

C# : WCF(2)

Modèle de programmation orienté
service

Sommaire

- Gestion des erreurs
- Hébergement du service
- Configuration du service
- Liaisons de service
- Comportement de service
- Tester la configuration du service
- Consommation par le client
- Exemples

Gestion des erreurs

- Exceptions CLR et erreurs SOAP
 - FaultException
 - Côté service : throw new FaultException(message)
 - Côté client : try ... catch(FaultException fe)
 - Gestion des exceptions dans environnement visual studio -> assistant intercepte les exceptions

Gestion des erreurs

- Exceptions fortement typées
 - FaultContract
 - Définition d'un contrat de données pour les erreurs
 - Annoter les opérations de service [FaultContract]
 - [FaultContract(typeof(MyError))]
 - Côté service : instantiation d'un objet « MyError » et `throw new FaultException<MyError>`
 - Côté client : `try catch(FaultException<MyError>)`
 - Accès aux champs via la propriété Detail

Gestion des erreurs

- Exceptions imprévues
 - Web.config / App.config
 - includeExceptionDetailInFaults

Première application WCF:

Étapes

- Définition de contrats
 - Contrat de service /opération
 - Contrat de données
- Implémentation des contrats
- Hébergement du service
- Configuration du service (point d'accès finaux)
- Consommation du service par une application cliente

Hébergement du service - hébergeurs

- IIS (employé à la séance WCF 1)
 - Affecter un port non dynamique
 - Propriétés du projet -> onglet Web -> port spécifique
- Service Windows
- WAS (Windows Activation Service)
- Propre application hôte
 - classe ServiceHost

Hébergement du service – Job de l'hébergeur

- Démarrer /Arrêter service
- Etre à l'écoute des requêtes des clients et les rediriger vers le service (en simultané)
- Envoyer les réponses du service vers les clients
- Respect des exigences de sécurité, fiabilité et transaction, ...

Hébergement de service – Hébergeur job

- le runtime WCF propose des canaux pour gérer cela.
- Canal de transport (trt messages)
 - Canal de transaction, méthodes d'encodage, ...
 - Pile de canaux -> passage de l'un à l'autre
 - ...

Hébergement du service – InstanceContextMode

- Objet InstanceContext
- Modes :
 - Per Call
 - Per Session
 - Single

```
ServiceBehavior (InstanceContextMode=  
    InstanceContextMode.Single])
```

Hébergement du service – classe ServiceHost

- Permet de créer une instance d'un objet service
- Configurer les points de terminaison (*.config)
- Exemple :

```
ServiceHost productServiceHost = new  
ServiceHost(typeof(ProductsServiceImpl));
```

Hébergement du service – classe ServiceHost

- Open()
 - Démarre le service
 - Le service se met à l'écoute des requêtes clientes
- Close()
 - Arrête le service

Consommation par le client

- Générer le(s) classe(s) proxy
 - Svcutil /d:z:\wcf
<http://localhost:8019/HelloWorldImpl.svc>
 - Svcutil se trouve dans c:\program files
(x86)\Microsoft SDKs\Windows\v7.0a\bin
- Ajouter les classes et configurations générées dans le projet client (output.config,)
- Renommer le output.config en app.config

Exemple de configuration de service

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <system.web>
    <compilation debug="true" targetFramework="4.0" />
  </system.web>
  <system.serviceModel>
    <services>
      <service name="WCFMediatheque.MediathequeServiceImpl">
        <endpoint address="MediathequeServiceImpl.svc" binding="basicHttpBinding"
          bindingConfiguration="" contract="WCFMediatheque.IMediathequeService" />
      </service>
    </services>
    <behaviors>
      <serviceBehaviors>
        <behavior>
          <serviceMetadata httpGetEnabled="true"/>
          <serviceDebug includeExceptionDetailInFaults="false"/>
        </behavior>
      </serviceBehaviors>
    </behaviors>
    <serviceHostingEnvironment multipleSiteBindingsEnabled="true" />
  </system.serviceModel>
```

Exemple de configuration de service (ABC)

- `<service name="WCFHelloWorldService.HelloWorldImpl" behaviorConfiguration="helloServiceBehavior">`
- `<endpoint address=""`
- `binding="basicHttpBinding"`
- `contract="WCFHelloWorldService.IHelloWorld" />`
- `<endpoint address="mex" binding="mexHttpBinding"`
- `contract="IMetadataExchange" />`
- `</host>`
- `<baseAddresses>`
- `<add baseAddress="http://localhost:8019/HelloWorldImpl.svc"/>`
- `</baseAddresses>`
- `</host>`
- `</service>`