

HMIN309 - Développement logiciel pour l'embarqué

TP 1 - Introduction à la plateforme Xamarin

Pré-requis

Téléchargez les sources du TP1 depuis la page Moodle du cours. Dans Visual Studio, au démarrage, cliqué sur Ouvrir et ouvrez le fichier **TP1-ProjetPartage.sln**.

Une fois chargé, vous devriez être face à une solution .NET comprenant 3 sous projets :

- **TP1-ProjetPartage** : les sources du projet partagé
- **TP1-ProjetPartage.iOS** : les sources de l'application iOS
- **TP1-ProjetPartage.Droid** : les sources de l'application Android

Dans ce 1er TP vous allez réaliser une application cross-plateforme pour visualiser en directe les horaires de TRAM de la ville de Montpellier.

Mise en Place

Naviguez avec URL suivante vers le site Open Data du Gouvernement Français :

<https://www.data.gouv.fr/fr/datasets/horaires-en-temps-reel-du-reseau-de-transport-tam/>

Pour simplifier, un fichier CSV est disponible sur l'ENT. Téléchargez le et ouvrez le.

Vous vous trouvez maintenant face au fichier CSV qui contient toutes les 3 prochaines horaires de passages pour chaque arrêts de TRAM de Montpellier.

```
course;stop_code;stop_id;stop_name;route_short_name;trip_headsign;direction_id;departure_time;is_theoretical;delay_sec;dest_ar_code
268436158;ANTIGRTW;41217;ANTIGONE;1;MOSSON;1;10:16:13;0;101;41101
268436056;ANTIGRTW;41217;ANTIGONE;1;MOSSON;1;10:18:02;0;210;41101
268436119;ANTIGRTW;41217;ANTIGONE;1;MOSSON;1;10:22:35;0;483;41101
```

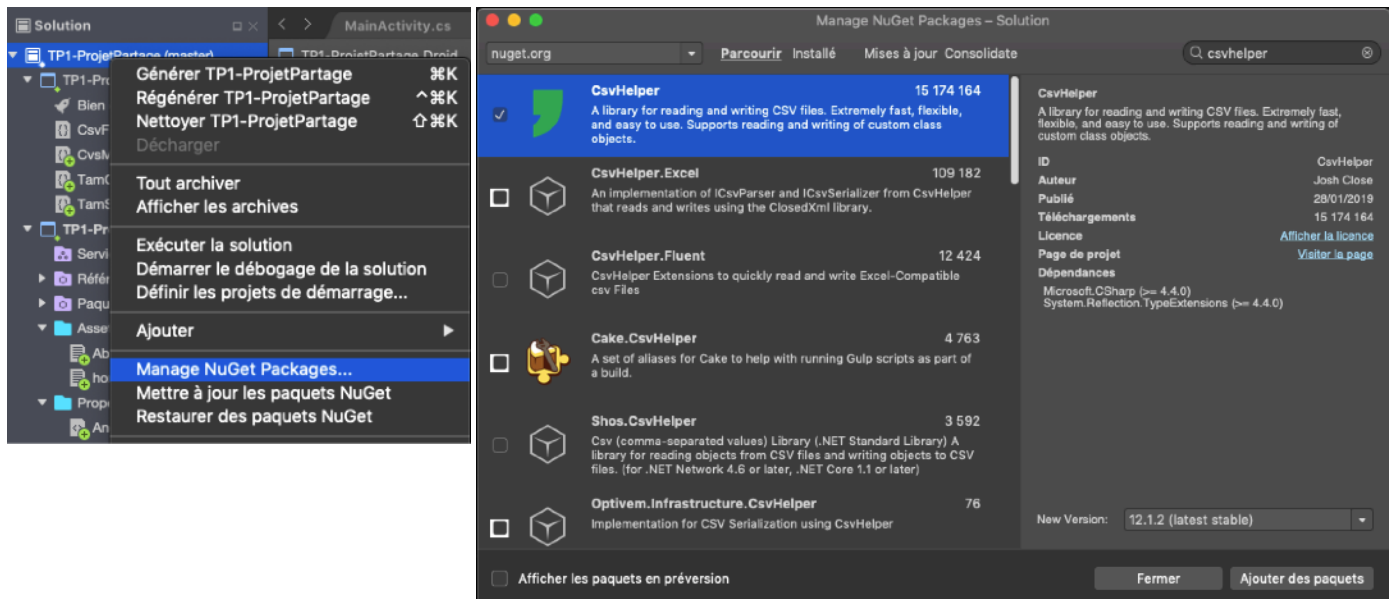
Le fichier CSV déclare les attributs suivants :

