

Compte rendu de l'atelier de professionnalisation n°2

Sommaire

Contexte 2/24

Mission 0 – préparer l'environnement de travail 3/24

Mission 2 – gérer les commandes 4/24

Mission 4 – mettre en place des authentifications 10/24

Mission 5 – assurer la sécurité, la qualité et intégrer des logs 14/24

Mission 6 – tester et documenter 18/24

Mission 7 – Déployer et gérer les sauvegardes de données 21/24

Contexte

Mediatek86 est un ensemble de médiathèques de la Vienne. Les médiathèques mettent à disposition des documents sur divers supports, proposent des formations sur site et offrent des vidéos d'auto-formation. Mediatek86 a fait appel à Infotech Services 86, une ESN spécialisée, entre autres, dans le développement informatique. Faire évoluer une application de bureau (C#) exploitant une API REST (PHP) pour l'accès à une base de données relationnelle MySQL et qui permet de gérer les documents des vidéothèques de la chaîne MediaTek86.

Mission 0 : préparer l'environnement de travail

Je récupère l'application sur le repo github d'où je dois la récupérer, que je setup avec les extensions nuget :

	Microsoft.AspNet.WebApi par Microsoft	5.3.0
This package contains everything you need to host ASP.NET Web API on IIS.		
	System.Net.Http par Microsoft	4.3.4
Provides a programming interface for modern HTTP applications, including HTTP client components that allow applications to consume web services over HTTP and HTTP components that can be used by both clients and servers for parsing HTTP headers.		
	Newtonsoft.Json par James Newton-King	13.0.1
Json.NET is a popular high-performance JSON framework for .NET		
		13.0.3

Ensuite, je récupère l'API rest que je transfère dans le dossier /www/ de Wamp, et j'importe la bdd mediatek86.sql dans phpmyadmin, et je teste l'api avec postman avec les credentials admin:adminpwd. J'ouvre l'API sur Visual Studio Code/Netbeans (j'utilise les deux) et c'est terminé.

Mission 2 : gérer les commandes

Je crée la vue sur le modèle des précédents onglets :

```
/// <summary>
/// Sur la sélection d'une ligne ou cellule dans le dgv
/// Affichage des commandes du livres
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
1 référence | didlyn9, Il y a 20 heures | 1 auteur, 1 modification
private void dgvLivresComListe_SelectionChanged(object sender, EventArgs e)
{
    if (dgvLivresComListe.CurrentCell != null)
    {
        try
        {
            Livre livre = (Livre)bdgLivresComListe.List[bdgLivresComListe.Position];
            AfficheLivresCommandeInfos(livre);
            txbLivresComNumLivre.Text = livre.Id;
        }
        catch
        {
            VideLivresComZones();
        }
    }
    else
    {
        txbLivresComNumLivre.Text = "";
        VideLivresComInfos();
    }
}
```

```

private void AfficheLivresCommandeInfos(Livre livre)
{
    string idLivre = livre.Id;
    VideLivresComInfos();
    lesCommandes = controller.GetCommandesLivres(idLivre);
    grpLivresCommandes.Text = livre.Titre + " de " + livre.Auteur;
    if (lesCommandes.Count == 0)
        VideLivresComInfos();
    RemplirLivresComListeCommandes(lesCommandes);
}

/// <summary>
/// Tri sur les colonnes
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
1 référence | didlyn9, Il y a 20 heures | 1 auteur, 1 modification
private void dgvLivresComListe_ColumnHeaderMouseClick(object sender, DataGridViewCellMouseEventArgs e)
{
    VideLivresComZones();
    string titreColonne = dgvLivresComListe.Columns[e.ColumnIndex].HeaderText;
    List<Livre> sortedList = new List<Livre>();
    switch (titreColonne)
    {
        case "Id":
            sortedList = lesLivresCom.OrderBy(o => o.Id).ToList();
            break;
        case "Titre":
            sortedList = lesLivresCom.OrderBy(o => o.Titre).ToList();
            break;
        case "Collection":
            sortedList = lesLivresCom.OrderBy(o => o.Collection).ToList();
            break;
        case "Auteur":
            sortedList = lesLivresCom.OrderBy(o => o.Auteur).ToList();
            break;
        case "Genre":
            sortedList = lesLivresCom.OrderBy(o => o.Genre).ToList();
            break;
        case "Public":
            sortedList = lesLivresCom.OrderBy(o => o.Public).ToList();
            break;
        case "Rayon":
            sortedList = lesLivresCom.OrderBy(o => o.Rayon).ToList();
            break;
    }
    RemplirLivresComListe(sortedList);
}

```

implémentation du tri dans le DataGridView (avec l'événement ColumnHeaderMouseClick)

