dgv.Columns["id"].Visible = false;
2755            dgv.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.AllCells;
2756        }
```

```
                2 références
2747        private void MasquerColonnesExemplaire(DataGridView dgv)
2748        {
2749            if (dgv.Columns["Photo"] != null) dgv.Columns["Photo"].Visible = false;
2750            if (dgv.Columns["photo"] != null) dgv.Columns["photo"].Visible = false;
2751            if (dgv.Columns["IdEtat"] != null) dgv.Columns["IdEtat"].Visible = false;
2752            if (dgv.Columns["idEtat"] != null) dgv.Columns["idEtat"].Visible = false;
2753            if (dgv.Columns["Id"] != null) dgv.Columns["Id"].Visible = false;
2754            if (dgv.Columns["id"] != null) dgv.Columns["id"].Visible = false;
2755            dgv.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.AllCells;
2756        }
2757
2758        #endregion
2759    }
2760 }
2761
```

```
2396            btnLivresChgtEtat.Click += BtnLivresChgtEtat_Click;
```

```
2462            btnDvdChgtEtat.Click += BtnDvdChgtEtat_Click;
```

```
2514            btnParutionsChgtEtat.Click += BtnParutionsChgtEtat_Click;
```

```
1 référence
private void BtnLivresSupprExemplaire_Click(object sender, EventArgs e)
{
    if (livreExemplaireCourant == null) return;

    DialogResult rep = MessageBox.Show(
        $"Supprimer l'exemplaire n°{livreExemplaireCourant.Numero} ?",
        "Confirmation", MessageBoxButtons.YesNo, MessageBoxIcon.Warning);
    if (rep != DialogResult.Yes) return;

    string idDoc = livreExemplaireCourant.Id;
    bool ok = controller.SupprimerExemplaire(livreExemplaireCourant.Numero, idDoc);
    MessageBox.Show(ok ? "Exemplaire supprimé." : "Échec de la suppression.");
    if (ok) ChargerExemplairesLivre(idDoc);
}
```

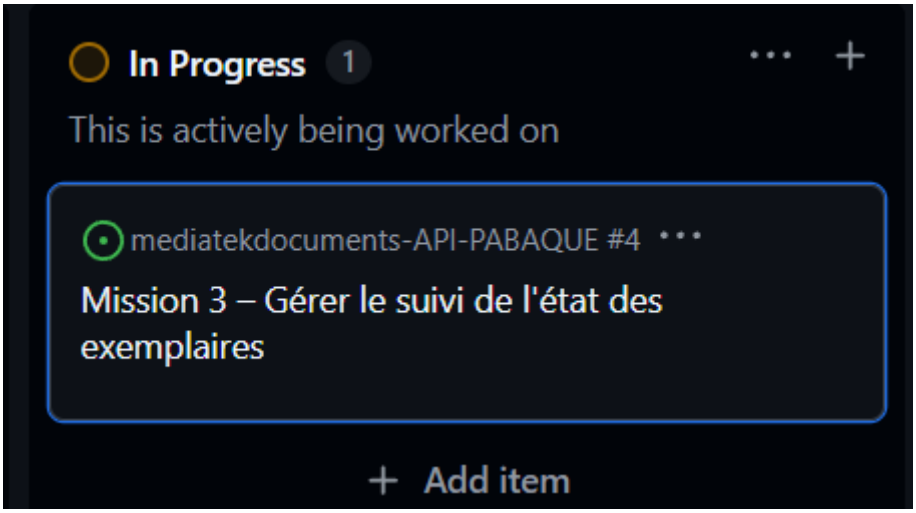
```
1 référence
private void BtnDvdSupprExemplaire_Click(object sender, EventArgs e)
{
    if (dvdExemplaireCourant == null) return;

    DialogResult rep = MessageBox.Show(
        $"Supprimer l'exemplaire n°{dvdExemplaireCourant.Numero} ?",
        "Confirmation", MessageBoxButtons.YesNo, MessageBoxIcon.Warning);
    if (rep != DialogResult.Yes) return;

