

TUYÊN BỐ BẢN QUYỀN

Tài liệu này thuộc loại sách giáo trình nên các nguồn thông tin có thể được phép dùng nguyên bản hoặc trích dùng cho các mục đích về đào tạo và tham khảo.

Mọi mục đích khác mang tính lệch lạc hoặc sử dụng với mục đích kinh doanh thiếu lành mạnh sẽ bị nghiêm cấm.

LỜI GIỚI THIỆU

Yêu cầu có các tài liệu tham khảo cho sinh viên của khoa Công nghệ Thông tin - Trường Cao đẳng Nghề ngày càng trở nên cấp thiết. Việc biên soạn tài liệu này nằm trong kế hoạch xây dựng hệ thống giáo trình các môn học của Khoa.

Đề cương của giáo trình đã được thông qua Hội đồng Khoa học của Khoa và Trường. Mục tiêu của giáo trình nhằm cung cấp cho sinh viên một tài liệu tham khảo chính về môn học Mạng máy tính, trong đó giới thiệu những khái niệm căn bản nhất về hệ thống mạng máy tính, đồng thời trang bị những kiến thức và một số kỹ năng chủ yếu cho việc bảo trì và quản trị một hệ thống mạng. Đây có thể coi là những kiến thức ban đầu và nền tảng cho các kỹ thuật viên, quản trị viên về hệ thống mạng.

Mặc dù đã có những cố gắng để hoàn thành giáo trình theo kế hoạch, nhưng do hạn chế về thời gian và kinh nghiệm soạn thảo giáo trình, nên tài liệu chắc chắn còn những khiếm khuyết. Rất mong nhận được sự đóng góp ý kiến của các thầy cô trong Khoa cũng như các bạn sinh viên và những ai sử dụng tài liệu này. Các góp ý xin gửi về Khoa Công nghệ thông tin - Trường Cao đẳng nghề Cần Thơ. Xin chân thành cảm ơn.

Cần Thơ, ngày tháng năm 2021

Tham gia biên soạn

1. Chủ biên Nguyễn Phát Minh

MỤC LỤC

TRANG

LỜI GIỚI THIỆU	2
MỤC LỤC	3
TRANG	3
GIÁO TRÌNH MÔN HỌC/MÔ ĐUN	5
Tên môn học/mô đun: MANG MÁY TÍNH.....	5
Mã môn học/mô đun: MH 07	5
CHƯƠNG 1: TỔNG QUAN VỀ CÔNG NGHỆ MẠNG MÁY TÍNH	7
Mã chương: MH 07 - 01	7
1. Lịch sử mạng máy tính:	Error! Bookmark not defined.
2. Giới thiệu mạng máy tính:	Error! Bookmark not defined.
2.1. Định nghĩa mạng máy tính và mục đích của việc kết nối mạng: ..	Error! Bookmark not defined.
2.2. Đặc trưng kỹ thuật của mạng máy tính:	Error! Bookmark not defined.
3. Phân loại mạng máy tính:	Error! Bookmark not defined.
3.1. Phân loại mạng theo khoảng cách địa lý :	Error! Bookmark not defined.
3.2. Phân loại theo kỹ thuật chuyển mạch:	Error! Bookmark not defined.
3.3. Phân loại theo kiến trúc mạng sử dụng:	Error! Bookmark not defined.
3.4. Phân loại theo hệ điều hành mạng:	Error! Bookmark not defined.
4. Giới thiệu các mạng máy tính thông dụng nhất:	Error! Bookmark not defined.
4.1. Mạng cục bộ:	Error! Bookmark not defined.
4.2. Mạng diện rộng với kết nối LAN TO LAN:	Error! Bookmark not defined.
4.3. Liên mạng INTERNET:	Error! Bookmark not defined.
4.4. Mạng INTRANET:	Error! Bookmark not defined.
5. CÂU HỎI ÔN TẬP:	Error! Bookmark not defined.
CHƯƠNG 2: MÔ HÌNH HỆ THỐNG MÔ OSI	Error! Bookmark not defined.
Mã chương: MH 07 - 02	Error! Bookmark not defined.
1. Mô hình tham khảo OSI:	Error! Bookmark not defined.
2. Các giao thức trong mô hình OSI	Error! Bookmark not defined.
3. Các chức năng chủ yếu của các tầng của mô hình OSI: ...	Error! Bookmark not defined.
3.1. Tầng 1: Vật lý (Physical).....	Error! Bookmark not defined.
3.2. Tầng 2: Liên kết dữ liệu (Data link)	Error! Bookmark not defined.
3.3. Tầng 3: Mạng (Network).....	Error! Bookmark not defined.
3.4. Tầng 4: Vận chuyển (Transport):	Error! Bookmark not defined.
3.5. Tầng 5: Giao dịch (Session):	Error! Bookmark not defined.
3.6. Tầng 6: Trình bày (Presentation).....	Error! Bookmark not defined.
3.7. Tầng 7: ứng dụng (Application)	Error! Bookmark not defined.
4. CÂU HỎI ÔN TẬP:	Error! Bookmark not defined.
CHƯƠNG 3: TÔ PÔ MẠNG	Error! Bookmark not defined.
Mã chương: MH 07 - 03	Error! Bookmark not defined.
1. Mạng cục bộ :	Error! Bookmark not defined.
2. Kiến trúc mạng cục bộ :	Error! Bookmark not defined.
2.1. Mạng hình sao:	Error! Bookmark not defined.
2.2. Mạng trục tuyến tính (Bus):	Error! Bookmark not defined.
2.3 Mạng hình vòng:.....	Error! Bookmark not defined.
2.4 Kết nối hỗn hợp:	Error! Bookmark not defined.
4. CÂU HỎI ÔN TẬP:	Error! Bookmark not defined.
5. PHẦN THỰC HÀNH:	Error! Bookmark not defined.

CHƯƠNG 4: CÁP MẠNG VÀ VẬT TẢI TRUYỀN	Error! Bookmark not defined.
Mã chương: MH 07 - 04	Error! Bookmark not defined.
1. Các thiết bị mạng thông dụng	Error! Bookmark not defined.
1.1. Các loại cáp truyền.....	Error! Bookmark not defined.
1.1.1. Cáp đôi dây xoắn (Twisted pair cable).....	Error! Bookmark not defined.
1.1.2. Cáp đồng trục (Coaxial cable) bằng tần cơ sở.....	Error! Bookmark not defined.
1.1.3. Cáp đồng trục băng rộng (Broadband Coaxial Cable).....	Error! Bookmark not defined.
defined.	
1.1.4. Cáp quang	Error! Bookmark not defined.
2. Các thiết bị ghép nối	Error! Bookmark not defined.
2.1. Card giao tiếp mạng (Network Interface Card viết tắt là NIC)...	Error! Bookmark not defined.
defined.	
2.2. Bộ chuyển tiếp (REPEATER).....	Error! Bookmark not defined.
2.3. Các bộ tập trung (Concentrator hay HUB)	Error! Bookmark not defined.
2.4. Switching Hub (hay còn gọi tắt là switch)	Error! Bookmark not defined.
2.5. Modem	Error! Bookmark not defined.
2.6. Multiplexor - Demultiplexor.....	Error! Bookmark not defined.
2.7. Router.....	Error! Bookmark not defined.
3. Một số kiểu nối mạng thông dụng và các chuẩn.....	Error! Bookmark not defined.
3.1. Các thành phần thông thường trên một mạng cục bộ gồm có	Error! Bookmark not defined.
defined.	
3.2. Kiểu 10BASE5:.....	Error! Bookmark not defined.
3.3. Kiểu 10BASE2:.....	Error! Bookmark not defined.
3.4. Kiểu 10BASE-T.....	Error! Bookmark not defined.
3.5. Kiểu 10BASE-F	Error! Bookmark not defined.
4. CÂU HỎI ÔN TẬP :	Error! Bookmark not defined.
5. BÀI THỰC HÀNH:	Error! Bookmark not defined.
CHƯƠNG 5: GIỚI THIỆU GIAO THỨC TCP/IP.....	Error! Bookmark not defined.
Mã chương: MH 07 - 05	Error! Bookmark not defined.
1. Giao thức IP	Error! Bookmark not defined.
1.1. Họ giao thức TCP/IP.....	Error! Bookmark not defined.
1.2. Chức năng chính của - Giao thức liên mạng IP(v4)	Error! Bookmark not defined.
1.3. Địa chỉ IP	Error! Bookmark not defined.
1.4. Cấu trúc gói dữ liệu IP	Error! Bookmark not defined.
1.5. Phân mảnh và hợp nhất các gói IP	Error! Bookmark not defined.
1.6. Định tuyến IP	Error! Bookmark not defined.
2. Một số giao thức điều khiển.....	Error! Bookmark not defined.
2.1. Giao thức ICMP	Error! Bookmark not defined.
2.2. Giao thức ARP và giao thức RARP	Error! Bookmark not defined.
2.3. Giao thức lớp chuyên tải (Transport Layer).....	Error! Bookmark not defined.
2.3.1 <i>Giao thức TCP</i>	Error! Bookmark not defined.
2.3.2 <i>Cấu trúc gói dữ liệu TCP</i>	Error! Bookmark not defined.
2.3.3 <i>Thiết lập và kết thúc kết nối TCP</i>	Error! Bookmark not defined.
3. CÂU HỎI ÔN TẬP:.....	Error! Bookmark not defined.
4. BÀI THỰC HÀNH:	Error! Bookmark not defined.
TÀI LIỆU THAM KHẢO.....	61

GIÁO TRÌNH MÔN HỌC/MÔ ĐUN

Tên môn học/mô đun: LẬP TRÌNH WEBSERVICE

Mã môn học/mô đun: MĐ 26

Vị trí, tính chất, ý nghĩa và vai trò của môn học/mô đun:

- Vị trí: Môn học được bố trí sau khi sinh viên học xong các môn học chung, các môn học cơ sở chuyên ngành đào tạo chuyên môn nghề.
- Tính chất: Là môn học cơ sở chuyên ngành bắt buộc.
- Ý nghĩa và vai trò của môn học/mô đun: Lập trình Webservice là môn học nâng cao từ môn Lập trình Website với ASP.NET để sinh viên tìm hiểu về các khái niệm về webservice để làm nền tảng cho sinh viên có thể tự học nâng cao kiến thức về lập trình website trên nền tảng .NET

Mục tiêu của môn học/mô đun:

- Về kiến thức:
 - o Hiểu về các tác nhân đối với Web Service: Service provider, Service registry
 - o Hiểu về kiến trúc phân tầng của Web Service
 - o Hiểu về mục đích, vai trò thiết kế của SOA đối với Web Service
 - o Hiểu về các giai đoạn chính để xây dựng Web Service
 - o Hiểu về các cách tiếp cận để quyết định cách thức xây dựng Web Service: bottom-up, top-down, from-scratch
 - o Nắm vững quy trình hoàn thiện Web Service
 - o Hiểu về cách khai thác các dịch vụ Web đã được cung cấp để xây dựng ứng dụng
 - o Hiểu về vấn đề bảo mật trên Internet và Web Service đáp ứng chuẩn an toàn cơ bản
 - o Hiểu về một số kiểu giả mạo, đánh cắp thông tin và cách phòng chống
- Về kỹ năng:
 - o Xây dựng dịch vụ Web bảo mật với Visual Studio - ASP.NET, triển khai và công bố dịch vụ Web
- Về năng lực tự chủ và trách nhiệm:
 - o Nghiêm túc, tỉ mỉ trong quá trình tiếp cận với công cụ mới.
- Bố trí làm việc khoa học đảm bảo an toàn cho người và phương tiện học tập.

