

Exercice : ITI.JsonParser

1 Enoncé

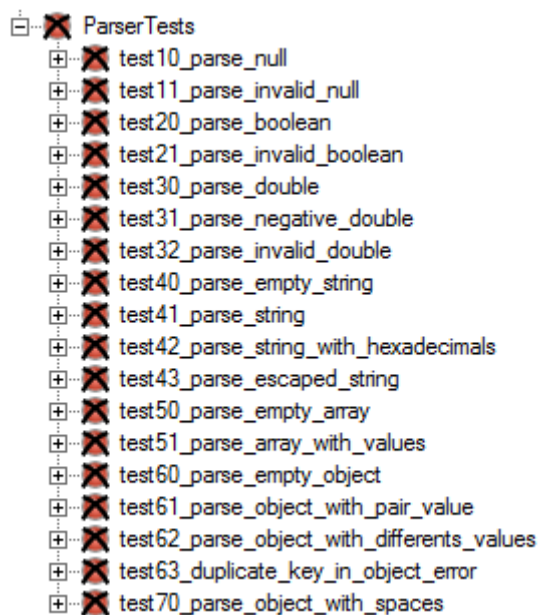
L'objectif de cet exercice est d'implémenter un parseur JSON.

Vous disposez pour cela d'une solution comprenant deux projets :

- ITI.JsonParser, est un squelette d'API à implémenter
- ITI.JsonParser.Tests, contient les tests unitaires

Pour faire tourner les tests il vous suffit de configurer le projet ITI.JsonParser.Tests en tant que projet de démarrage puis d'exécuter la solution.

Comme vous pouvez le constater, pour le moment, tous les tests sont rouges :



A vous de faire en sorte qu'ils passent en vert. Pour cela, vous avez le droit de faire ce que bon vous semble dans le projet ITI.JsonParser.

Les tests unitaires sont là pour spécifier de façon détaillée les fonctionnalités attendues. Cependant le présent document va décrire brièvement ce qui est attendu.

2 ITI.JsonParser

Le projet ITI.JsonParser contient la classe Parser dont l'API est la suivante :

```
public static class Parser
{
    public static object ParseNull(string value, ref int start, ref int count) {...}
    public static bool ParseBoolean(string value, ref int start, ref int count) {...}
    public static double ParseDouble( string value, ref int start, ref int count ) {...}
    public static string ParseString( string value, ref int start, ref int count ) {...}
    public static object[] ParseArray( string value, ref int start, ref int count ) {...}
    public static Dictionary<string, object> ParseObject( string value, ref int start, ref int count ) {...}
}
```

Il est recommandé d'implémenter dans un premier temps la méthode [ParseNull](#), ensuite [ParseBoolean](#), puis [ParseDouble](#), puis [ParseString](#), puis [ParseArray](#) et enfin [ParseObject](#). Les tests sont classés suivant cet ordre d'implémentation qui, est de difficulté croissante, et les méthodes implémentées plus tôt pourront être réutilisées plus tard.

Vous avez toujours la possibilité de suivre l'ordre qui vous convient, l'essentiel est de faire passer tous les tests au vert.

3 Barème

Chaque test rapporte un point. Sachant qu'il y a 18 tests à passer cela fait 18 points. 2 points supplémentaires seront accordés à la qualité du code. Au total, 20 points sont à acquérir.