Création des classes Commande et CommandeDocument :

```
using System;
```

```
namespace MediaTekDocuments.model
{
    public class Commande
    {
        public string Id { get; }

        public DateTime DateCommande { get; set; }

        public double Montant { get; }

        public Commande(string id, DateTime dateCommande, double montant)
        {
            Id = id;
            DateCommande = dateCommande;
            Montant = montant;
        }
    }
}
```

```
using System;
```

```
namespace MediaTekDocuments.model
{
    public class CommandeDocument : Commande
    {
        public int NbExemplaire { get; }

        public string IdLivreDvd { get; }

        public int IdSuivi { get; }

        public string Etat { get; }

        public CommandeDocument(string id, DateTime dateCommande, float montant, int nbExemplaire,
            string idLivreDvd, int idSuivi, string etat)
            : base(id, dateCommande, montant)
        {
            this.NbExemplaire = nbExemplaire;
            this.IdLivreDvd = idLivreDvd;
            this.IdSuivi = idSuivi;
            this.Etat = etat;
        }
    }
}
```

Dans Access :

```
public List<CommandeDocument> GetCommandesLivres(string idLivre)
{
    String jsonIdDocument = ConvertToJson("idLivreDvd", idLivre);
    List<CommandeDocument> lesCommandesLivres = TraitementRecup<CommandeDocument>(GET, "commandedocument/" + jsonIdDocument);
    return lesCommandesLivres;
}
```

Dans l'API :

```
/**
 * requete arrivée en POST (insert)
 * @param string $table nom de la table
 * @param array $champs nom et valeur des champs
 */
public function post($table, $champs){
    $result = null;
    if ($table == "commandedocument"){
        $result = $this->accessBDD->insertCommande($champs);
    }else if ($table == "abonnement"){
        $result = $this->accessBDD->insertAbonnement($champs);
    }else{
        $result = $this->accessBDD->insertOne($table, $champs);
    }
    if ($result == null || $result == false) {
        $this->reponse(400, "requete invalide");
    }else{
        $this->reponse(200, "OK");
    }
}
```

```
/**
 * requete arrivée en PUT (update)
 * @param string $table nom de la table
 * @param string $id valeur de l'id
 * @param array $champs nom et valeur des champs
 */
public function put($table, $id, $champs){
    $result = null;
    if ($table == "commandedocument"){
        $result = $this->accessBDD->updateCommande($id, $champs);
    }elseif ($table == "abonnement"){
        $result = $this->accessBDD->updateAbonnement($id, $champs);
    }elseif ($table == "exemplaire"){
        $result = $this->accessBDD->updateOne($table, $id, $champs, $champs["Numero"]);
    }else{
        $result = $this->accessBDD->updateOne($table, $id, $champs);
    }
    if ($result == null || $result == false){
        $this->reponse(400, "requete invalide");
    }else{
        $this->reponse(200, "OK");
    }
}
```

```

public function insertAbonnement($champs)
{
    $champsCommande = [ "id" => $champs["Id"], "dateCommande" => $champs["DateCommande"],
        "montant" => $champs["Montant"]];
    $champsAbonnement = [ "id" => $champs["Id"], "dateFinAbonnement" => $champs["DateFinAbonnement"],
        "idRevue" => $champs["IdRevue"]];
    $result = $this->insertOne("commande", $champsCommande);
    if ($result == null || $result == false){
        return null;
    }
    return $this->insertOne( "abonnement", $champsAbonnement);
}

```

```

public function insertCommande($champs)
{
    $champsCommande = [ "id" => $champs["Id"], "dateCommande" => $champs["DateCommande"],
        "montant" => $champs["Montant"]];
    $champsCommandeDocument = [ "id" => $champs["Id"], "nbExemplaire" => $champs["NbExemplaire"],
        "idLivreDvd" => $champs["IdLivreDvd"], "idsuivi" => $champs["IdSuivi"]];
    $result = $this->insertOne("commande", $champsCommande);
    if ($result == null || $result == false){
        return null;
    }
    return $this->insertOne( "commandedocument", $champsCommandeDocument);
}

```

Procédure stockée pour ajouter des enregistrements dans la table exemplaire

Nom de la procédure:

Type: PROCEDURE ▾

	Direction	Nom	Type	Taille/Valeurs*	Options	
Paramètres	IN ▾	<input type="text" value="nbExemplaire"/>	INT ▾	<input type="text"/>	<input type="text"/>	Supprimer
	IN ▾	<input type="text" value="idDocument"/>	VARCHAR ▾	10	Jeu de caractères ▾	Supprimer
	IN ▾	<input type="text" value="dateAchatC"/>	DATE ▾	---	---	Supprimer

