

BOURGEOIS Charles
StJoSup, Le Havre

SIO2

Compte-rendu
Sérialisation

2023

Sommaire

1) C'est quoi la sérialisation ?	p3
2) Les 3 types de sérialisation	p4
a) SOAP	p5
b) Binaire	p6
c) XML	p7
3) La définition des méthodes sous forme algorithmique	p8
4) Conclusion	p9

1) C'est quoi la sérialisation ?

La *sérialisation* est le codage d'une information sous la forme d'une suite d'informations réduites.

C'est un processus de conversion d'un objet en flux d'octets pour stocker l'objet ou le transmettre à la mémoire, une base de données ou un fichier. Grâce à la sérialisation, nous pourrions enregistrer l'état d'un objet afin de pouvoir le recréer dès que demandé.

L'inverse de la sérialisation s'appelle la *désérialisation*, elle correspond au fait de décoder cette suite pour créer une copie conforme de l'information d'origine,

La sérialisation a différents usages :

- Pouvoir transférer un objet entre différents domaines ou vers une application
- Créer une copie intégrale d'un objet
- Stocker des objets inutilisés

Elle peut donc être également utile dans le domaine de la sauvegarde et dans la portabilité des encodages.

La sérialisation de données au format texte, elle, concerne des fichiers dont le code binaire correspond uniquement à des caractères textuels.

2) Les 3 types de sérialisation

a) Sérialisation SOAP

SOAP est un protocole simplifié permettant l'échange d'informations dans un environnement décentralisé et réparti.

Il est basé sur XML, qui définit trois parties :

- *Enveloppe*
- *Règles de codage*
- *Styles de communication*

Il est le plus souvent transmis via un protocole http, permettant d'adapter SOAP à la compatibilité avec d'autres infrastructures internet existante.

SOAP est conçu spécifiquement pour transporter des appels de procédure à l'aide de XML.

Lorsqu'un client envoie une requête à un service Web SOAP, les données sont converties en un format XML spécifique, qui est ensuite envoyé au service Web. Le service Web utilise ensuite ce format pour désérialiser les données et effectuer les opérations demandées. De même, lorsque le service Web envoie une réponse au client, les données sont sérialisées en XML avant d'être envoyées.

b) S rialisation binaire

La s rialisation binaire consiste   convertir une structure de donn es ou un objet en s quence de bits.

Elle se distingue par le fait que les champs publics et priv s de l'objet et le nom de la classe, y compris l'assembly contenant la classe, sont convertis en flux d'octets.

La s rialisation binaire permet de modifier les membres priv s   l'int rieur d'un objet et donc de modifier l' tat de celui-ci. C'est la m thode la plus simple   mettre en place et qui peut stocker le plus d'informations sur le type de l'objet. Elle est  galement utile dans le stockage de donn es et permet une transmission de donn es fiable et rapide.

c) S rialisation XML

La s rialisation XML s rialise uniquement les champs publics et les valeurs de propri t  d'un objet dans un flux de donn es XML.

Le principal avantage de la s rialisation XML est sa flexibilit  au niveau du flux. Le processus de s rialisation XML implique de prendre les donn es de la structure et de les organiser dans une structure de balisage XML en utilisant des balises et des attributs appropri s.

De plus, la s rialisation XML permet d'ajouter des informations de m tadonn es aux donn es, telles que des descriptions de champs, des informations sur les types de donn es et des informations sur la version.

3) La définition des méthodes sous forme algorithmique

Sous Visual Studio, il existe des classes de sérialisation, il suffit de faire un *using* et ensuite d'exploiter les méthodes existantes.

```
using System.Runtime.Serialization.Formatters.Soap;  
using System.Runtime.Serialization.Formatters.Binary;
```

```
using System.Xml.Serialization;
```

Les méthodes existantes, utiliser dans des boutons de Windows Form, cela donne :

POUR SOAP :

```
1 référence  
private void SerializeSOAP_Click(object sender, EventArgs e)  
{  
    Voiture v = new Voiture("1234YT93", "laguna", "bleue", 1250, "12/1/2005");  
    Voiture v1 = new Voiture("4444GT93", "twingo", "verte", 10400, "1/11/2004");  
    Garage g = new Garage();  
    g.AddVoiture(v);  
    g.AddVoiture(v1);  
    FileStream fichier = new FileStream("garage.sr", FileMode.Create);  
    SoapFormatter sf = new SoapFormatter();  
    sf.Serialize(fichier, g);  
    fichier.Close();  
}
```

```
1 référence  
private void DeserializeSOAP_Click(object sender, EventArgs e)  
{  
    FileStream fichier = new FileStream("garage.sr", FileMode.Open);  
    SoapFormatter sf = new SoapFormatter();  
    Garage g = (Garage)sf.Deserialize(fichier);  
    textBox1.Text = g.GetVoiture(1).immatriculation;  
    fichier.Close();  
}
```

POUR BINARY :

1 référence

```
private void SerializeBinaire_Click(object sender, EventArgs e)
{
    Voiture v = new Voiture("1234YT93", "laguna", "bleue", 1250, "12/1/2005");
    Voiture v1 = new Voiture("4444GT93", "twingo", "verte", 10400, "1/11/2004");
    Garage g = new Garage();
    g.AddVoiture(v);
    g.AddVoiture(v1);
    FileStream fichier = new FileStream("garageBinaire.txt", FileMode.Create);
    BinaryFormatter bf = new BinaryFormatter();
    bf.Serialize(fichier, g);
    fichier.Close();
}
```

1 référence

```
private void DeSerializeBinaire_Click(object sender, EventArgs e)
{
    FileStream fichier = new FileStream("garageBinaire.txt", FileMode.Open);
    BinaryFormatter bf = new BinaryFormatter();
    Garage g = (Garage)bf.Deserialize(fichier);
    textBox1.Text = g.GetVoiture(1).immatriculation;
    fichier.Close();
}
```

POUR XML :

1 référence

```
private void SerializeXML_Click(object sender, EventArgs e)
{
    Voiture v = new Voiture("121GFT93", "Twingo", "verte", 1235, "12/12/2004");
    XmlSerializer serializer = new XmlSerializer(typeof(Voiture));
    FileStream fichier = new FileStream("voitures.xml", FileMode.Open);
    serializer.Serialize(fichier, v);
    fichier.Close();
}
```

1 référence

```
private void DeSerializeXML_Click(object sender, EventArgs e)
{
    FileStream fichier = new FileStream("voitures.xml", FileMode.Open);
    XmlSerializer serializer = new XmlSerializer(typeof(Voiture));
    Voiture v = (Voiture)serializer.Deserialize(fichier);
    textBox1.Text = v.immatriculation;
    fichier.Close();
}
```


4) Conclusion

Nous avons donc exploité un morceau de ce à quoi sert la sérialisation. C'est-à-dire qu'elle sert à convertir des données en un format qui peut être stocké ou transféré de manière efficace et qui peut être utilisé pour recréer les données d'origine.

Adapté à différentes données, elle est utile pour le stockage et la récupération de données, la communication de ces données et également en cybersécurité.

Le terme de cybersécurité approprié est le *chiffrement*, les données sont sérialisées avant d'être chiffré.