- **course** (ex 268436087): identifiant unique de la course.
- **stop_code** (ex BOUTORTW): nom de code pour d'un arrêt de Tram de la 'TAM'
- **stop_id** (ex 41233): identifiant unique du terminus de la course
- **stop_name** (ex BOUTONNET): nom complet de l'arrêt de Tram concerné
- **route_short_name** (ex 1): id de la route prise par le Tram
- **trip_headsign** (ex MOSSON): nom complet de la direction de la course
- **direction_id** (ex 1):
- **departure_time** (ex 10:24:36): Heure de départ du tram
- **is_theoretical** (ex 0): bool, indique si l'heure est théorique
- **delay_sec** (ex 604): en seconde, indique le retard éventuel
- **dest_ar_code** (ex 41101): Identifiant du terminus

Placez le fichier CSV **horairesTram.csv** dans le dossier Assets du projet Android et Resources pour le projet iOS.

Exercice 1

1. Nous allons ajouter une classe pour lire et instancier les lignes du fichier CSV dans des objets. Créer dans le projet Partagé **TP1-ProjetPartage** une nouvelle classe **TamCSVTPsReel**.
2. Dans cette classe, créez les attributs correspondants aux attributs du présent dans le fichier CSV. Attention, il doivent avoir exactement le même nom et la même casse !
3. Créez une autre classe, **CSVManager** qui se chargera en mémoire le contenu d'un fichier CSV pour créer un tableau de d'objets **TamCSVTPsReel**. Pour cela, utilisez **NuGet Manager** pour télécharger le plugin **CSVHelper**.



4. Dans CSVManager, créez une méthode **ReadCSV** qui utilisera **CSVHelper** pour charger le texte du fichier CSV dans votre application (documentation CSVHelper <https://joshclose.github.io/CsvHelper/getting-started>). Attention nous avons vu dans le cours que les accès aux ressources étaient spécifique aux plateformes ! Vous devez toute de même, en une seule méthode, charger à la fois le CSV pour iOS et Android. Comment faire ?
5. Afficher le contenu du fichier CSV dans la console. La encore, vous devez faire en sorte que cela fonctionne pour les deux plateformes.

documentations :

- **IEnumerator**
- **Ajouter Asset dans Android**
- **CVSHelper**
- **Projet Partagé**

Exercice 2

6. Créez une autre classe **TamSchedule** qui contiendra les attributs :
 - a. `_tramStrop` : le nom de l'arrêt de tram concerné
 - b. `_tramDestination[3]` : le nom du terminus des 3 prochaines course
 - c. `_tramHoraires[3]` : un tableau qui contiendra les 3 prochaines horaires de passage.

7. Créez une classe **TamScheduleManager**. Cette dernière contiendra les méthodes pour transformer notre tableau de **TamCSVTPsReel** en objets **TamSchedule**. Créez une méthode **ExtractTamSchedule** qui renvoie la liste des objets **TamSchedule** à partir du tableau **TamCSVTPsReel**.
8. Créez une méthode **ToString** dans **TamSchedule** pour créer un affichage similaire à celui des panneaux d'affichage du Tram de Montpellier (en ASCII)
9. Faites un affichage dans la console.



Exercice 3

Faites la conception d'une interface graphique simple pour iOS et Android où les infos de l'objet **TamSchedule** s'affichent dans une liste sur la page principal de l'application. Utilisez pour cela une **ListView** avec les informations des objets **TamSchedule**.

Créer pour cela une fonction dans **TamScheduleManager** qui renvoie un **ArrayList** des **TamSchedule** en utilisant la fonction **ToString**.

Résultat possible :



Documentations :

- <https://docs.microsoft.com/en-us/xamarin/ios/>
- <https://docs.microsoft.com/en-us/xamarin/android/>