Ajouter un paramètre

Définition:

```

1 DECLARE nb INTEGER;
2 DECLARE maxId INTEGER;
3 SET nb = 0;
4 WHILE nb < nbExemplaire DO
5     SELECT MAX(numero) INTO maxId FROM exemplaire WHERE id = idDocument;
6     IF (maxId is null) THEN
7         SET maxId = 0;
8     END IF;
9     SET maxId = maxId + 1;
10    INSERT INTO exemplaire(id, numero, dateAchat, idEtat, photo)
11    VALUES (idDocument, maxId, dateAchatC, '00001', "");
12    SET nb = nb + 1;
13 END WHILE ;
14 END

```

Est déterministe: ☐

Ajuster les privilèges: ☒

Exécuter Fermer

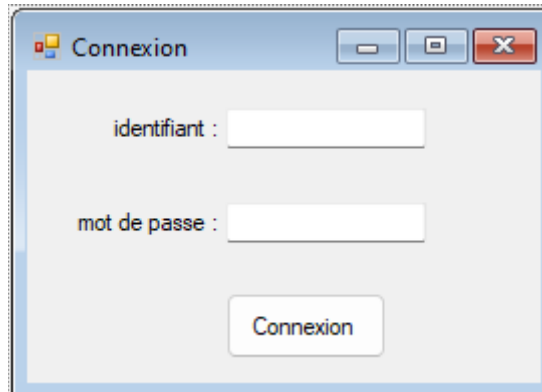
Créations des triggers pour la contrainte d'exclusion entre commandeDocument et abonnement.

Nom du déclencheur	<input type="text" value="InsAbonnement"/>
Table	<input type="text" value="abonnement"/>
Moment	<input type="text" value="BEFORE"/>
Évènement	<input type="text" value="INSERT"/>
Définition	<pre>1 BEGIN 2 DECLARE nb INTEGER ; 3 SELECT COUNT(*) INTO nb 4 FROM commandedocument 5 WHERE id = NEW.id ; 6 IF (nb = 1) THEN 7 SIGNAL SQLSTATE "45000" 8 SET MESSAGE_TEXT = "opération impossible"; 9 END IF ; 10 END</pre>

Nom du déclencheur	<input type="text" value="InsCommandeDocument"/>
Table	<input type="text" value="commandedocument"/>
Moment	<input type="text" value="BEFORE"/>
Évènement	<input type="text" value="INSERT"/>
Définition	<pre>1 BEGIN 2 DECLARE nb INTEGER ; 3 SELECT COUNT(*) INTO nb 4 FROM abonnement 5 WHERE id = NEW.id ; 6 IF (nb = 1) THEN 7 SIGNAL SQLSTATE "45000" 8 SET MESSAGE_TEXT = "opération impossible"; 9 END IF ; 10 END</pre>

Mission 4 : gérer les authentifications

Je crée rapidement la vue pour la connexion (possibilités d'identifiant : adresse mail ou son nom) :



Je crée les tables service et utilisateur et la classe utilisateur dans l'application C#

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
<input type="checkbox"/>	1 id	varchar(10)	utf8mb4_general_ci		Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/>	2 nom	varchar(20)	utf8mb4_general_ci		Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/>	3 prenom	varchar(20)	utf8mb4_general_ci		Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/>	4 mail	varchar(20)	utf8mb4_general_ci		Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/>	5 password	varchar(100)	utf8mb4_general_ci		Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/>	6 idservice	varchar(5)	utf8mb4_general_ci		Non	Aucun(e)			Modifier Supprimer Plus

☐ Tout cocher Avec la sélection : Parcourir Modifier Supprimer Primaire Unique Index S

Imprimer Déplacer des colonnes Normaliser

Ajouter colonne(s) après idservice ▼ Exécuter

Index

Action	Nom de l'index	Type	Unique	Compressé	Colonne	Cardinalité	Interclassement	Null	Commentaire
Éditer Renommer Supprimer	PRIMARY	BTREE	Oui	Non	id	5	A	Non	
Éditer Renommer Supprimer	idservice	BTREE	Non	Non	idservice	4	A	Non	

Structure de table Vue relationnelle

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
<input type="checkbox"/>	1 id	varchar(5)	utf8mb4_general_ci		Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/>	2 libelle	varchar(20)	utf8mb4_general_ci		Non	Aucun(e)			Modifier Supprimer Plus

☐ Tout cocher Avec la sélection : Parcourir Modifier Supprimer Primaire Unique Index Sp