Nội dung của môn học/mô đun:

Số TT	Tên chương, mục	Thời gian			
		Tổng số	Lý thuyết	Thực hành Bài tập	Kiểm tra* (LT hoặc TH)
I	Công nghệ WEB SERVICE Web Service là gì ?	4	2	2	

	Đặc điểm của Web service				
	Nền tảng của Web service				
	Các công nghệ của Web service				
II	Kiến trúc WEB SERVICE	4	2	2	
	Cơ chế hoạt động của Web Service				
	Kiến trúc phân tầng của Web Service				
	Kiến trúc hướng dịch vụ SOA				
III	Xây dựng WEB SERVICE	22	11	10	1
	Các vấn đề cần xác định rõ trước khi bắt tay xây dựng ứng dụng Web service				
	Xây dựng Web Service với ASP.NET				
	Xây dựng Web Service với Java				
IV	Khai thác WEB SERVICE	22	11	10	1
	Ứng dụng Window Form kết nối tới Web Service				
	Ứng dụng Java Swing kết nối tới Web Service				
	Ứng dụng Web ASP.NET kết nối tới Web Service				
V	Bảo mật trong WEB SERVICE	8	4	3	1
	Tổng quan về vấn đề bảo mật				
	Một số kiểu giả mạo, đánh cắp thông tin và cách phòng chống				
	Bảo mật trong web service				
	Thực hiện bảo mật trong web service				
	Cộng	60	30	27	3

BÀI 1: CÔNG NGHỆ WEB SERVICE

Mã bài: MĐ 26 - 01

Giới thiệu:

Trong bài này trình bày những kiến thức căn bản về Webservice, các đặc điểm của webservice cùng với các nền tảng và công nghệ của webservice

Mục tiêu của bài:

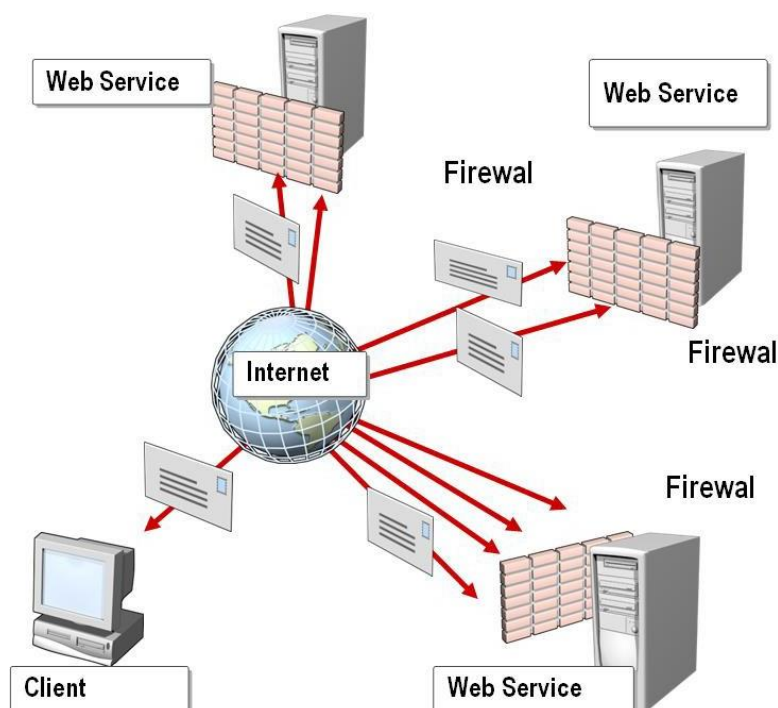
- Trình bày được khái niệm webservice.
- Trình bày được đặc điểm của webservice.
- Trình bày được các nền tảng của webservice.
- Trình bày được các công nghệ của webservice
- Thực hiện các thao tác an toàn với máy tính.

Nội dung chính:

1. Web Service là gì ?

Theo định nghĩa của W3C (World Wide Web Consortium), Web service là một hệ thống phần mềm được thiết kế để hỗ trợ khả năng tương tác giữa các ứng dụng trên các máy tính khác nhau trong môi trường Internet thông qua các giao diện (Interface) chung và sự gắn kết được mô tả bằng XML

Web service là tài nguyên phần mềm có thể xác định bằng địa chỉ URL để thực hiện các chức năng và đưa thông tin ra cho người dùng.

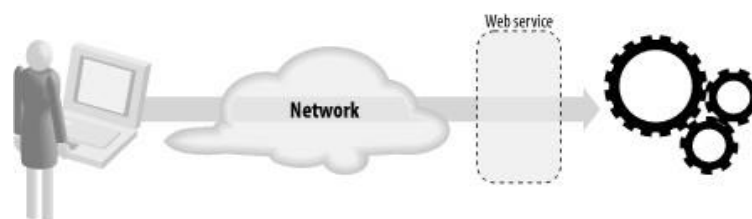


Web service được tạo ra bằng cách lấy các chức năng và đóng gói chúng sao cho các ứng dụng khác dễ dàng tìm thấy và truy cập tới các dịch vụ mà nó cung cấp, đồng thời vẫn có thể yêu cầu thông tin từ các dịch vụ khác. Web service bao gồm các mô đun độc lập để thực hiện yêu cầu nghiệp vụ của doanh nghiệp và được thực thi trên Server.

Ứng dụng cơ bản của Web service là tích hợp các hệ thống và là một trong những hoạt động chính khi phát triển hệ thống. Trong hệ thống này, các ứng dụng cần được tích hợp với cơ sở dữ liệu (CSDL) và các ứng dụng khác, người sử dụng sẽ giao tiếp với CSDL để tiến hành phân tích và lấy dữ liệu.

2. Đặc điểm của Web service

Web Service cho phép các ứng dụng khác nhau từ các nguồn khác nhau có thể giao tiếp với các ứng dụng khác mà không đòi hỏi nhiều thời gian coding, do tất cả các quá trình giao tiếp đều tuân theo định dạng XML, cho nên Web Service không bị phụ thuộc vào bất kỳ hệ điều hành hay ngôn ngữ lập trình nào. Web service cho phép client và server có thể tương tác được với nhau trên các nền tảng khác nhau mà không cần bất cứ thay đổi hay yêu cầu đặc biệt nào. Ví dụ, chương trình viết bằng ngôn ngữ Java cũng có thể trao đổi dữ liệu với các chương trình viết bằng Perl, các ứng dụng chạy trên nền Windows cũng có thể trao đổi dữ liệu với các ứng dụng chạy trên nền Linux. Công nghệ Web Service không yêu cầu phải sử dụng trình duyệt và ngôn ngữ HTML.



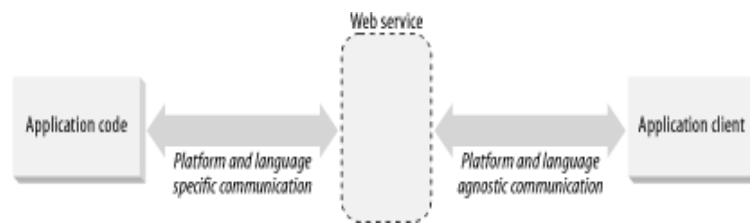
Phần lớn kỹ thuật của Web service được xây dựng trên mã nguồn mở và được phát triển từ các chuẩn đã được công nhận. Nó tích hợp các ứng dụng trên nền web lại với nhau bằng cách sử dụng các công nghệ XML, SOAP, WSDL, và UDDI trên nền tảng các giao thức Internet với mục tiêu tích hợp ứng dụng và truyền thông điệp. XML được sử dụng để đánh dấu dữ liệu, SOAP được dùng để truyền dữ liệu, WSDL được sử dụng để mô tả các dịch vụ có sẵn và UDDI được sử dụng để liệt kê những dịch vụ nào hiện tại đang có sẵn để có thể sử dụng. Web Service cho phép các tổ chức có thể trao đổi dữ liệu với nhau mà không cần phải có kiến thức hiểu biết về hệ thống thông tin đứng sau Firewall.

Web service có thể gồm nhiều mô đun và được công bố trên Internet.

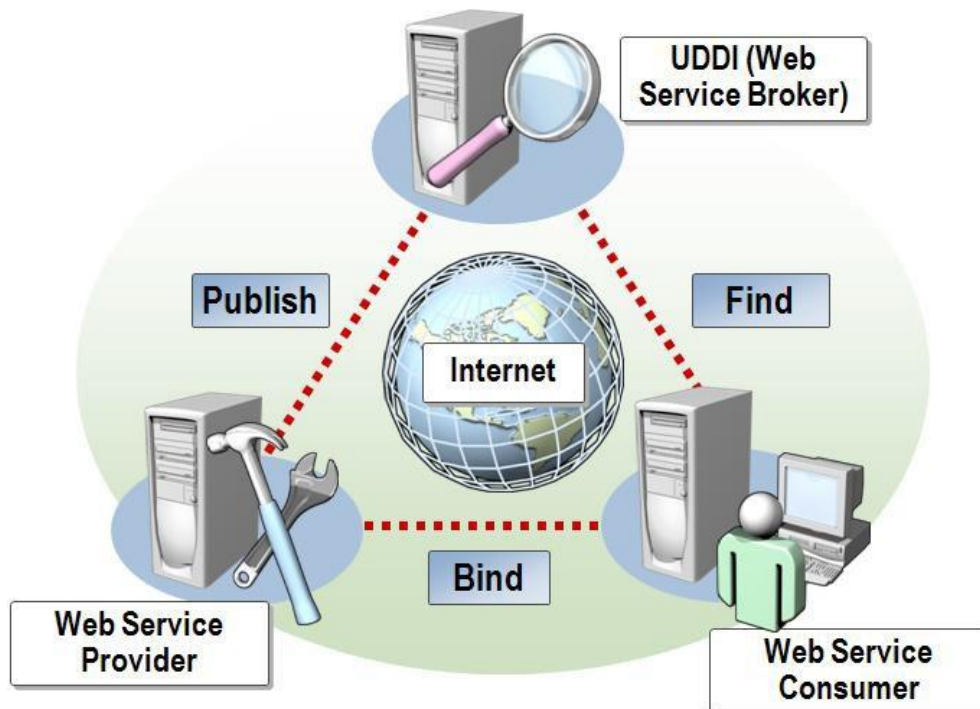
Là sự kết hợp của việc phát triển theo hướng từng thành phần với những lĩnh vực cụ thể và cơ sở hạ tầng Web, đưa ra lợi ích cho cả doanh nghiệp, khách hàng, những nhà cung cấp dịch vụ khác và cả các cá nhân thông qua mạng Internet.

Web service khi được triển khai sẽ hoạt động theo mô hình client-server. Nó có thể được triển khai bởi một phần mềm ứng dụng phía Server như PHP, JSP, ASP.NET, ... Không giống như mô hình client-server truyền thống, chẳng hạn như hệ thống Web server- web page, Web Service không cung cấp cho người dùng một giao diện đồ họa nào, Web Service đơn thuần chỉ là việc chia sẻ các dữ liệu logic và xử lý các dữ liệu đó

thông qua một giao diện chương trình ứng dụng được cài đặt xuyên suốt trên mạng máy tính.



Tính tương thích (Inteoperability) là một lợi thế vô cùng mạnh mẽ của Web Service, thông thường, các công nghệ Java và công nghệ của Microsoft rất khó có thể tích hợp được với nhau, nhưng với Web Service thì các Application và Client sử dụng 2 công nghệ trên hoàn toàn có khả năng tương tác với nhau thông qua Web Service.



