

Introduction à la sérialisation

- Sérialisation, qu'est-ce que sait?
- JsonSerializer
- Exemple de sérialisation
- Exemple de désérialisation

Sérialisation, qu'est-ce que sait?

- La sérialisation est le principe de convertir les données d'une variables dans un format texte.
- En C#, il est possible de **sérialiser** des données de pratiquement toutes les types (string, int, char, double, bool, array[], list<>, etc.)
- Cela permet de sauvegarder et charger facilement des données dans un fichier texte.

Évidemment, sérialiser une `string` n'est pas super pertinent, mais c'est le même principe, peu importe le type, et vous serez définitivement heureux de pouvoir le faire quand viendra le temps de l'appliquer sur une liste.

JsonSerializer

- Nous utiliserons la classe `JsonSerializer` pour sérialiser et désérialiser nos données.
- Cette classe utilise le format **JSON** (JavaScript Object Notation).
 - Un format de données léger, largement utilisé pour échanger des informations.
- En Json, les données sont organisées sous forme d'**objets** (`{}`) et de **tableaux** (`[]`).
 - *Vous apprendrez tout ce qu'il y a à savoir sur le JSON la session prochaine.*

System.Text.Json.JsonSerializer

La classe JsonSerializer fait partie de la bibliothèque `System.Text.Json`.

1. **Ajouter le using (*implicite import*)** : `using System.Text.Json;` si celui-ci n'est pas importé implicitement.
2. **Méthodes** : `JsonSerializer.Serialize()` et `JsonSerializer.Deserialize<TYPE>()` pour écrire et lire les données JSON.
3. **Sauvegarder/charger le JSON**: dans un fichier texte avec la classe `FILE`

Exemple de sérialisation

```
int[] scores = [85, 90, 78];  
List<string> amis = ["Bob", "Charlie"];  
  
string strScores = JsonSerializer.Serialize(scores);  
string strAmis = JsonSerializer.Serialize(scores);
```

Exemple de désérialisation

Important! Vous devez spécifier le type de donnée dans la désérialisation.

```
int[] scoresCopy = JsonSerializer.Deserialize<int[]>(strScores);  
List<string> amisCopy = JsonSerializer.Deserialize<List<string>>(strAmis);
```