Imprimer Déplacer des colonnes Normaliser

Ajouter 1 colonne(s) après libelle Exécuter

Index

Action	Nom de l'index	Type	Unique	Compressé	Colonne	Cardinalité	Interclassement	Null	Commentaire
Éditer Renommer Supprimer	PRIMARY	BTREE	Oui	Non	id	4	A	Non	

```
namespace MediaTekDocuments.model
{
    public class Utilisateur
    {
        /// <summary>D
        /// Valorise les propriétés
        /// </summary>
        /// <param name="id"></param>
        /// <param name="nom"></param>
        /// <param name="prenom"></param>
        /// <param name="mail"></param>
        /// <param name="idService"></param>
        /// <param name="service"></param>

        public Utilisateur(string id, string nom, string prenom, string mail, string idService, string service)
        {
            this.Id = id;
            this.Nom = nom;
            this.Prenom = prenom;
            this.Mail = mail;
            this.Service = new Service(idService, service);
        }

        public string Id { get; }

        public string Nom { get; set; }

        public string Prenom { get; set; }

        public string Mail { get; set; }

        public Service Service { get; }
    }
}
```

Et je modifie la classe Program pour qu'elle démarre sur FrmConnexion

```
Application.EnableVisualStyles();
Application.SetCompatibleTextRenderingDefault(false);
Application.Run(new FrmConnexion());
```

Import de System.Security.Cryptography pour les fonctions de hashage

```
public bool GetLogin(string mail, string password)
{
    password = "lalala" + password;
    string hash = "";
    using (SHA256 sha256Hash = SHA256.Create())
    {
        hash = GetHash(sha256Hash, password);
    }
    utilisateur = access.GetLogin(mail, hash);
    if (utilisateur != null)
    {
        Init();
        return true;
    }

    return false;
}
```

Pour générer un mot de passe et créer un utilisateur dans sa table, il nous suffit juste de hasher un mot de passe en clair en mettant bien le sel « lalala » devant et de le mettre dans le mdp de l'utilisateur sur phpmyadmin.

```
/// <summary>
/// Vérifie si les droits utilisateur
/// </summary>
/// <param name="lutilisateur"></param>
1 référence | didlyn9, Il y a 21 heures | 1 auteur, 1 modification
private void VerifDroitAccueil(Utilisateur lutilisateur)
{
    if (!controller.VerifDroitAccueil(lutilisateur))
    {
        MessageBox.Show("Vous n'avez pas les droits");
        Application.Exit();
    }
}
```

dans AccessBDD.php

```
public function selectUtilisateur($champs)
{
    $param = array(
        "mail" => $champs["mail"],
        "password" => $champs["password"]
    );
    $req = "Select u.id, u.nom, u.prenom, u.mail, u.idservice, s.libelle as service ";
    $req .= "from utilisateur u join service s on u.idservice=s.id ";
    $req .= "where u.mail = :mail ";
    $req .= "and u.password = :password ";
    $req .= "or u.nom = :mail ";
    $req .= "and u.password = :password";
    return $this->conn->query($req, $param);
}
```

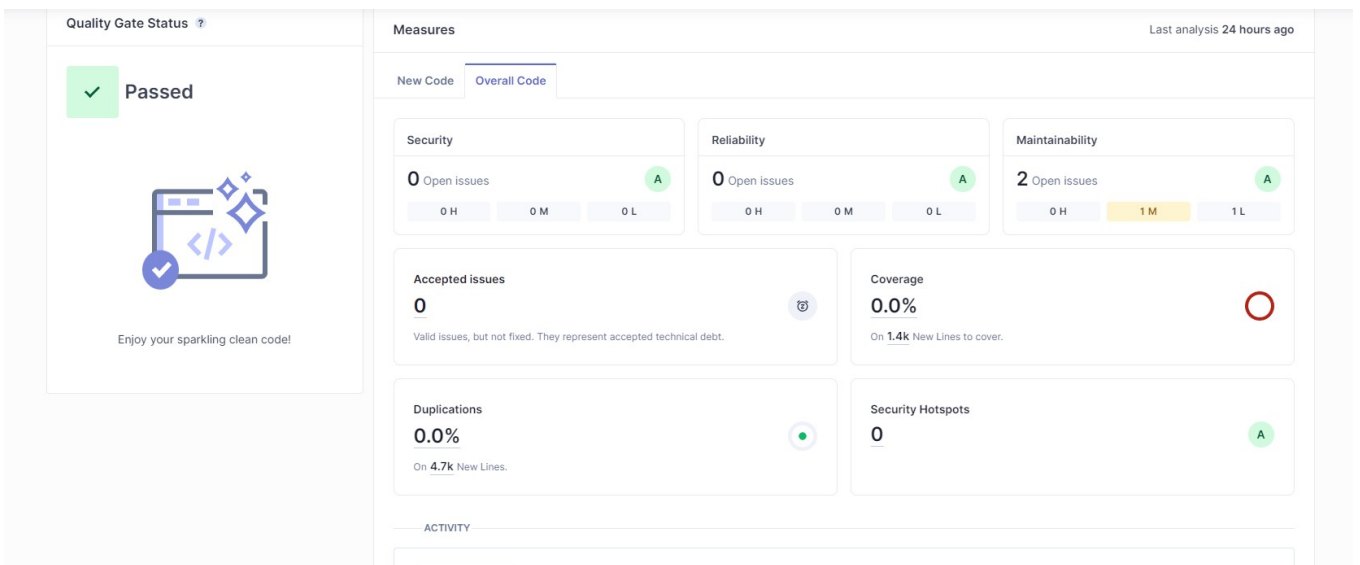
On a plus qu'à implémenter dans les entrées des tabs (ou de l'application pour le service culture)

```
private void TabLivresCom_Enter(object sender, EventArgs e)
{
    if (!controller.VerifCommande(utilisateur))
    {
        MessageBox.Show("Vous n'avez pas les droits.");
        tabOngletsApplication.SelectedIndex = 0;
    }
    else
    {
        tabCommandesLivres.AutoScroll = false;
        tabCommandesLivres.HorizontalScroll.Enabled = false;
        tabCommandesLivres.HorizontalScroll.Visible = false;
        tabCommandesLivres.HorizontalScroll.Maximum = 0;
        tabCommandesLivres.AutoScroll = true;
        leslivresCom = controller.GetAllLivres();
        RemplirComboCategorie(controller.GetAllGenres(), bdgGenres, cbxLivresComGenres);
        RemplirComboCategorie(controller.GetAllPublics(), bdgPublics, cbxLivresComPublics);
        RemplirComboCategorie(controller.GetAllRayons(), bdgRayons, cbxLivresComRayons);
        RemplirComboSuivi(controller.GetAllSuivis(), bdgLivresComEtat, cbxLivresComEtat);
        EnCoursModifLivresCom(false);
        RemplirLivresComListeComplete();
    }
}
```