3. Nền tảng của Web service

Dịch vụ Web cũng có thể được nói một cách khác là các khối cơ bản được xây dựng để di chuyển trong hệ thống máy tính phân tán trên Internet. Các chuẩn mở và việc tập trung vào giao tiếp và làm việc cộng tác giữa con người và các ứng dụng đã tạo nên một môi trường nơi mà Web service đang trở thành nền tảng cho việc tích hợp ứng dụng. Các ứng dụng được xây dựng sử dụng các Web service các loại từ nhiều nguồn khác nhau làm việc cùng với nhau bất kể là chúng ở đâu hoặc chúng đã được triển khai như thế nào. Có thể có các định nghĩa khác nhau về Web service khi các công ty xây dựng chúng, nhưng hầu hết tất cả các định nghĩa đều có chung các điểm sau:

- Thứ nhất, Web service đưa ra chức năng hữu dụng cho người sử dụng Web thông qua một giao thức chuẩn Web. Trong hầu hết các trường hợp, giao thức được sử dụng đó là SOAP.

□ Thứ hai, Web service đưa ra cách mô tả các giao diện của chúng một cách đủ chi tiết nhằm cho phép người sử dụng xây dựng một ứng dụng máy trạm để giao tiếp được với chúng. Mô tả này thường được cung cấp ở dạng một tài liệu XML gọi là một tài liệu về ngôn ngữ mô tả Web service – WSDL (Web service Description Language).

□ Thứ ba, Web service được đăng ký sao cho các khách hàng tiềm năng là người sử dụng có thể tìm thấy chúng một cách dễ dàng. Điều này được thực hiện với UDDI (Universal Discovery Description and Integration).

Web service như một dịch vụ phần mềm được trình bày trên Web thông qua giao thức SOAP, được mô tả bằng một tệp WSDL và được đăng ký trong UDDI. Các dịch vụ Web service là nguồn thông tin mà ta có thể dễ dàng kết hợp vào các ứng dụng. Dễ dàng nhận ra toàn bộ lớp ứng dụng có thể được xây dựng để phân tích và tích hợp thông tin ta quan tâm và trình bày nó theo nhiều cách khác nhau.

Việc trình bày các ứng dụng đang có như các dịch vụ Web service cho phép người sử dụng xây dựng các ứng dụng có các tính năng mạnh hơn thông qua việc sử dụng Web service như những block được xây sẵn. Ví dụ, người sử dụng có thể phát triển một ứng dụng mua bán để tự động lấy các thông tin về giá cả từ nhiều nhà cung cấp khác nhau, cho phép người dùng chọn một nhà cung cấp, chuyển đơn hàng và sau đó theo dõi việc chuyển hàng cho tới khi nhận được hàng. Ứng dụng của nhà cung cấp, khi trình bày các dịch vụ của họ trên Web, có thể quay ra sử dụng các dịch vụ Web service để kiểm tra tín dụng của khách hàng, lấy tiền từ tài khoản của khách hàng và thiết lập việc chuyển hàng với một công ty vận tải.

4. Các công nghệ của Web service

4.1. Ngôn ngữ XML – RPC

- XML : được viết tắt của cụm từ Extensible Markup Language – Ngôn ngữ đánh dấu dữ liệu.
- RPC – được viết tắt của cụm từ Remote Procedure Call – Thủ tục gọi từ xa. RPC cung cấp cho người phát triển kỹ thuật để định nghĩa ra một giao diện mà có thể được gọi từ xa thông qua môi trường mạng máy tính. Giao diện này có thể là một hàm đơn giản nhưng cũng có thể là một thư viện API khổng lồ.
- XML – RPC là một hướng tiếp cận dễ và rõ ràng nhất cho Web Service, nó cung cấp phương thức gọi một ứng dụng từ một máy tính local đến một máy tính từ xa thông qua môi trường mạng.
- XML – RPC cho phép chương trình có khả năng tạo ra các hàm hoặc các thủ tục gọi hàm thông qua mạng máy tính.
- XML – RPC sử dụng giao thức HTTP để vận chuyển thông tin từ Client đến Server.
- XML – RPC sử dụng ngôn ngữ XML để mô tả các thông điệp yêu cầu và các thông điệp đáp ứng gắn gửi với ngôn ngữ tự nhiên.

- XML – RPC Client chỉ ra cụ thể các thông tin về tên thủ tục, các tham biến trong thông điệp XML request, và Server trả về lỗi hoặc trả về thông điệp response trong thông điệp XML response.
- Các tham số của XML-RPC đơn giản chỉ là kiểu dữ liệu và nội dung – tuy nhiên các cấu trúc dữ liệu phức tạp như struct, array cũng được hỗ trợ bởi XML –RPC

4.2. Giao thức truyền thông điệp SOAP

SOAP viết tắt cho cụm từ - Simple Object Access Protocol. Trong kiến trúc phân tầng của Web Service, SOAP nằm ở tầng Packaging, SOAP là một giao thức đóng gói cho các dữ liệu chia sẻ giữa các ứng dụng. Xét về cơ bản, SOAP là XML, chính vì thế SOAP là một ứng dụng cụ thể của XML. SOAP được xây dựng lên từ các chuẩn XML như XML Schema và XML Namespaces dùng cho việc định nghĩa SOAP và các chức năng của nó

4.3. Ngôn ngữ mô tả Web Service - WSDL

4.3.1. Tổng quan về WSDL

WSDL viết tắt của cụm từ Web Service Description Language – Ngôn ngữ mô tả Web Service. WSDL ra đời dưới sự phát triển của IBM và Microsoft.

WSDL dựa trên giao thức XML để trao đổi thông tin trong môi trường tập trung hoặc phân tán. WSDL mô tả cách thức truy cập tới Web Service và các hành động thực thi trên Web Service đó.

WSDL là ngôn ngữ cho việc mô tả các giao diện Web Service dựa trên nền tảng XML. WSDL là ngôn ngữ mà UDDI sử dụng.

4.3.2. Các thành phần của WSDL

Một tài liệu WSDL thường bao gồm các thành phần chính sau đây:

Thành phần	Mô tả
<type>	Định nghĩa kiểu dữ liệu được dùng trong Web Service
<message>	Các thông điệp được sử dụng trong Web Service
<porttype>	Các thao tác được thực thi bởi Web Service
<binding>	Các giao thức giao tiếp dùng cho Web Service

4.4. Đăng ký dịch vụ UDDI

4.4.1. Tổng quan về UDDI

UDDI là một chuẩn dựa trên XML dùng cho việc mô tả, công bố và tìm kiếm Web Service. UDDI được viết tắt của Universal Description, Discovery and Integration.

UDDI là thư mục dùng cho việc lưu trữ các thông tin về Web Service. UDDI là thư mục của một giao diện Web Service được mô tả bởi WSDL.

UDDI giao tiếp thông qua SOAP. UDDI cùng với SOAP và WSDL được xem là 3 chuẩn của Web Service. UDDI là một kỹ thuật mở đầu tiên cho phép các quy trình thương mại điện tử có thể khám phá lẫn nhau và định nghĩa cách thức tương tác với nhau qua Internet.

4.4.2. Các thành phần của UDDI

UDDI gồm 2 thành phần chính:

- Phần đăng ký của tất cả các Web Service's metadata, bao gồm cả việc trỏ đến tài liệu WSDL mô tả dịch vụ[16].
- Phần thiết lập WSDLPort type định nghĩa cho các thao tác và tìm kiếm thông tin đăng ký.

UDDI xây dựng dựa trên các giao thức chuẩn Internet được công bố bởi W3C và IETF như XML, HTTP, và DNS. UDDI sử dụng WSDL để mô tả giao diện của Web Service. Thêm nữa tính năng độc lập với nền tảng ngôn ngữ lập trình đã được điều hợp cùng với giao thức SOAP

Bài tập:

Bài tập nâng cao:

VPS là loại mạng gì? Trình bày ưu điểm và khuyết điểm?

Những trọng tâm cần chú ý trong bài:

- Lợi ích khi kết nối các máy tính thành mạng
- Các loại mạng thường gặp và đặc trưng của nó

Yêu cầu về đánh giá kết quả học tập:

Nội dung:

+ Về kiến thức: Trình bày được các lợi ích khi kết nối các máy tính thành mạng, vì sao mạng chuyển mạch gói có tốc độ trao đổi thông tin nhanh hơn tốc độ trao đổi thông tin trong mạch chuyển mạch tin báo?, thế nào là mạng cục bộ LAN (Local Area Networks) và nêu các đặc trưng cơ bản của nó.

+ Về kỹ năng: phân biệt được mạng cục bộ, mạng đô thị, mạng diện rộng.

+ Năng lực tự chủ và trách nhiệm: Tỉ mỉ, cẩn thận, chính xác, ngăn nắp trong công việc.

Phương pháp:

+ Về kiến thức: Được đánh giá bằng hình thức kiểm tra viết, trắc nghiệm, vấn đáp

+ Về kỹ năng: Đánh giá kỹ năng phân biệt được các loại mạng.

+ Năng lực tự chủ và trách nhiệm: Tỉ mỉ, cẩn thận, chính xác, ngăn nắp trong công việc.

BÀI 2: KIẾN TRÚC WEB SERVICE

Mã bài: MĐ 26 - 02

Giới thiệu:

Trong bài này trình bày các cơ chế hoạt động webservice, kiến trúc phân tầng và hướng dịch vụ SOA của webservice

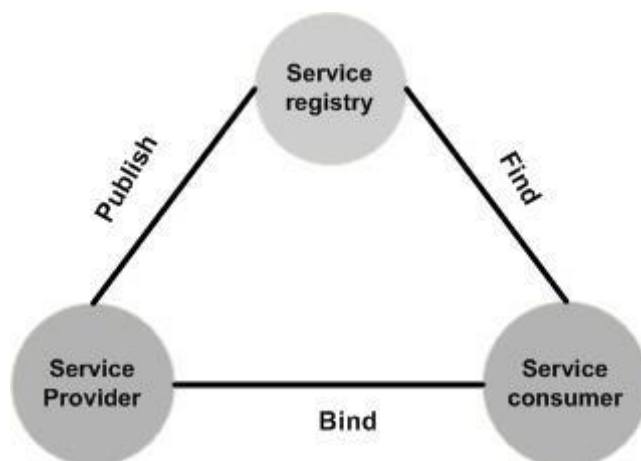
Mục tiêu của bài:

- Trình bày được cơ chế hoạt động webservice.
- Trình bày được kiến trúc phân tầng của webservice.
- Trình bày được kiến trúc hướng dịch vụ SOA.
- Thực hiện các thao tác an toàn với máy tính.

Nội dung chính:

1. Cơ chế hoạt động của Web Service

Cơ chế hoạt động của Web Service yêu cầu phải có 3 thao tác đó là : Find, Public, Bind.



Trong kiến trúc Web Service, Service Provider công bố các mô tả về các service thông qua Service Registry. Service Consumer tìm kiếm trong các Service Registry để tìm ra các service mà họ cần sử dụng. Service Consumer có thể là một người hoặc cũng có thể là một chương trình.

Kĩ thuật mô tả dịch vụ là một trong những thành phần chủ chốt của kiến trúc Web Service. Các thông tin mô tả đầy đủ nhất về kiến trúc Web Service được thể hiện trong hai tài liệu riêng biệt, đó là NASSL – Network Accessible Service Specification Language và WDS – Web-Defined Service. NASSL là một tài liệu dưới dạng chuẩn của XML cho các service chạy trên nền Network, nó được sử dụng để chỉ ra các thông tin hoạt động của Web Service, chẳng hạn như danh sách các service, các mô tả về service, ngày hết hạn của service và các thông tin liên quan đến các Service Provider, như tên, địa chỉ. Tài liệu WDS là một tài liệu mang tính đáp ứng đầy đủ cho tài liệu NASSL. Khi

ta kết hợp hai tài liệu này với nhau ta sẽ có được sự mô tả một cách đầy đủ về các dịch vụ để cho phía yêu cầu dịch vụ có thể dễ dàng tìm kiếm và gọi các dịch vụ đó

2. Kiến trúc phân tầng của Web Service



Mô hình kiến trúc phân tầng của Web Service tương tự với mô hình TCP/IP được sử dụng để mô tả kiến trúc Internet



Các tầng truyền thống như Packaging, Description, và Discovery trong mô hình Web Service Stack là những tầng cung cấp khả năng tích hợp và cần thiết cho mô hình ngôn ngữ lập trình trung lập.