    string idDoc = dvdExemplaireCourant.Id;
    bool ok = controller.SupprimerExemplaire(dvdExemplaireCourant.Numero, idDoc);
    MessageBox.Show(ok ? "Exemplaire supprimé." : "Échec de la suppression.");
    if (ok) ChargerExemplairesDvd(idDoc);
}
```

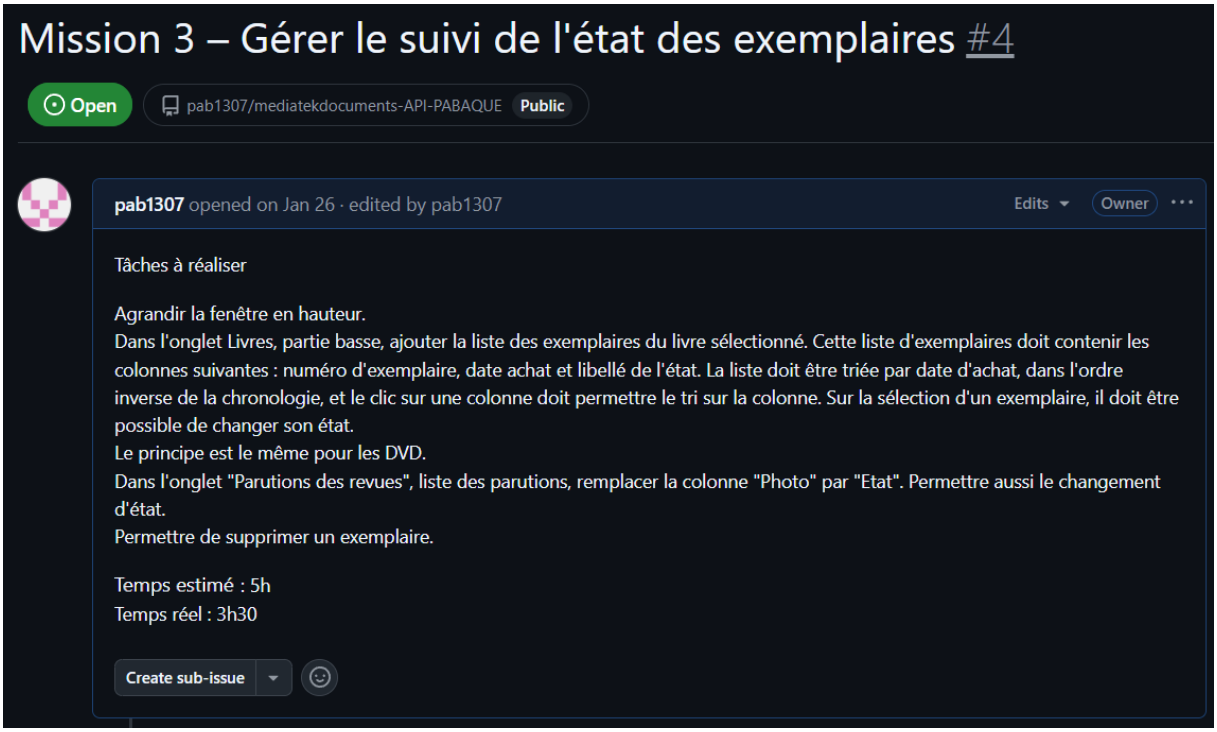
```
1 référence
private void BtnParutionsSupprExemplaire_Click(object sender, EventArgs e)
{
    if (paritionExemplaireCourant == null) return;

    DialogResult rep = MessageBox.Show(
        $"Supprimer la parution n°{paritionExemplaireCourant.Numero} ?",
        "Confirmation", MessageBoxButtons.YesNo, MessageBoxIcon.Warning);
    if (rep != DialogResult.Yes) return;

    string idDoc = paritionExemplaireCourant.Id;
    bool ok = controller.SupprimerExemplaire(paritionExemplaireCourant.Numero, idDoc);
    MessageBox.Show(ok ? "Parution supprimée." : "Échec de la suppression.");
    if (ok) AfficheReceptionExemplairesRevue();
}
```

- pour l'onglet **Livres**, une section **« Exemplaires »** a été créée dynamiquement en bas de l'onglet ;
- pour l'onglet **DVD**, le même principe a été appliqué ;
- pour l'onglet **Parutions**, la colonne **« Photo »** a été remplacée par **LibelleEtat**, et un groupe dédié à la **gestion de l'état** et à la **suppression** a été ajouté ;
- le **chargement automatique des exemplaires** a été mis en place lors de la sélection d'un livre ou d'un DVD ;
- un **tri par date d'achat décroissant par défaut** a également été appliqué ;
- enfin, le **tri par clic sur les colonnes** a été rendu possible dans les grilles.

Ces modifications ont permis d'améliorer l'ergonomie générale de l'application. Lorsqu'un utilisateur sélectionne un document, les exemplaires associés sont affichés automatiquement, ce qui facilite la consultation. Le remplacement de certaines colonnes et l'ajout de zones spécifiques rendent l'interface plus adaptée au suivi des exemplaires et à la gestion de leur état. Le tri automatique par date d'achat permet en outre de mettre en avant les exemplaires les plus récents, tandis que le tri manuel sur les colonnes offre davantage de souplesse à l'utilisateur lors de la consultation.

KanBan :


In Progress 1
This is actively being worked on
mediatekdocuments-API-PABAQUE #4
Mission 3 – Gérer le suivi de l'état des exemplaires
Add item


Mission 3 – Gérer le suivi de l'état des exemplaires #4
Open
pab1307/mediatekdocuments-API-PABAQUE Public
pab1307 opened on Jan 26 · edited by pab1307
Edits
Owner
Tâches à réaliser
Agrandir la fenêtre en hauteur.
Dans l'onglet Livres, partie basse, ajouter la liste des exemplaires du livre sélectionné. Cette liste d'exemplaires doit contenir les colonnes suivantes : numéro d'exemplaire, date achat et libellé de l'état. La liste doit être triée par date d'achat, dans l'ordre inverse de la chronologie, et le clic sur une colonne doit permettre le tri sur la colonne. Sur la sélection d'un exemplaire, il doit être possible de changer son état.
Le principe est le même pour les DVD.
Dans l'onglet "Parutions des revues", liste des parutions, remplacer la colonne "Photo" par "Etat". Permettre aussi le changement d'état.
Permettre de supprimer un exemplaire.
Temps estimé : 5h
Temps réel : 3h30
Create sub-issue