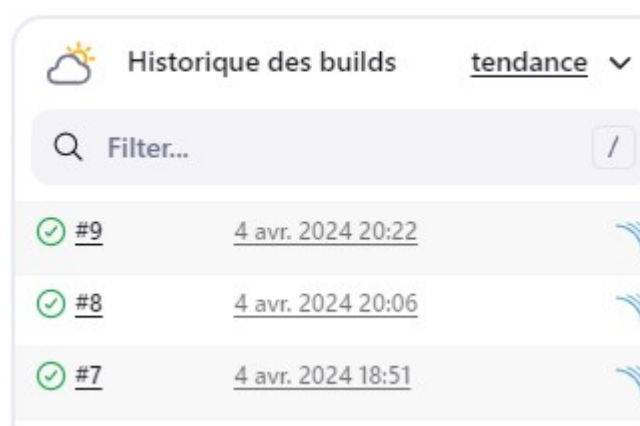
Mission 5 : Assurer la sécurité, la qualité et intégrer des logs

Après installation et configuration de SonarQube, on scan :

```
C:\WINDOWS\system32\cmd. X + v
ALL-UNNAMED --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java.io=ALL-UNNAMED --add-opens=java.rmi/s
un.rmi.transport=ALL-UNNAMED --add-exports=java.base/jdk.internal.ref=ALL-UNNAMED --add-opens=java.base/java.nio=ALL-UNN
AMED --add-opens=java.base/sun.nio.ch=ALL-UNNAMED --add-opens=java.management/sun.management=ALL-UNNAMED --add-opens=jdk
.management/com.sun.management.internal=ALL-UNNAMED -Dcom.redhat.fips=false -Xmx512m -Xms128m -XX:+HeapDumpOnOutOfMemory
Error -Dhttp.nonProxyHosts=localhost|127.*|[::1] -cp ./lib/sonar-application-10.4.1.88267.jar;C:\Users\Saerine\Desktop\sonarqube-10.4.
1.88267\temp\sq-process17297534326188410428properties
WARNING: A terminally deprecated method in java.lang.System has been called
WARNING: System::setSecurityManager has been called by org.sonar.process.PluginSecurityManager (file:/C:/Users/Saerine/D
esktop/sonarqube-10.4.1.88267/lib/sonar-application-10.4.1.88267.jar)
WARNING: Please consider reporting this to the maintainers of org.sonar.process.PluginSecurityManager
WARNING: System::setSecurityManager will be removed in a future release
2024.04.05 14:12:04 INFO app[[o.s.a.SchedulerImpl] Process[web] is up
2024.04.05 14:12:04 INFO app[[o.s.a.ProcessLauncherImpl] Launch process[COMPUTE_ENGINE] from [C:\Users\Saerine\Desktop
\sonarqube-10.4.1.88267]: C:\Program Files\Java\jdk-17\bin\java -Djava.awt.headless=true -Dfile.encoding=UTF-8 -Djava.io
.tmpdir=C:\Users\Saerine\Desktop\sonarqube-10.4.1.88267\temp -XX:-OmitStackTraceInFastThrow --add-opens=java.base/java.u
til=ALL-UNNAMED --add-exports=java.base/jdk.internal.ref=ALL-UNNAMED --add-opens=java.base/java.lang=ALL-UNNAMED --add-o
pens=java.base/java.nio=ALL-UNNAMED --add-opens=java.base/sun.nio.ch=ALL-UNNAMED --add-opens=java.management/sun.managem
ent=ALL-UNNAMED --add-opens=jdk.management/com.sun.management.internal=ALL-UNNAMED -Dcom.redhat.fips=false -Xmx512m -Xms
128m -XX:+HeapDumpOnOutOfMemoryError -Dhttp.nonProxyHosts=localhost|127.*|[::1] -cp ./lib/sonar-application-10.4.1.88267
.jar;C:\Users\Saerine\Desktop\sonarqube-10.4.1.88267\lib\jdbc\h2\h2-2.2.224.jar org.sonar.ce.app.CeServer C:\Users\Saeri
ne\Desktop\sonarqube-10.4.1.88267\temp\sq-process12473405180795534522properties
WARNING: A terminally deprecated method in java.lang.System has been called
WARNING: System::setSecurityManager has been called by org.sonar.process.PluginSecurityManager (file:/C:/Users/Saerine/D
esktop/sonarqube-10.4.1.88267/lib/sonar-application-10.4.1.88267.jar)
WARNING: Please consider reporting this to the maintainers of org.sonar.process.PluginSecurityManager
WARNING: System::setSecurityManager will be removed in a future release
2024.04.05 14:12:06 INFO app[[o.s.a.SchedulerImpl] Process[ce] is up
2024.04.05 14:12:06 INFO app[[o.s.a.SchedulerImpl] SonarQube is operational
```