Tầng Discovery : Tầng Discovery cung cấp cơ chế cho người dùng khả năng lấy các thông tin mô tả về các Service Provider. Công nghệ được sử dụng tại tầng này đó chính là UDDI – Universal Description, Discovery and Integration.

Tầng Description : Khi Web Service được thực thi, nó cần phải đưa ra các quyết định về các giao thức trên các tầng Network, Transport, Packaging mà nó sẽ hỗ trợ trong quá trình thực thi. Các mô tả về dịch vụ sẽ đưa ra phương pháp để làm thế nào mà các Service Consumer có thể liên kết và sử dụng các service đó. Tại tầng Description, công nghệ được sử dụng ở đây chính là WSDL (Web Service Description Language) – Ngôn ngữ mô tả Web Service. Ngoài ra, ít phổ biến hơn, chúng ta còn có 2 ngôn ngữ khác được định nghĩa bởi tổ chức W3C đó là ngôn ngữ mô tả tài nguyên - W3C's Resource Description Framework (RDF) và ngôn ngữ đánh dấu sự kiện DARPA

Cả hai ngôn ngữ này đều có khả năng cung cấp việc mô tả Web Service mạnh hơn ngôn ngữ WSDL tuy nhiên do tính phức tạp của chúng nên không được phát triển rộng rãi. Chúng tôi sẽ đề cập đến ngôn ngữ WSDL một cách cụ thể hơn trong phần “Các công nghệ của Web Service ” tại chương 2 của khóa luận này.

□ Tầng Packaging: Việc thực hiện vận chuyển các dữ liệu Web Service được thực hiện bởi tầng Transport, tuy nhiên trước khi được vận chuyển, các dữ liệu cần phải được đóng gói lại theo các định dạng đã định trước để các thành phần tham gia vào mô hình Web Service có thể hiểu được, việc đóng gói dữ liệu được thực hiện bởi tầng Packaging. Việc đóng gói dữ liệu bao gồm các công việc định dạng dữ liệu, mã hóa các giá trị đi kèm dữ liệu đó và các công việc khác.

Các dữ liệu có thể được đóng gói dưới dạng các tài liệu HTML, tuy nhiên với các tài liệu HTML thường không thuận tiện cho yêu cầu này bởi vì HTML chỉ có ưu điểm trong việc thể hiện dữ liệu hơn là trình bày ý nghĩa dữ liệu đó. XML là một định dạng cơ bản nhất cho việc trình bày dữ liệu, bởi vì XML có thể được sử dụng để trình bày ý nghĩa dữ liệu được vận chuyển, và hơn thế nữa, hiện tại đa số các ứng dụng chạy trên nền Web-Base đều hỗ trợ các bộ phân tích cú pháp XML.

SOAP là công nghệ chủ yếu được sử dụng tại tầng này, nó là một giao thức đóng gói dữ liệu phổ biến dựa trên nền tảng XML. Chúng ta sẽ đề cập sâu hơn đến giao thức đóng gói dữ liệu SOAP trong phần “Các công nghệ của Web Service” trong chương 2 của khóa luận này.

Tầng Transport : Tầng Transport có vai trò đảm nhiệm việc vận chuyển các Web Service Message, tại đây bao gồm một vài dạng công nghệ khác nhau cho phép các giao tiếp trực tiếp giữa các Application – to – Application dựa trên tầng Network. Mỗi công nghệ bao gồm các giao thức như tcp, http, smtp và jabber..v.v.

Việc lựa chọn giao thức vận chuyển được dựa trên mỗi nhu cầu giao tiếp của các Web Service. ví dụ: với giao thức HTTP là một giao thức vận chuyển khá phổ biến được sử dụng cho các ứng dụng Web-Base, nhưng nó không cung cấp cơ chế giao tiếp bất đối xứng. Jabber, xét trên phương diện khác, nó không phải là một chuẩn nhưng có khả năng cung cấp tốt các kênh giao tiếp bất đối xứng.

Tầng Network : Tầng Network trong công nghệ Web Service chính xác giống tầng Network trong mô hình giao thức TCP/IP. Nó cung cấp khả năng giao tiếp cơ bản, định địa chỉ và định tuyến

3. Kiến trúc hướng dịch vụ SOA

3.1. Khái niệm kiến trúc hướng dịch vụ SOA

SOA - viết tắt của thuật ngữ Service Oriented Architecture (kiến trúc hướng dịch vụ) là “Khái niệm về hệ thống trong đó mỗi ứng dụng được xem như một nguồn cung cấp dịch vụ”.

Dịch vụ là yếu tố then chốt trong SOA. Có thể hiểu dịch vụ như là hàm chức năng (module phần mềm) thực hiện quy trình nghiệp vụ nào đó, một cách cơ bản, SOA là tập hợp các dịch vụ kết nối mềm dẻo với nhau (nghĩa là một ứng dụng có thể nói chuyện với một ứng dụng khác mà không cần biết các chi tiết kỹ thuật bên trong), có giao tiếp (dùng để gọi hàm dịch vụ) được định nghĩa rõ ràng và độc lập với nền tảng hệ thống, và có thể tái sử dụng. SOA là cấp độ cao hơn của phát triển ứng dụng, chú trọng đến quy trình nghiệp vụ và dùng giao tiếp chuẩn để giúp che đi sự phức tạp của kỹ thuật bên dưới.

Thiết kế SOA tách riêng phần thực hiện dịch vụ (phần mềm) với giao tiếp gọi dịch vụ. Điều này tạo nên một giao tiếp nhất quán cho ứng dụng khách sử dụng dịch vụ bất chấp công nghệ thực hiện dịch vụ. Thay vì xây dựng các ứng dụng đơn lẻ và đồ sộ, nhà phát triển sẽ xây dựng các dịch vụ có tính linh hoạt có thể triển khai và tái sử dụng trong toàn bộ quy trình nghiệp vụ. Điều này cho phép tái sử dụng phần mềm tốt hơn,

cũng như tăng sự linh hoạt vì nhà phát triển có thể cải tiến dịch vụ mà không làm ảnh hưởng đến Client sử dụng dịch vụ.

Thực ra khái niệm SOA không hoàn toàn mới, DCOM và CORBA cũng có kiến trúc tương tự. Tuy nhiên các kiến trúc cũ ràng buộc các thành phần với nhau quá chặt, ví dụ các ứng dụng phân tán muốn làm việc với nhau phải đạt được thoả thuận về chi tiết tập hàm API, một thay đổi mã lệnh trong thành phần COM sẽ yêu cầu những thay đổi tương ứng đối với mã lệnh truy cập thành phần COM này.

Ưu điểm quan trọng nhất của SOA là khả năng kết nối mềm dẻo (nhờ sự chuẩn hoá giao tiếp) và tái sử dụng. Các dịch vụ có thể được sử dụng với trình Client chạy trên nền tảng bất kì và được viết bởi ngôn ngữ bất kì.

3.2. Nguyên tắc thiết kế của SOA

SOA dựa trên hai nguyên tắc thiết kế quan trọng:

Mô-đun: đó là tách các vấn đề lớn thành nhiều vấn đề nhỏ hơn

Đóng gói : Che đi dữ liệu và lô-gic trong từng mô-đun đối với các truy cập từ bên ngoài.

Hai tính chất này sẽ dẫn đến đặc điểm thiết kế của kiến trúc SOA đó là các dịch vụ tương tác với nhau qua các thành phần giao tiếp, tuy nhiên các dịch vụ đó vẫn hoạt động độc lập với nhau, chia sẻ các lược đồ dữ liệu cho nhau và tuân thủ các chính sách của kiến trúc chung nhất.

Bài tập:

Bài tập nâng cao:

VPS là loại mạng gì? Trình bày ưu điểm và khuyết điểm?

Những trọng tâm cần chú ý trong bài:

- Lợi ích khi kết nối các máy tính thành mạng
- Các loại mạng thường gặp và đặc trưng của nó

Yêu cầu về đánh giá kết quả học tập:

Nội dung:

+ Về kiến thức: Trình bày được các lợi ích khi kết nối các máy tính thành mạng, vì sao mạng chuyển mạch gói có tốc độ trao đổi thông tin nhanh hơn tốc độ trao đổi thông tin trong mạch chuyển mạch tin báo?, thế nào là mạng cục bộ LAN (Local Area Networks) và nêu các đặc trưng cơ bản của nó.

+ Về kỹ năng: phân biệt được mạng cục bộ, mạng đô thị, mạng diện rộng.

+ Năng lực tự chủ và trách nhiệm: Tỉ mỉ, cẩn thận, chính xác, ngăn nắp trong công việc.

Phương pháp:

+ Về kiến thức: Được đánh giá bằng hình thức kiểm tra viết, trắc nghiệm, vấn đáp

+ Về kỹ năng: Đánh giá kỹ năng phân biệt được các loại mạng.

+ Năng lực tự chủ và trách nhiệm: Tỉ mỉ, cẩn thận, chính xác, ngăn nắp trong công việc.

BÀI 3: XÂY DỰNG WEB SERVICE

Mã bài: MĐ 26 - 03

Giới thiệu:

Trong bài này trình bày các vấn đề cần xác định trước khi xây dựng ứng dụng webservice, cách xây dựng webservice với ASP.NET và Java

Mục tiêu của bài:

- Trình bày được các vấn đề cần xác định rõ trước khi bắt tay xây dựng ứng dụng Web service.
- Xây dựng được Web Service với ASP.NET.
- Xây dựng được Web Service với Java.
- Thực hiện các thao tác an toàn với máy tính.

Nội dung chính:

1. Các vấn đề cần xác định rõ trước khi bắt tay xây dựng ứng dụng Web service

Có 4 giai đoạn chính để xây dựng một dịch vụ Web là xây dựng, triển khai, tiến hành và quản lý, trong đó:

- Giai đoạn xây dựng bao gồm phát triển và chạy thử ứng dụng dịch vụ Web, xây dựng các chức năng và định nghĩa dịch vụ. Có hai cách khác nhau để tiến hành trong giai đoạn này, đó là Red-path- solid và Blue-path-dashed. Với Red- path-solid, chúng ta sẽ xây dựng một dịch vụ Web mới từ trạng thái ban đầu hoặc với một dịch vụ đã có sẵn. Từ đó, xây dựng định nghĩa service (WSDL) với các đối tượng, hàm chức năng mà chúng ta mong muốn. Nếu theo cách Blue-path-dashed, dịch vụ Web sẽ được xây dựng từ đầu hoặc từ một định nghĩa dịch vụ WSDL. Sử dụng WSDL này, xây dựng hoặc sửa đổi lại mã để thực hiện các yêu cầu mong muốn trong dịch vụ Web.

- Giai đoạn triển khai: công bố định nghĩa dịch vụ, xây dựng WSDL và triển khai mã thực thi của dịch vụ Web. Triển khai dịch vụ Web tới một ứng dụng phía server, sau đó sẽ công bố dịch vụ Web trên mạng Internet để các client có thể nhìn thấy. Sử dụng UDDI registry để công bố lên mạng.

- Giai đoạn tiến hành: tìm kiếm và gọi thực thi dịch vụ Web bởi những người dùng muốn sử dụng dịch vụ.

Quản lý: Quản lý và quản trị dịch vụ, duy trì sự ổn định của dịch vụ, cập nhật thông tin mới, sửa lỗi khi nó xảy ra...

