RICHARD Arthur 2TSIO

Tests unitaires

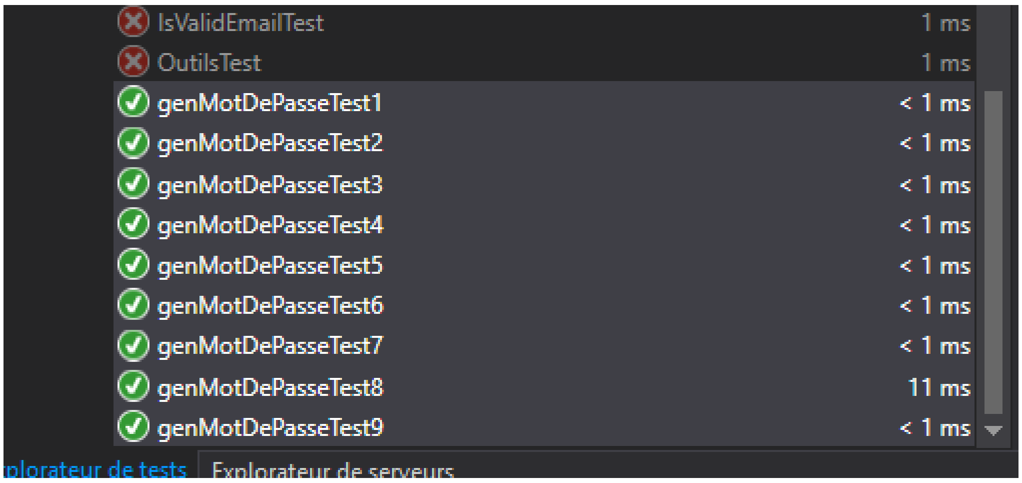
Fonction genMotDePasse dans la classe Outils :

```
public static string genMotDePasse(string nom, string prenom)
{
    string mdp;
    // Nom composé
    nom = nom.Replace("De la", "la");
    if (nom.Length>6)
    {
        mdp = nom.Substring(0,6) + " " + prenom.Substring(0, 1);
    }
    else
    {
        mdp = nom + " " + prenom.Substring(0, 1);
    }


    // Caractères spéciaux
    mdp = mdp.Replace(" ", "_");
    mdp = mdp.Replace("-", "_");
    mdp = mdp.Replace("__","_");
    mdp = mdp.Replace("{","");
    mdp = mdp.Replace("'", "_");
    mdp = mdp.Replace("@", "");
    mdp = mdp.Replace("}", "");
    mdp = mdp.Replace("%", "");
    mdp = mdp.Replace("#", "");
    mdp = mdp.Replace("!", "");
    mdp = mdp.Replace("?", "");
    mdp = mdp.Replace("/", "");
    mdp = mdp.Replace(")", "");
    mdp = mdp.Replace("(", "");
    mdp = mdp.Replace("[", "");
    mdp = mdp.Replace("]", "");

    // Accents
    mdp = mdp.Replace("é", "e");
    mdp = mdp.Replace('è', 'e');
    mdp = mdp.Replace('ê', 'e');
    mdp = mdp.Replace('ë', 'e');
    mdp = mdp.Replace('à', 'a');
    mdp = mdp.Replace('ù', 'u');


    mdp = mdp.ToLower();

    if (mdp.Length>8 || mdp.Length<3)
    {
        mdp = "error";
    }
    return mdp;
}
```

Tests unitaire :

```
[TestMethod()]
        public void genMotDePasseTest1()
        {
            string nom = "Dupont";
            string prenom = "paul";
            string output = "dupont_p";
            Assert.AreEqual(Outils.genMotDePasse(nom, prenom), output);
        }
        [TestMethod()]
        public void genMotDePasseTest2()
        {
            string nom = "paul";
            string prenom = "pierre";
            string output = "paul_p";
            Assert.AreEqual(Outils.genMotDePasse(nom, prenom), output);
        }
        [TestMethod()]
        public void genMotDePasseTest3()
        {
            string nom = "Ledu";
            string prenom = "germain";
            string output = "ledu_g";
            Assert.AreEqual(Outils.genMotDePasse(nom, prenom), output);
        }
        [TestMethod()]
        public void genMotDePasseTest4()
        {
            string nom = "Le Du";
            string prenom = "germain";
            string output = "le_du_g";
            Assert.AreEqual(Outils.genMotDePasse(nom, prenom), output);
        }
        [TestMethod()]
        public void genMotDePasseTest5()
        {
            string nom = "Le du bas";
            string prenom = "Germain";
            string output = "le_du_g";
            Assert.AreEqual(Outils.genMotDePasse(nom, prenom), output);
        }
        [TestMethod()]
        public void genMotDePasseTest6()
        {
            string nom = "De la Roche";
            string prenom = "Edith";
            string output = "la_roc_e";
            Assert.AreEqual(Outils.genMotDePasse(nom, prenom), output);
        }
        [TestMethod()]
        public void genMotDePasseTest7()
        {
            string nom = "Roche-Martin";
            string prenom = "élodie";
            string output = "roche_e";
            Assert.AreEqual(Outils.genMotDePasse(nom, prenom), output);
        }
        [TestMethod()]
        public void genMotDePasseTest8()
        {
            string nom = "Roc-Martin";
            string prenom = "élise";
            string output = "roc_ma_e";
            Assert.AreEqual(Outils.genMotDePasse(nom, prenom), output);
        }
        [TestMethod()]
        public void genMotDePasseTest9()
        {
            string nom = "Lécuyer";
            string prenom = "Antoine";
            string output = "lecuye_a";
            Assert.AreEqual(Outils.genMotDePasse(nom, prenom), output);
        }
```

Jeux d'essais :



```
IsValidEmailTest          1 ms
OutilsTest                1 ms
genMotDePasseTest1      < 1 ms
genMotDePasseTest2      < 1 ms
genMotDePasseTest3      < 1 ms
genMotDePasseTest4      < 1 ms
genMotDePasseTest5      < 1 ms
genMotDePasseTest6      < 1 ms
genMotDePasseTest7      < 1 ms
genMotDePasseTest8       11 ms
genMotDePasseTest9      < 1 ms
```