Ensuite, on relie le projet à Jenkins :



On ouvre les ports sur mafreebox.freebox.fr

Redirection de port

IP Destination :

192.168.0.13

?

Redirection active :

☒

IP source :

Toutes

?

Protocole :

TCP

?

Port de début :

8080

?

Port de fin :

8080

?

Port de destination :

8080

?

Commentaire :

Annuler

Sauvegarder

On configure le webhook github après avoir obtenu notre API secret dans Jenkins :

We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in [our developer documentation](#).

Payload URL *

Content type

Secret

If you've lost or forgotten this secret, you can change it, but be aware that any integrations using this secret will need to be updated. — [Change Secret](#)

Which events would you like to trigger this webhook?

☒ Just the push event.

☐ Send me everything.

☐ Let me select individual events.

☒ **Active**

We will deliver event details when this hook is triggered.

On vérifie qu'il n'y ait pas d'erreur :

✓ <http://82.64.41.244:8080/github-we...> (push)


Le projet est bien relié à SonarQube via Jenkins et les builds seront lancé à chaque push.

Note : il faut télécharger nuget.exe (en x86) et ajouter une commande batch nuget restore dans l'étape du build pour que la génération se fasse.

<div> Historique des builds</div> <div>tendance </div>		
<div> Filter... </div>		
 <u>#9</u>	<u>4 avr. 2024 20:22</u>	
 <u>#8</u>	<u>4 avr. 2024 20:06</u>	
 <u>#7</u>	<u>4 avr. 2024 18:51</u>	

Mission 6 : tester et documenter

Tests unitaires :

Explorateur de tests			
			
Test	Durée	Caractéris...	Message d'erreu
MediaTekDocuments (7)	5 ms		
MediaTekDocuments.Tests (7)	5 ms		
AbonnementTests (1)	5 ms		
CommandeDocumentTests (1)	< 1 ms		
CommandeTests (1)	< 1 ms		
ExemplaireTests (1)	< 1 ms		
SuiviTests (2)	< 1 ms		
UtilisateurTests (1)	< 1 ms		

Tests postman :

GET http://localhost/rest_med GET http://localhost/rest_med + ...

http://localhost/rest_mediatekdocuments/utilisateur

GET http://localhost/rest_mediatekdocuments/utilisateur

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Query Params

Key	Value
Key	Value

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON

```
1  {
2    "code": 200,
3    "message": "OK",
4    "result": [
5      {
6        "id": "0001",
7        "nom": "Albert",
8        "prenom": "Desrue",
9        "mail": "albert@desrue.fr",
10       "password": "5cef3e7970ec491fd718df7046f121ed6bf80498717006589638b857125d8547",
11       "idservice": "0001"
12     },
13     {
14       "id": "0002",
15       "nom": "Eric",
16       "prenom": "Duciel",
17       "mail": "eric@duciel.fr",
18       "password": "5cef3e7970ec491fd718df7046f121ed6bf80498717006589638b857125d8547",
19       "idservice": "0002"
20     },
21     {
22       "id": "0003"
```

GET

http://localhost/rest_mediatekdocuments/genre

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings

Query Params

	Key
	Key

Body

Cookies

Headers (7)

Test Results

Pretty

Raw

Preview

Visualize

JSON

```
1  {
2    "code": 200,
3    "message": "OK",
4    "result": [
5      {
6        "id": "10018",
7        "libelle": "Actualités"
8      },
9      {
10       "id": "10007",
11       "libelle": "Aventures"
12     },
13     {
14       "id": "10001",
15       "libelle": "Bande dessinée"
16     },
17     {
18       "id": "10003",
19       "libelle": "Biographie"
20     },
21     {
22       "id": "10013"
```

Ayant eu des problèmes avec SpecFlow, je n'ai pas pu faire de tests fonctionnels.