Để xây dựng một dịch vụ Web, chúng ta cần hiểu được những việc phải làm và nên bắt đầu từ đâu. Có 3 cách tiếp cận chủ yếu để xây dựng nên một dịch vụ Web, có thể từ một ứng dụng đã có (bottom-up); từ một định nghĩa dịch vụ, WSDL để phát sinh một ứng dụng mới (top-down) hoặc có thể từ một nhóm các dịch vụ Web hiện có, kết

hợp lại với nhau để tạo nên các chức năng mới hoặc mở rộng thêm chức năng. Những hướng tiếp cận này dựa trên những gì mà chúng ta đã có, tùy thuộc vào yêu cầu của hệ thống, trong đó tối đa việc sử dụng lại các chức năng, các thành phần, môđun đã được xây dựng.

Quy trình xây dựng một dịch vụ Web bao gồm các bước sau:

- Định nghĩa và xây dựng các chức năng, các dịch vụ mà dịch vụ sẽ cung cấp (sử dụng ngôn ngữ Java chẳng hạn).
- Tạo WSDL cho dịch vụ
- Xây dựng SOAP server
- Đăng ký WSDL với UDDI registry để cho phép các client có thể tìm thấy và truy xuất server
- Client nhận file WSDL và từ đó xây dựng SOAP client để có thể kết nối với SOAP
- Xây dựng ứng dụng phía client (chẳng hạn sử dụng Java) và sau đó gọi thực hiện dịch vụ thông qua việc kết nối tới SOAP server.

Lựa chọn một ngôn ngữ, xây dựng các tiến trình nghiệp vụ và chúng ta bắt đầu tạo nên một dịch vụ Web như ý muốn. Sau đó là cung cấp dịch vụ Web này trên Internet.

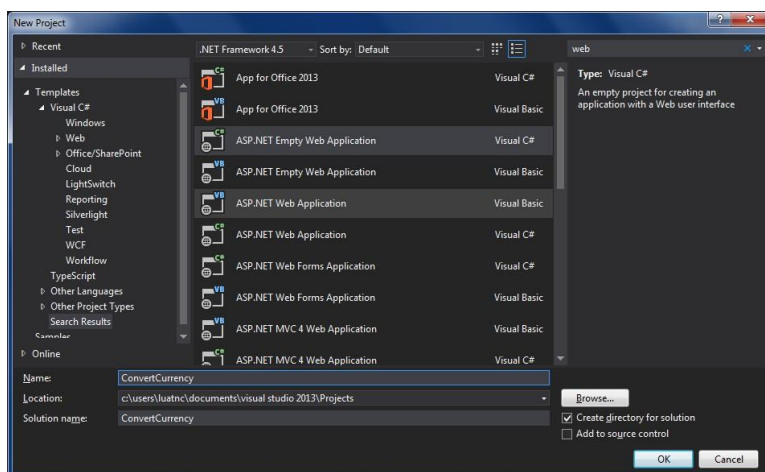
2. Xây dựng Web Service với ASP.NET

Hiện thực hóa bài học trước, ta sẽ cùng xây dựng 1 ứng dụng nhỏ cho Web Service trên nền tảng ASP.NET cho phép chuyển đổi tiền tệ từ Việt Nam đồng sang USD hoặc Euro và ngược lại.

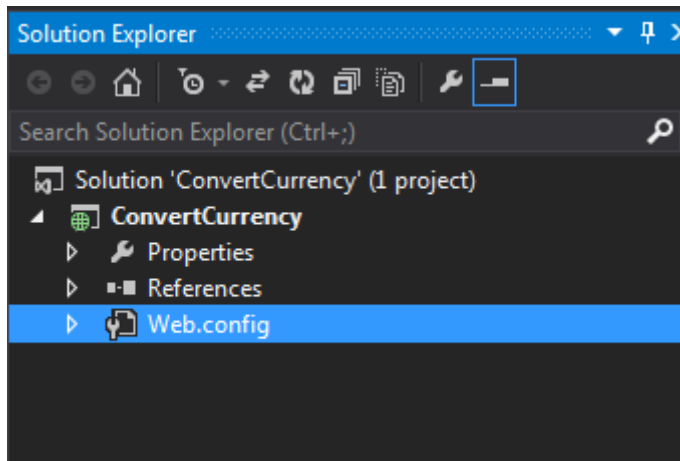
Công cụ sử dụng bao gồm: Visual Studio và ngôn ngữ lập trình ASP.NET, C#.

a. Tạo mới Web Application Project

Chọn menu File > New > Project trong Visual Studio, chọn mục ASP.NET Web Application và đặt tên ConvertCurrency, chọn thư mục lưu trữ project và click OK

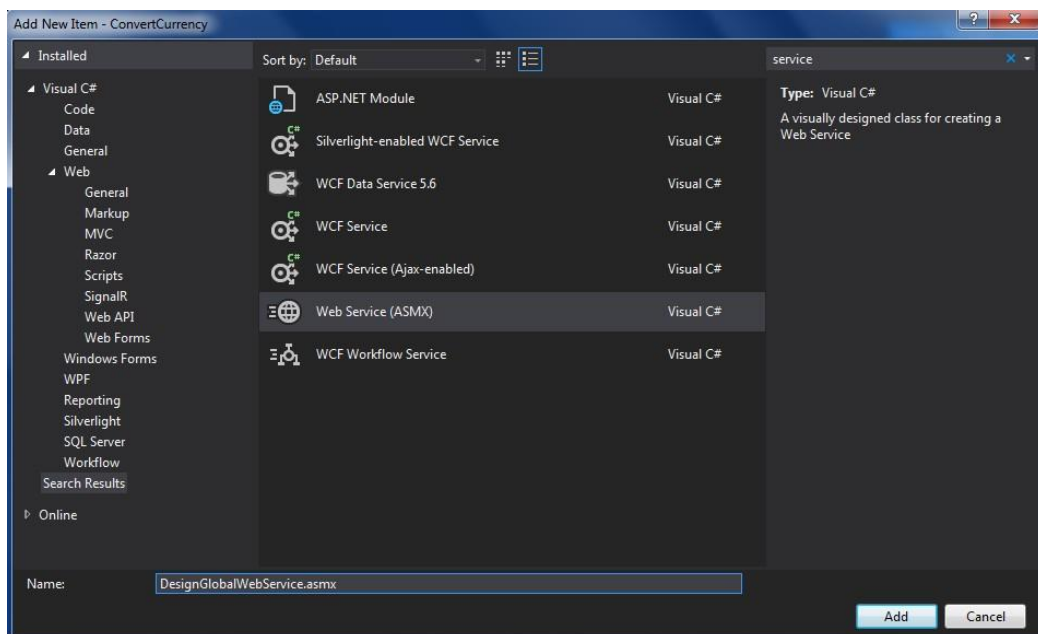


Kết quả, ta sẽ thấy trong cửa sổ Solution Explorer như hình sau

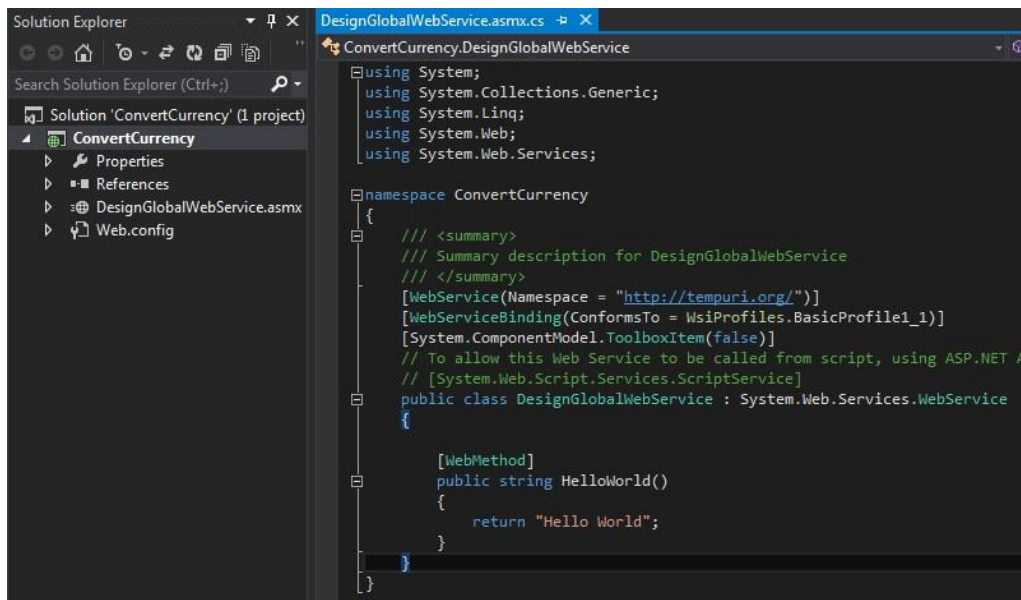


b. Tạo mới Web Service

Click chuột phải vào project và chọn Add new > New item và chọn như hình sau



Chọn mục Web Service (ASMX) và đặt tên DesignGlobalWebService.asmx và click Add, ta sẽ thấy kết quả như hình sau



Nội dung mặc định sẽ được Visual Studio tạo ra như trên, ta tiến hành chỉnh sửa lại Web Service để cung cấp các dịch vụ như mong muốn như sau

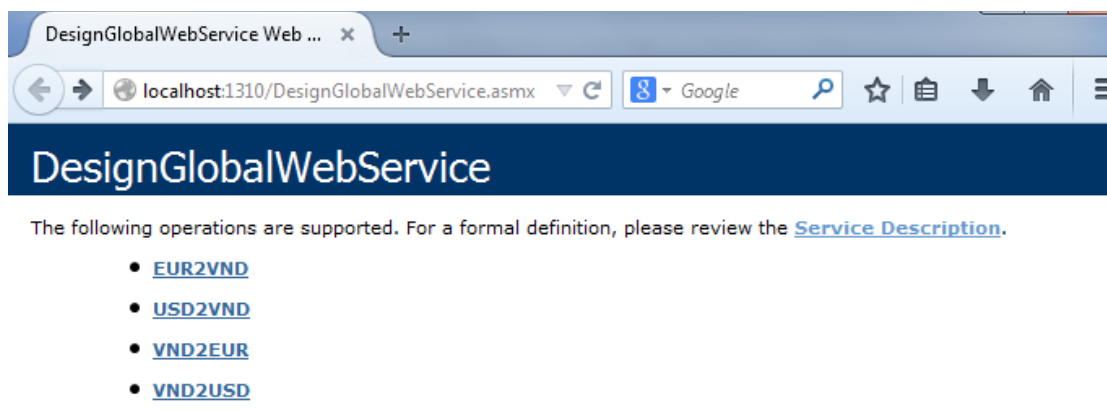
```
using System;
using System.Collections;
using System.ComponentModel; using System.Data;
using System.Web;
using System.Web.Services;
using System.Web.Services.Protocols;
namespace ConvertCurrency
{
    /// <summary>
    /// Web service này dùng chuyển đổi ngoại tệ từ tiền Việt sang USD,
    /// Euro và ngược lại
    /// </summary>
    [WebService(Namespace = "http://tempuri.org/")] [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)] [ToolboxItem(false)]
    // To allow this Web Service to be called from script, using ASP.NET AJAX
    // [System.Web.Script.Services.ScriptService]
    public class ConvertWS : System.Web.Services.WebService
    {
        private const double USD_RATE = 20000; private const double EUR_RATE = 30000; [WebMethod]
        public double VND2USD(double dong)
```

```

    {
        return dong/USD_RATE;
    }
    [WebMethod]
    public double VND2EUR(double dong)
    {
        return dong / EUR_RATE;
    }
    [WebMethod]
    public double USD2VND(double usd)
    {
        return usd * USD_RATE;
    }
    [WebMethod]
    public double EUR2VND(double eur)
    {
        return eur * EUR_RATE;
    }
}
}

```

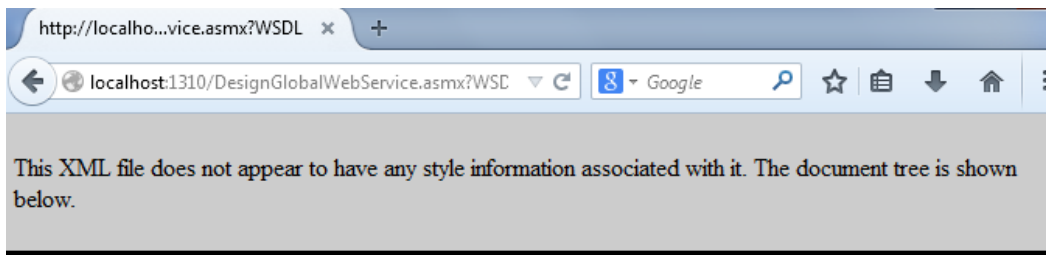
Chạy ứng dụng bằng cách nhấn F5, VS sẽ triển khai ứng dụng. Kết quả như sau



This web service is using <http://tempuri.org/> as its default namespace.

