



TP service web SOAP

Exercice 1 : Outils pour tester les services web.

Postman et SoapUI sont deux outils qui peuvent être utilisés pour le test des services web.

- Consultez la documentation de ces deux outils pour tester vos services web.
 - SoapUI
 - <https://fr.wikipedia.org/wiki/SoapUI>
 - <https://www.soapui.org/>
 - <https://www.soapui.org/soap-and-wsdl/getting-started.html>
 - <https://www.soapui.org/getting-started/soap-test.html>
 - Postman
 - <https://www.postman.com/>
 - <https://blog.postman.com/2017/11/18/postman-makes-soap-requests-too/>
 - <https://learning.postman.com/docs/postman/sending-api-requests/making-soap-requests/>
 - Comparaison
 - <https://www.quora.com/What-is-the-major-difference-between-SoapUI-and-Postman-And-which-one-is-preferred-for-testing-REST-APIs>
 - <https://www.katalon.com/resources-center/blog/soapui-vs-postman-katalon-api-tools/>

Exercice 2 : Création et consommation d'un service web simple.

Suivez le tutoriel ci-dessous pour créer et utiliser un service web SOAP en C# :

- <https://support.microsoft.com/fr-fr/help/308359/how-to-write-a-simple-web-service-by-using-visual-c-net>

Exercice 3 : Utilisation de service web existant.

Consultez le WSDL d'un des services web suivant et proposez une application en C# qui utilise les opérations proposées :

- Service web SOAP dans le domaine de la banque :
 - o <http://www.banguat.gob.gt/variables/ws/TipoCambio.asmx?>
 - o <http://www.banguat.gob.gt/variables/ws/TipoCambio.asmx?WSDL>
- Service web SOAP : calculatrice
 - o <http://www.dneonline.com/calculator.asmx?>
 - o <http://www.dneonline.com/calculator.asmx?WSDL>
- Service web SOAP : Météo
 - o https://graphical.weather.gov/xml/SOAP_server/ndfdXMLserver.php
 - o https://graphical.weather.gov/xml/SOAP_server/ndfdXMLserver.php?wsdl
- Service web SOAP dans le domaine de la banque :
 - o <http://webservices.lb.lt/BLiquidity/BLiquidity.asmx>
 - o <http://webservices.lb.lt/BLiquidity/BLiquidity.asmx?WSDL>

Exercice 4 : Création et utilisation de services web.

L'objectif de ce TP est de développer en C# une application de réservation d'hôtels en ligne. Cette application propose une interface permettant à un utilisateur de saisir les informations suivantes : Une ville de séjour, une date d'arrivée, une date de départ, un intervalle de prix souhaité, une catégorie d'hôtel : nombre d'étoiles, le nombre de personnes à héberger.

En réponse, l'application lui retourne une liste d'hôtels qui répondent à ses critères et où pour chaque hôtel les informations suivantes sont affichées : nom de l'hôtel, adresse de l'hôtel (pays, ville, rue, numéro, lieu-dit, position GPS), le prix à payer, nombre d'étoile, nombre de lits proposés.

L'utilisateur choisira un hôtel dans la liste proposée et l'application lui demandera les informations suivantes : le nom et prénom de la personne principale à héberger, les informations de la carte de crédit de paiement. L'application utilisera ces informations pour réaliser la réservation de l'hôtel en question.

Question 1 : Version sans distribution.

1.1 Proposez une conception en UML de l'application non distribuée permettant de répondre aux requêtes de l'utilisateur. Au minimum, pensez aux concepts suivants : hôtel, chambre, réservation, client.

1.2 Proposez une implémentation en C# de la conception UML précédente. Cette implémentation ne doit pas gérer la persistance des données via une base de données (les données seront codées en dur dans le code ou dans des fichiers).

Question 2 : Version distribuée-Agence de voyage et hôtels

- 2.1 Modifiez la conception UML précédente pour intégrer le fait qu'un hôtel peut avoir des agences partenaires qui chacune bénéficie d'un tarif propre (un pourcentage par rapport à un prix de base d'une chambre).
- 2.2 Proposez une version distribuée de l'application sans base de données où :
- La réservation d'hôtels se fait via une agence de voyage qui accède aux données des hôtels via leurs services web respectifs. Les prix proposés par les hôtels peuvent être adaptés en fonction des agences (les agences peuvent avoir des prix différents pour les mêmes produits/chambres). L'utilisateur final est un client de l'agence qui dispose d'un menu pour bénéficier des services de l'agence.
 - Chaque hôtel aura son propre serveur et ainsi ses propres services web pour pouvoir :
 - Service web 1 : Consulter les disponibilités par les agences partenaires où :
 - Les données en entrée du service web concernent :
 - Identifiant et mot de passe de l'agence.
 - Dates début et fin de la réservation.
 - Nombre de personnes à héberger.
 - Les données en réponse du service web :
 - Une liste d'offres où une offre est caractérisée par :
 - Un Identifiant de l'offre.
 - Type de chambre : nombre de lits.
 - Date de disponibilité.
 - Prix.
 - Service web 2 : Effectuer une réservation où :
 - Les données en entrée du service web concernent :
 - Identifiant de l'agence.
 - Login.
 - Mot de passe.
 - Identifiant de l'offre.
 - Informations personne principale.
 - Les données en réponse du service web :
 - Confirmation de la réservation ou problème.
 - Référence de la réservation si cette dernière est confirmée.

Question 3 : Version distribuée- service web intègre des images.

- 3.1 Modifier l'application précédente pour que le service « Service web 1 : Consulter les disponibilités par les agences partenaires » puisse retourner une image de la chambre proposée au client.

Question 4 (Bonus) : Version distribuée- Base de données et interface graphique.

- 4.1 Modifiez la version distribuée de l'application pour intégrer une base de données pour stocker les données persistantes.
- 4.2 Proposez une interface graphique pour l'application agence.