La documentation technique de l'application C# a été faite avec Sandcastle

La documentation technique de l'API a été faite avec phpDocumentor.

Mission 7 : déployer et gérer les sauvegardes de données

Je crée une VM azure sous Ubuntu 20.04 qui écoute sur les ports 80(HTTP) 443(HTTPS) et 22 (pour le SSH)

J'installe tout ce dont j'ai besoin

```
>sudo apt update
```

```
>sudo apt upgrade
```

```
>sudo apt install apache2
```

```
>sudo apt install mysql-server
```

```
>sudo mysql
```

```
ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'turkiye362';
```

```
>exit
```

```
>sudo mysql_secure_installation
```

```
>sudo apt install php libapache2-mod-php php-mysql
```

```
>sudo apt install phpmyadmin
```

Je configure apache pour qu'il soit « restful » :

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html

    # Désactiver le listing des répertoires
    Options -Indexes

    # Activer la prise en charge de .htaccess en définissant AllowOverride
    <Directory "/var/www/html">
        AllowOverride All
        Require all granted
    </Directory>

    # Spécifier le fichier index par défaut
    DirectoryIndex index.php

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

et

```
>sudo nano /etc/apache2/apache2.conf
```

J'ajoute la ligne

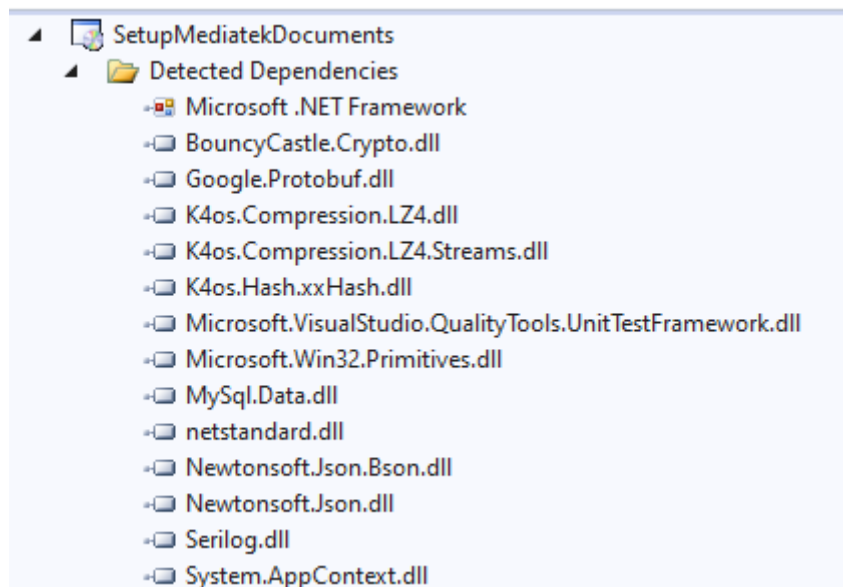
Include /etc/phpmyadmin/apache.conf (pour activer phpmyadmin après l'installation)

Pour le déploiement, il a fallu changer le fichier .htaccess

```
RewriteEngine on
RewriteRule ^$ mediatekdocuments.php?error=404
RewriteRule ^([a-zA-Z0-9_+)]$ mediatekdocuments.php?table=$1
#RewriteRule ^([a-zA-Z0-9_+)]/([s\S]*)$ mediatekdocuments.php?table=$1&champs=$2
RewriteRule ^([a-zA-Z0-9_+)]/([.]*))$ mediatekdocuments.php?table=$1&champs=$2 [B,L]
RewriteRule ^([a-zA-Z0-9_+)]/([a-zA-Z0-9_+)]/([.]*))$ mediatekdocuments.php?table=$1&id=$2&champs=$3 [B,L]
RewriteCond %{HTTP:Authorization} .
RewriteRule .* - [E=HTTP_AUTHORIZATION:%{HTTP:Authorization}]
```

Ainsi que la configuration de la bdd dans le fichier .env

J'importe la bdd, je change l'uri dans l'application C# et c'est terminé, j'ai plus qu'à générer le setup de l'application avec l'extension Microsoft Visual Studio Installer Projects