Recommendation: Change the default namespace before the XML Web service is made public.

Kiểm tra WSDL của service, ta nhấn link “Service Description” ta có



```

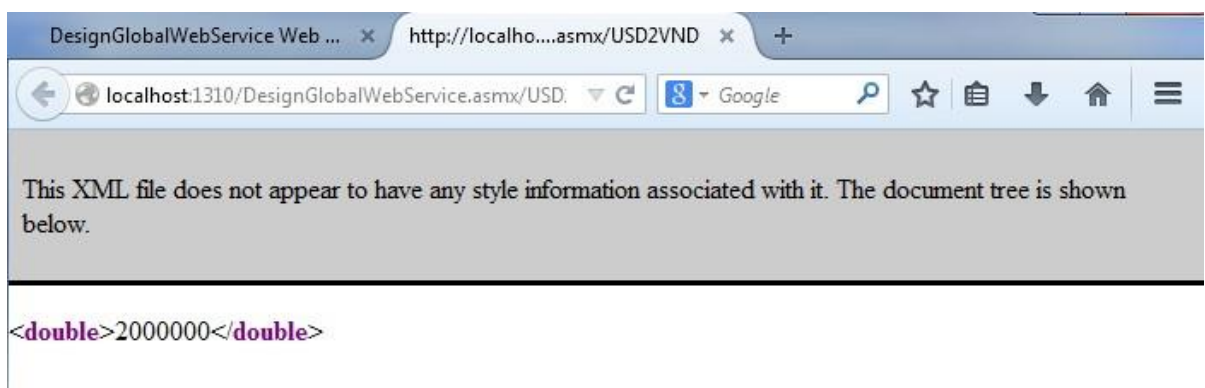
- <wsdl:definitions targetNamespace="http://tempuri.org/">
- <wsdl:types>
- <s:schema elementFormDefault="qualified" targetNamespace="http://tempuri.org/">
- <s:element name="VND2USD">
- <s:complexType>
- <s:sequence>
  <s:element minOccurs="1" maxOccurs="1" name="dong" type="s:double"/>
</s:sequence>
</s:complexType>
</s:element>
- <s:element name="VND2USDResponse">
- <s:complexType>
- <s:sequence>
  <s:element minOccurs="1" maxOccurs="1" name="VND2USDResult" type="s:double"/>
</s:sequence>
</s:complexType>
</s:element>

```

Để thử service, ta có thể chọn bất kỳ link nào trong 4 link EUR2VND, USD2VND VND2EUR, VND2USD. Ở đây ta thử link USD2VND, kết quả như sau



Nhập usd có giá trị 100, nhấn Invoke, kết quả nhận được là



Vậy Web Service đã hoạt động và cung cấp trên Internet để các dịch vụ khách có thể truy vấn và sử dụng

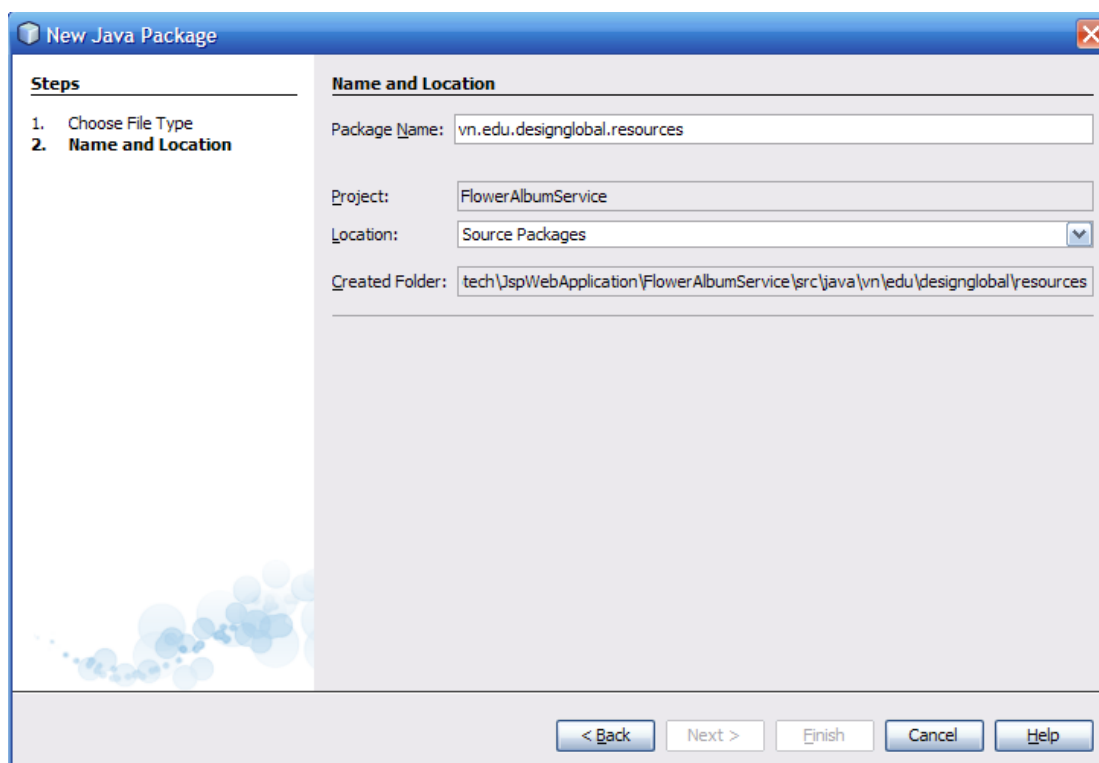
3. Xây dựng Web Service với Java

Công cụ sử dụng bao gồm: Netbeans 7.4 và ngôn ngữ lập trình Java.

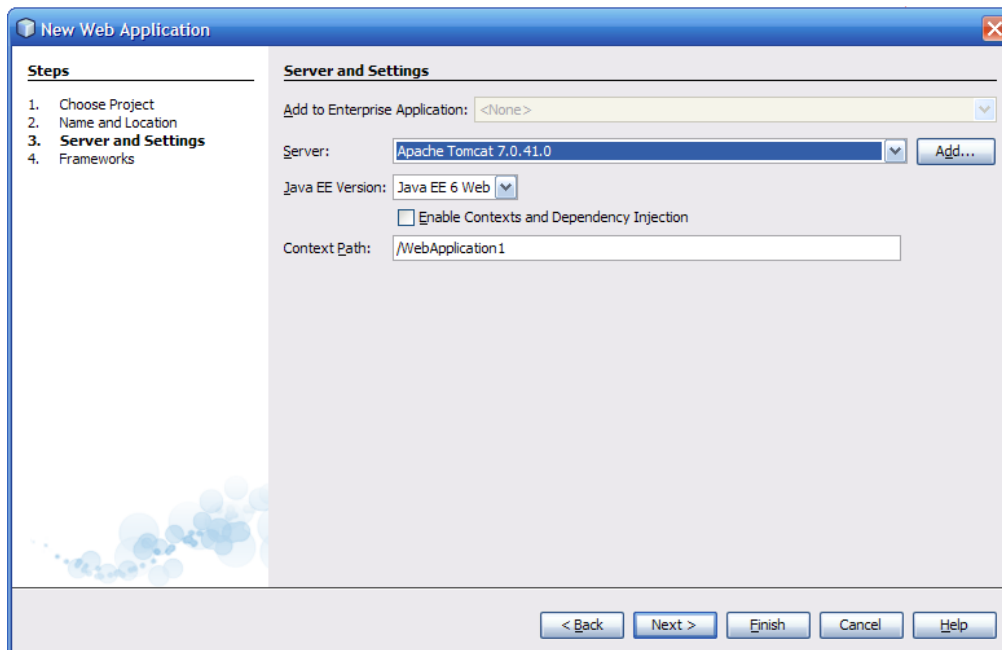
a. Tạo mới JSP Web Application

Mở menu của Netbeans, chọn File > New Project, ta chọn mục Java Web và chọn Web Application ở cửa sổ bên cạnh, chọn Next

Đặt tên FlowerAlbumService cho project, chọn thư mục lưu giữ project. Để các tùy chọn khác mặc định và chọn Next.

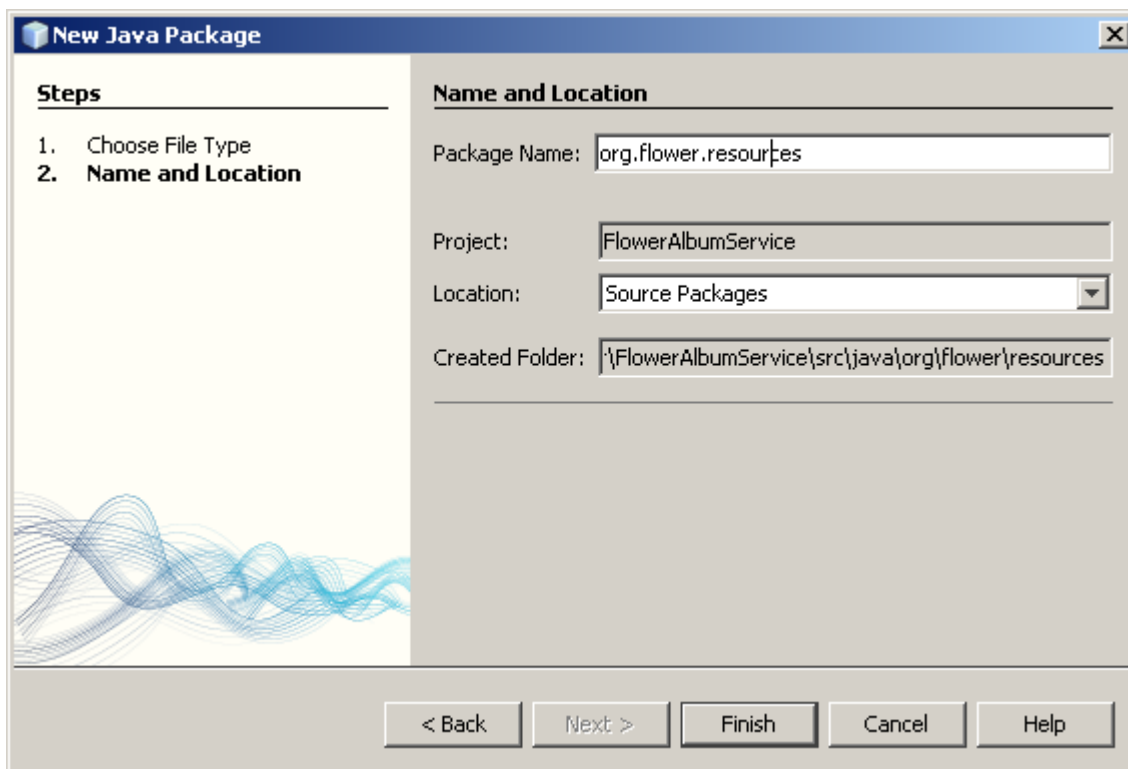


Cửa sổ Server and Settings mở ra, ta chọn GlassFish server and Java EE version Java EE 6 Web hoặc Java EE 7 Web. Chọn Finish, FlowerAlbumService project được tạo và trong cửa sổ Project ta thấy như sau



b. Thêm Resource vào project

Chọn chuột phải vào Source Packages và chọn New > Java Package



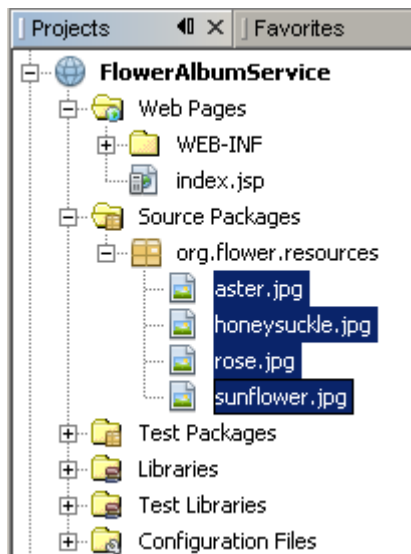
Đặt tên vn.edu.designglobal.resources và chọn Finish

