



L'objectif de ce TD est de tester et valider votre intuition de détection de comportements. Vous allez donc récupérer et traiter les données en langage C#. Bien-entendu, le pré-requis est d'avoir terminé le TD2 et ainsi de posséder sa base de référence.

└ Exercice 1 : Lire un JSON ▯

Nous allons utiliser l'API fournie pour récupérer les informations :

1. Créer un nouveau projet Console
2. Nous allons utiliser une librairie permettant de manipuler des données JSON : la librairie Json.NET de Newtonsoft. Ajouter le package "Newtonsoft.json" via l'outil NuGet :
<https://docs.nuget.org/consume/package-manager-dialog>.
Qu'est-ce qu'un package ? Quel est le rôle de NuGet ?
3. Afin d'importer la librairie dans votre projet : Ajouter *using Newtonsoft.Json* ;
4. Tester l'exemple "jsonDeserialize" qui deserialize une collection json.
5. Expliquer la ligne :

```
List<Account> account = JsonConvert.DeserializeObject<List<Account>>(json);
```

6. Supprimer le champ *Active* de la classe Account. Cela fonctionne-t-il toujours ?
7. Utiliser un tableau de string plutôt qu'une liste pour la donnée Roles
8. Associer une donnée de type tableau de tableau de double (double [][] ou double [,])

```
{ // 1er élément  
'Email': 'james@example.com',  
'Active': true,  
'CreateDate': '2013-01-20T00:00:00Z',  
'Roles': ['User', 'Admin'],  
'Data' : [[2, 0.25], [51.8, 5.5]]  
}
```

└ Exercice 2 : Récupérer la donnée sur le web ▯

Maintenant que vous savez convertir une collection JSON en liste il faut récupérer les données au format json depuis le serveur. Pour cela, nous allons simplement utiliser le service web fourni.

1. Utilisez la directive using à l'espace de noms System.IO et l'espace de noms System.NET :
`using System.Net;`
`using System.IO;`
2. Créer un objet WebClient dont le rôle sera d'exécuter la requête web et de retourner le résultat obtenu :

```
WebClient webClient = new WebClient();

webClient.QueryString.Add("username", "");
webClient.QueryString.Add("password", "");
webClient.QueryString.Add("database", "");
webClient.QueryString.Add("collection", "");
webClient.QueryString.Add("action", "");

string jsonString = webClient.DownloadString("http://ibo.labs.esilv.fr/~webservice/api/data");

Console.WriteLine(jsonString);
```

Exercice 3 : Récupérer les données et les “deserializer” dans un objet

Vous savez convertir une collection en liste d'objets et vous savez récupérer les données en format json depuis le serveur : vous pouvez maintenant récupérer vos données et les stocker dans un objet.

1. Proposer une classe permettant de stocker les informations. On ne stockera pas l'information `_id`.
2. Afficher les données d'une série de mesures

Exercice 4 : Reconnaître les mouvements

1. Écrire l'algorithme, qui, à partir d'une série de mesures, prédit le type d'action associée. Un algorithme ne contient pas de code mais doit décrire la logique de votre raisonnement permettant de détecter l'action associée.
2. Implémenter cet algorithme.
3. Proposer une méthodologie pour mesurer l'efficacité de votre algorithme.