Pour finir, je configure le script de sauvegarde de la base de donnée dans le dossier /usr/backup/

backup_db.sh

```
#!/bin/bash

# Définir le nom de la base de données, l'utilisateur et le mot de passe
DB_NAME="mediatek86"
DB_USER="root"
DB_PASS="turkiye362"

# Définir le répertoire de sauvegarde
BACKUP_DIR="/usr/backup/"

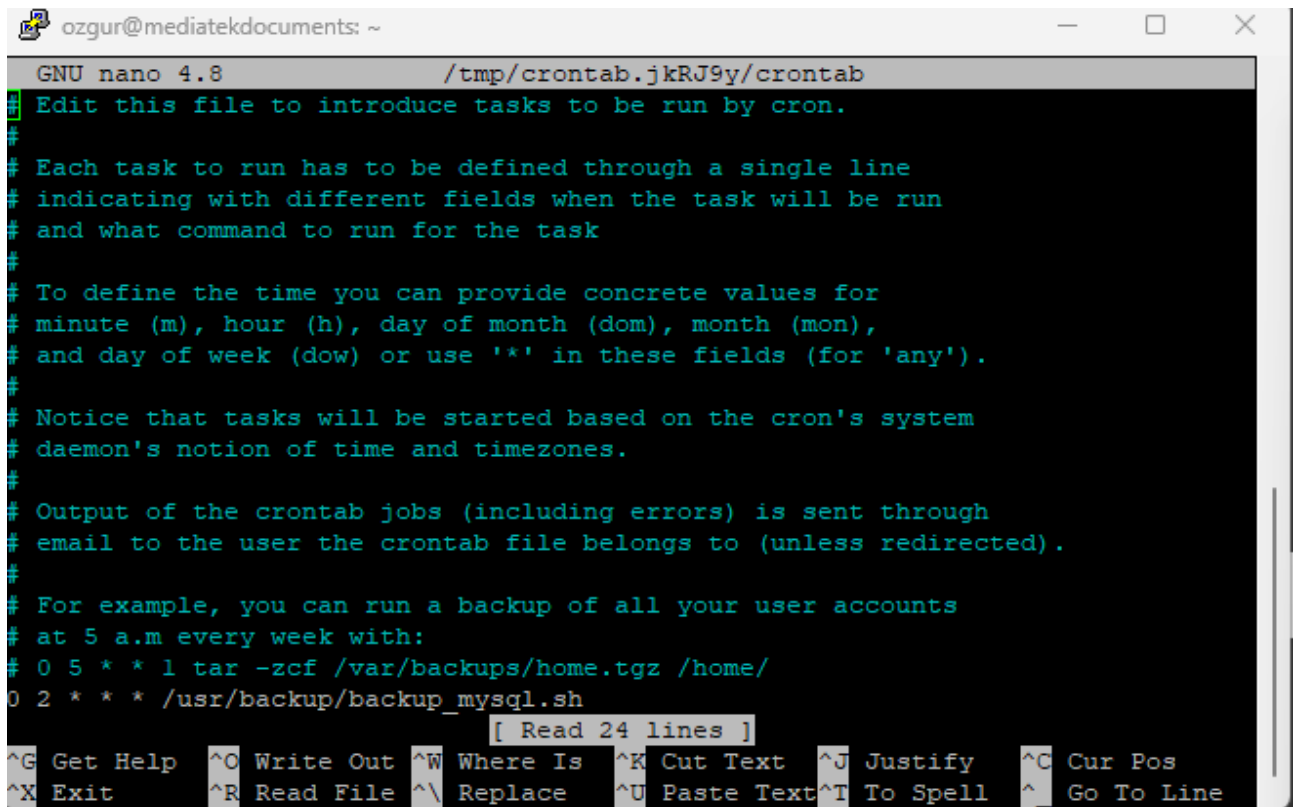
# Définir le format de date pour le nom du fichier
DATE=$(date +"%Y-%m-%d_%H-%M-%S")

# Chemin complet du fichier de sauvegarde
BACKUP_PATH="${BACKUP_DIR}/${DB_NAME}_${DATE}.sql.gz"

# Commande de dump
mysqldump -u ${DB_USER} -p${DB_PASS} ${DB_NAME} | gzip > ${BACKUP_PATH}

echo "La sauvegarde de la base de données '${DB_NAME}' a été créée: ${BACKUP_PATH}"
```

Après l'avoir converti avec dos2unix, je peux ouvrir et faire la tâche cron de l'application



```
GNU nano 4.8 /tmp/crontab.jkRJ9y/crontab
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
0 2 * * * /usr/backup/backup_mysql.sh
[ Read 24 lines ]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^_ Go To Line
```

C'est ainsi que s'achève ce deuxième atelier.