Ta copy các ảnh sau (tìm trong phần tài nguyên của giáo trình) vào trong mục Resources

- □ rose.jpg
- □ sunflower.jpg

- aster.jpg
- honeysuckle.jpg

Ta sẽ thấy như hình sau



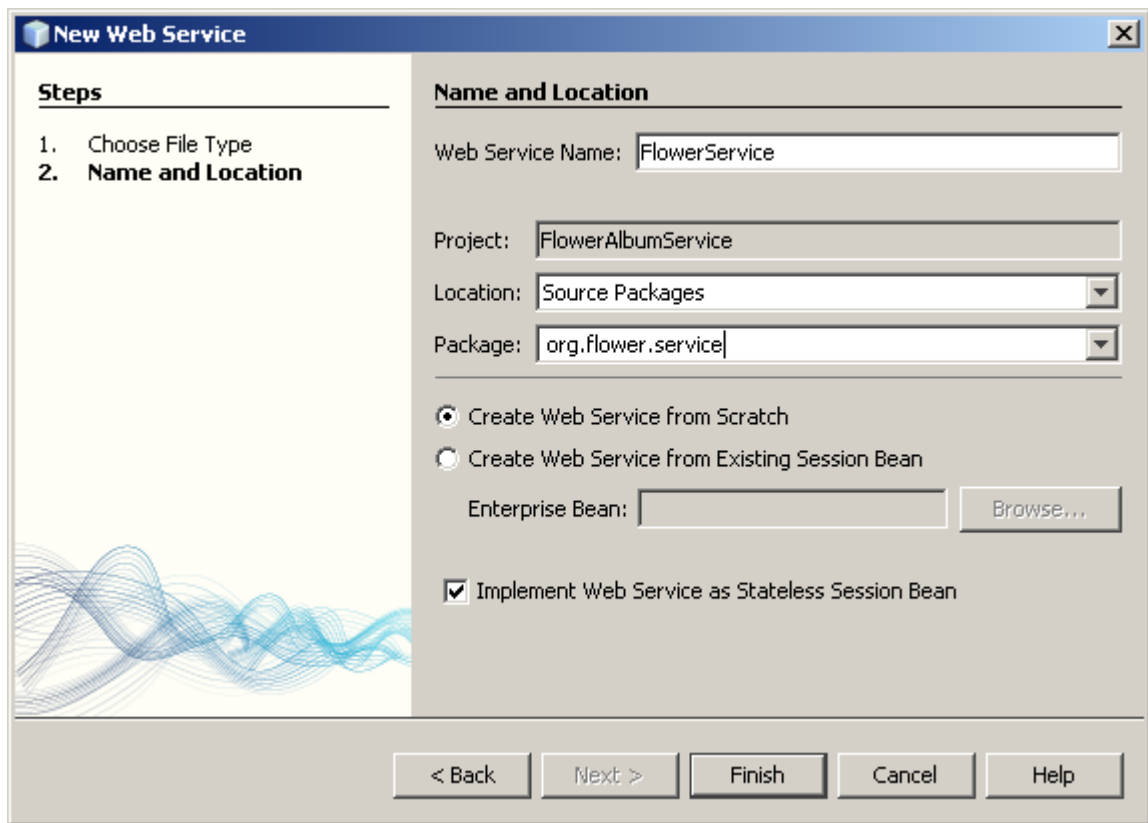
c. Thêm mới Web Service

Trong phần này, ta sẽ tạo Web service kèm với thuộc tính Stateless session bean, theo các bước chính sau

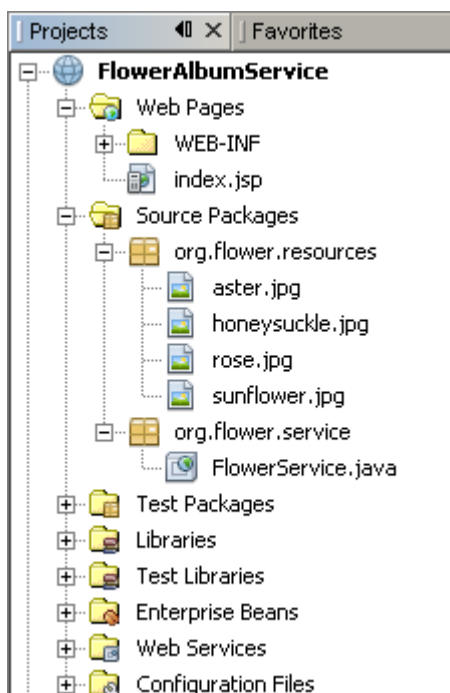
- Tạo hàm cho phép lấy tên và đối tượng Image tương ứng
- Tạo hàm cho phép lấy Thumbnails của tất cả danh mục Image và trả về thông qua đối tượng List

Ta thực hiện trình tự các bước sau:

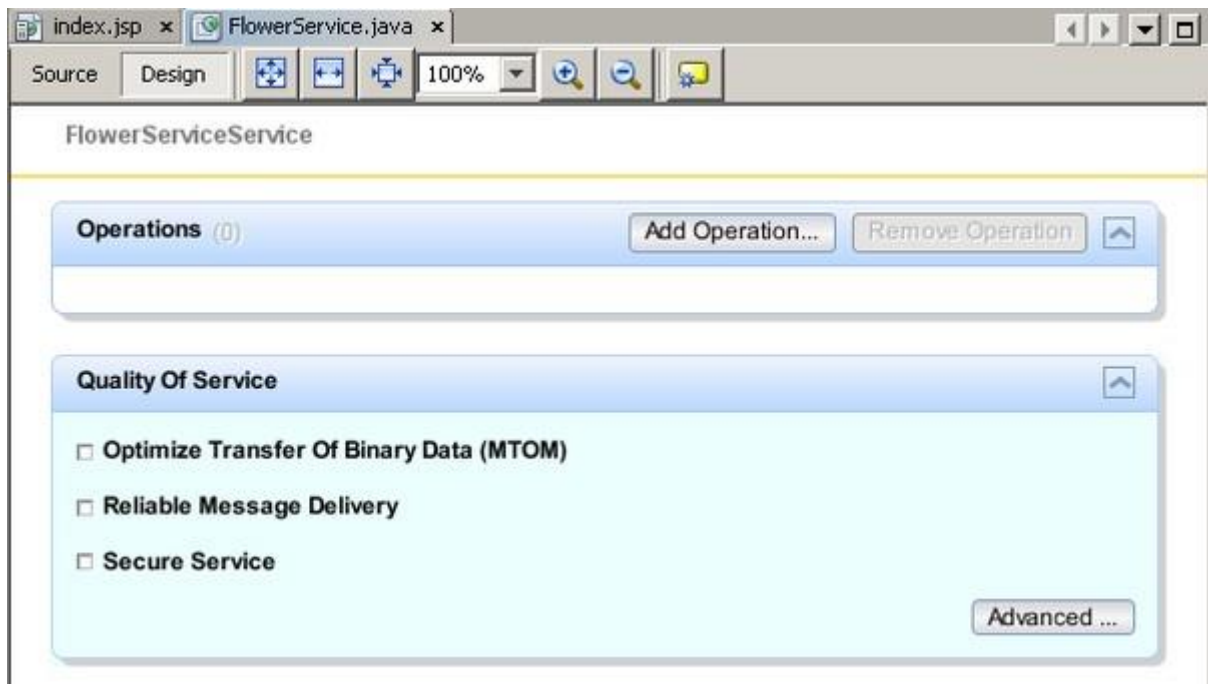
Chọn chuột phải vào project FlowerAlbumService, chọn New > Web Service. Tại cửa sổ mở ra, ta đặt tên FlowerService, chọn package vn.edu.designglobal.services. Chọn Create Web Service from Scratch và chọn Implement Service as Stateless Session Bean. Click chọn Finish để hoàn tất.



Sau khi tạo xong, ta sẽ thấy Web service xuất hiện trong cửa sổ project như hình sau

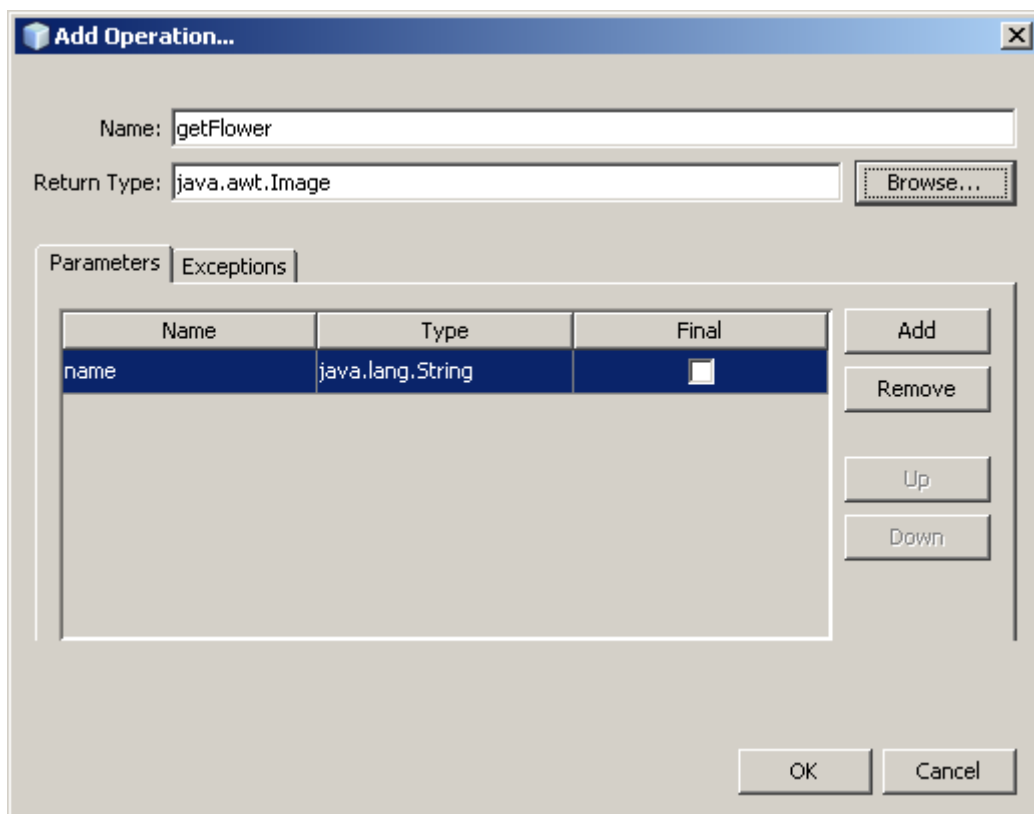


Mở file FlowerService.java trong cửa sổ soạn thảo của Netbeans, ta chuyển sang chế độ xem Design. Một cửa sổ xuất hiện cho phép ta thêm các hàm và theo dõi chất lượng (QoS) cho Web Service.

