Commonly a JSON is made up of two structures, the first is the object that is represented by curly braces "{ }" and contains key and value. as shown in the following example:

{ "key": "value" }

The second structure is the array and this is represented by brackets "[ ]", these brackets can have a list of values ​​for example:

[ "value\_1", "value\_2", "value\_3" ]

A JSON can be made up of nested objects and arrays, this allows complex data to be represented where each key is a string between quotes, the data type of the values ​​can be strings, numbers, booleans, objects, arrays or null, as shown in the following example representing a collection of books along with their category:

{

"book":

[

{

"id": 1,

"title": "Title1"

},

{

"id": 2,

"title": "Title2"

}

],

"Category" : "Science fiction"

}

Let's see a simple example of how to decode a JSON using the System.Text.Json library, the Json to decode would be the following:

[

{

"id": 1,

"title": "Title1”

},

{

"id": 2,

"title": "Title2"

}

]

The first step would be to create our data model. Defining the model is important because it represents the structure of the JSON once it is decoded.

public class Book

{

public int Id { get; set; }

public string Title { set; get; }

}

Having our model, the next step would be to decode this JSON format using the JsonSerializer.Deserialize<T>( ) method.

string **jsonString** =

"[{ \"id\": 1, \"Title\": \"Title1\" }, { \"id\": 2, \"Title\": \"Title2\" }]";

List<Book> **books** = JsonSerializer.Deserialize<List<Book>>(**jsonString**);

foreach(var item in **books**)

{

Console.WriteLine($"Id: {item.Id}");

Console.WriteLine($"Title: {item.Title}");

}