

SOAP



Objectifs

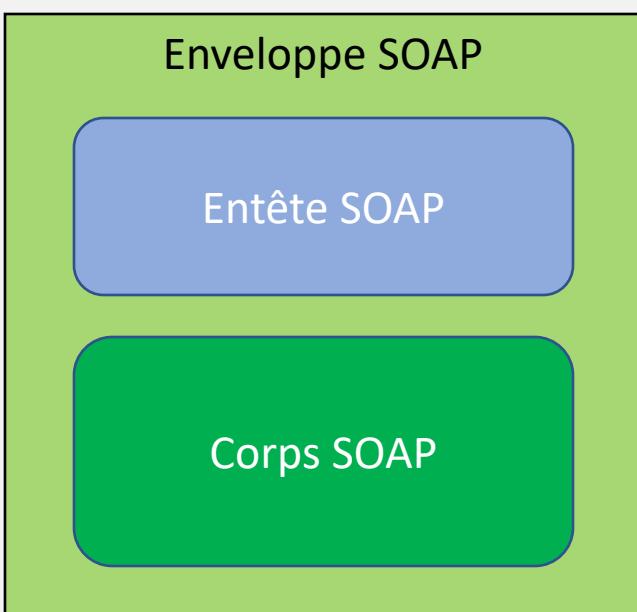
- Protocole SOAP
- Exemple d'implantation en C#

SOAP

- SOAP (*Simple* Object Access Protocol) : protocole d'échanges de données se basant sur XML :
 - Une enveloppe qui définit la structure du message et comment l'interpréter
 - Des règles et des définitions de types
 - Une convention de description d'appels de fonctions et de valeur de retour
- 3 principales caractéristiques :
 - Extensible : beaucoup d'extension de type WS-XYZ comme WS-Security, WS-AT (ACID), etc.
 - Neutre : peut fonction au dessus de TCP, d'UDP, HTTP, etc.
 - Indépendant : ne nécessite pas de technologie/langage de programmation particuliers

SOAP

- On parle souvent d'appel de méthodes distantes ou Remote Procedure Call (RPC)
- Structure



```
<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
                xmlns:m="http://www.example.org">
    <soap:Header>
        </soap:Header>

    <soap:Body>
        <m:GetStockPrice>
            <m:StockName>T</m:StockName>
        </m:GetStockPrice>
    </soap:Body>
</soap:Envelope>
```

Web Services Description Language - WSDL

- Description
 - Des types
 - Format des messages
 - Des fonctions d'un service
- Écrit en XML

```
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://tempuri.org/PingService" xmlns:http="http://schemas.microsoft.com/ws/2004/09/policy/http" xmlns:msc="http://schemas.microsoft.com/ws/2004/09/policy/msc/wsdl" xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata" targetNamespace="http://tempuri.org">
```

Implantation en C# - Serveur

- Créez un projet ASP.Net MVC core
- Ajoutez le package Nuget « SoapCore »
- Créez une interface par service dans un projet séparé afin de les partager avec la partie cliente
- Implantez ces interfaces
- Déclarez les correspondances entre les interfaces et les classes concrètes dans la méthode « ConfigureServices » de la classe « StartUp »
- Dans notre cas nous allons utiliser le protocole HTTP

Implantation en C# - Serveur

```
[ServiceContract]
public interface IEchoService
{
    [OperationContract]
    string Echo(string p_message);
    [OperationContract]
    decimal CalculInteretAnnuel(decimal p_montant, decimal p_taux);
}
```

Implantation en C# - Serveur

```
public class EchoService : IEchoService
{
    public decimal CalculInteretAnnuel(decimal p_montant, decimal p_taux) {
        if (p_montant < 0.0m) {
            throw new ArgumentOutOfRangeException(nameof(p_montant),
                "Le montant ne doit pas être négatif");
        }

        if (p_taux < 0.0m || p_taux > 1.0m) {
            throw new ArgumentOutOfRangeException(nameof(p_taux),
                "Le taux doit être compris entre 0.0 et 1.0");
        }

        return p_montant * p_taux;
    }

    public string Echo(string p_message) {
        return p_message;
    }
}
```

Implantation en C# - Serveur

```
public class Startup
{
    public void ConfigureServices(IServiceCollection services)
    {
        services.AddSingleton<IEchoService, EchoService>();
        services.AddSoapCore();
    }

    public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
    {
        app.UseSoapEndpoint<IEchoService>("/EchoService.svc",
                                         new BasicHttpBinding(),
                                         SoapSerializerDataContractSerializer);
        app.UseSoapEndpoint<IEchoService>("/EchoService.asmx",
                                         new BasicHttpBinding(),
                                         SoapSerializer.XmlSerializer);
        // ...
    }
}
```

Implantation en C# - Client

```
Binding binding = new BasicHttpBinding();
EndpointAddress endpoint = new EndpointAddress(new Uri("http://localhost:5000/EchoService.svc"));
ChannelFactory<IEchoService> channelFactory = new ChannelFactory<IEchoService>(binding, endpoint);
IEchoService echoService = channelFactory.CreateChannel();
```

Implantation en C# - Client

```
string echo = echoService.Echo("Bonjour DSED !");
try
{
    echoService.CalculInteretAnnuel(-1, 0);
}
catch (Exception ex)
{
    Console.Error.WriteLine(ex.Message);
}
```

Implantation en C# - Client

```
int nombreEssais = 0;
int nombreMaximumEssais = 3;
bool appelEffectue = false;
decimal interet = -1m;
while (!appelEffectue && nombreEssais < nombreMaximumEssais) {
    ++nombreEssais;
    try {
        interet = echoService.CalculInteretAnnuel(100, .199m);
        appelEffectue = true;
    }
    catch (Exception) {
        if (nombreEssais >= nombreMaximumEssais) {
            throw;
        }
    }
}
```

Référence

- <https://en.wikipedia.org/wiki/SOAP>
- <https://github.com/DigDes/SoapCore>