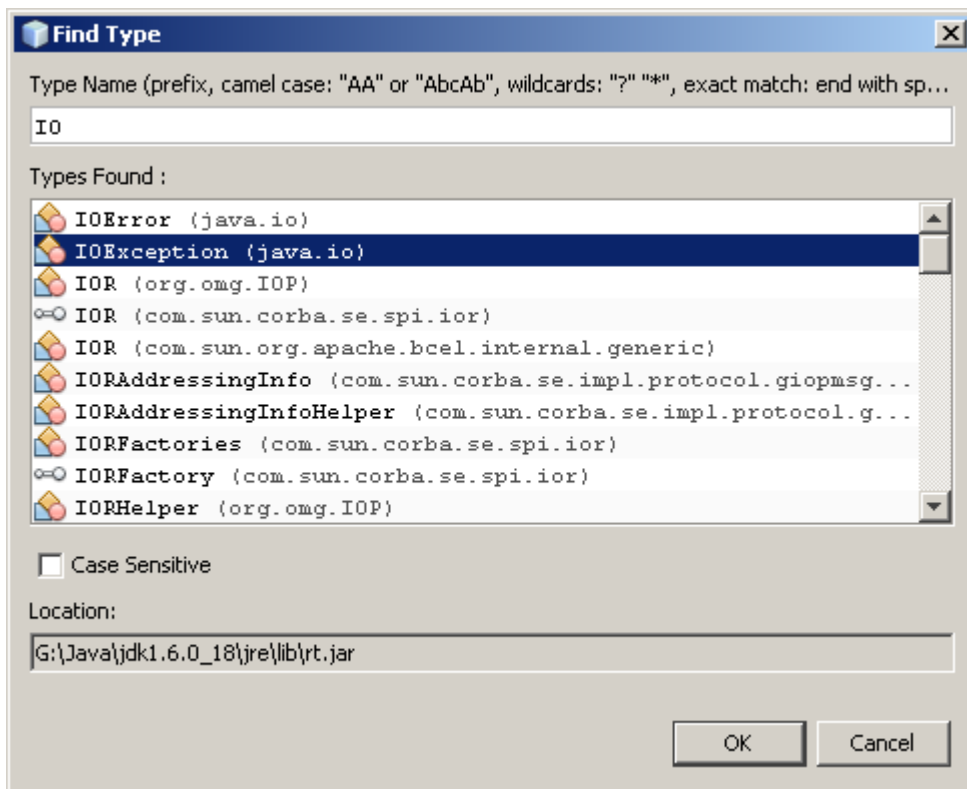


Chọn Add Operation để thêm mới các hàm cung cấp bởi dịch vụ.

- Đặt tên getFlower cho hàm
- Chọn kiểu dữ liệu trả về java.awt.Image.
- Bên thẻ Parameters, chọn Add, đặt tên name cho tham số và chọn kiểu dữ liệu java.lang.String mặc định. Xem hình sau



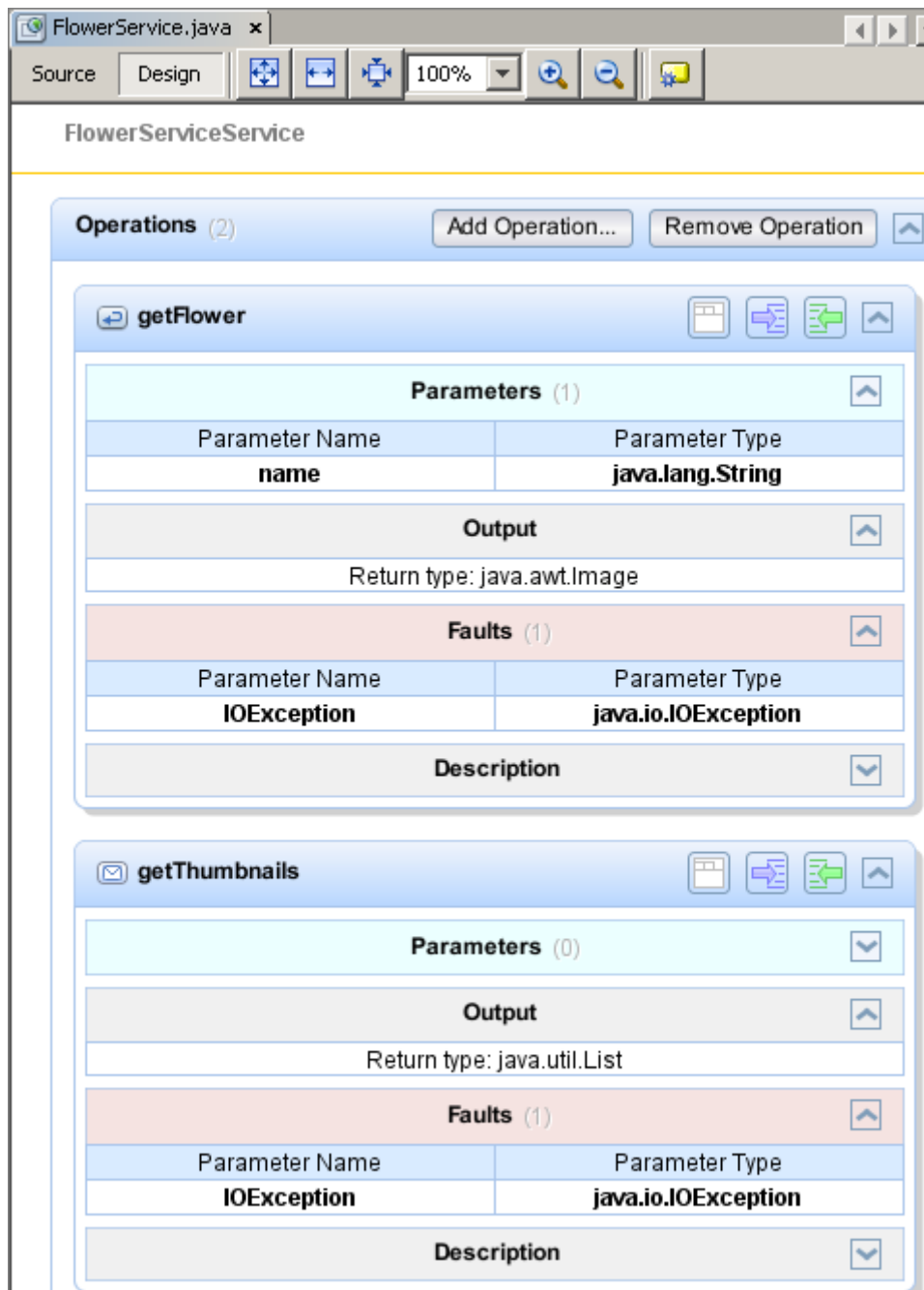
Chọn thẻ Exceptions, thêm khai báo ngoại lệ IOException cho hàm



Chọn OK và đóng cửa sổ, tương tự, ta sẽ thêm khai báo hàm cho Web service như sau

- Name: getThumbnails
- Return type: java.util.List
- Exception: IOException

Cửa sổ Design hiển thị các hàm getFlower, getThumbnails và khi chuyển sang chế độ xem List view ta sẽ thấy như sau



d. Xây dựng hàm cho Web Service

Chuyển sang chế độ Source View, ta thêm khai báo sau vào file FlowerService.java mã nguồn như sau

```
private byte[] getFlowerBytes(String name) throws IOException {
    URL resource =
this.getClass().getResource("/org/flower/resources/"+name+".jpg"); return
getBytes(resource);
}
```

Hàm getFlower trả về Image theo tên dịch vụ khách yêu cầu

Hàm nhận đầu vào là tên ảnh và tìm tới đường dẫn chứa ảnh trong thư mục chứa của Web Service để trả về đối tượng Image tương ứng dưới định dạng mảng byte.

```
private byte[] getBytes(URL resource) throws IOException { InputStream in =
resource.openStream(); ByteArrayOutputStream bos = new ByteArrayOutputStream();
byte[] buf = new byte[1024];

for(int read; (read = in.read(buf)) != -1;) { bos.write(buf, 0, read);
}

return bos.toByteArray();
}
```

Hàm nhận đầu vào là đường dẫn ảnh vật lý trực tiếp, đọc file theo từng block 1024 byte và lưu vào đối tượng InputStream để trả về dưới định dạng mảng byte.

```
private Image getImage(byte[] bytes, boolean isThumbnail) throws IOException
{ ByteArrayInputStream bis = new ByteArrayInputStream(bytes);

Object source = bis; // File or InputStream

ImageInputStream iis = ImageIO.createImageInputStream(source); Iterator
readers = ImageIO.getImageReadersByFormatName("jpeg"); ImageReader reader =
(ImageReader) readers.next(); ImageReadParam param =
reader.getDefaultReadParam();

if (isThumbnail) { param.setSourceSubsampling(4, 4, 0, 0);
}

reader.setInput(iis, true); return reader.read(0, param);
}
```

Hàm sẽ tạo đối tượng ImageInputStream từ mảng byte đầu vào. Và sử dụng đối tượng Iterator để đăng ký danh sách các đối tượng xử lý ImageReader giải mã định dạng ảnh jpeg. Nếu hàm yêu cầu tạo ra định dạng Thumbnails (param.setSourceSubsampling(4, 4, 0, 0)) của ảnh thông qua biến isThumbnail. Cuối cùng, ImageReader sẽ xử lý đối tượng ImageInputStream để cho ra đối tượng Image được biểu diễn bởi mảng byte đầu vào.

Thực thi phương thức getFlower của Web Service như sau

```
@WebMethod(operationName = "getFlower")

public Image getFlower(@WebParam(name = "name") String name) throws
IOException { byte[] bytes = getFlowerBytes(name);

return getImage(bytes, false);
}
```


Từ đầu vào là tên ảnh, hệ thống sẽ tìm trong thư mục lưu trữ ảnh với tên tương ứng. Khi tìm thấy, đường dẫn ảnh sẽ được trả lại và hàm `getFlowersBytes` sẽ đọc và trả về nội dung ảnh trong mảng byte. Cuối cùng hàm `getImage` sẽ chuyển đổi mảng byte về đối tượng `Image` trả về cho dịch vụ khách yêu cầu.

Hàm `getThumbnails` trả về danh sách thumbnails ảnh có trên server

Ta thêm khai báo hằng số chứa mảng danh sách tên ảnh cung cấp bởi server

```
private static final String[] FLOWERS = {"aster", "honeysuckle", "rose",
"sunflower"};
```

Hàm `allFlowers` sẽ đọc danh sách `Image` ở trên từ server và trả về trong danh sách mảng byte.

```
private List allFlowers() throws IOException { List flowers = new ArrayList();
for (String flower:FLOWERS) {
    URL resource =
this.getClass().getResource("/org/flower/resources/"+flower+".jpg");
flowers.add(getBytes(resource));
}
return flowers;
}
```

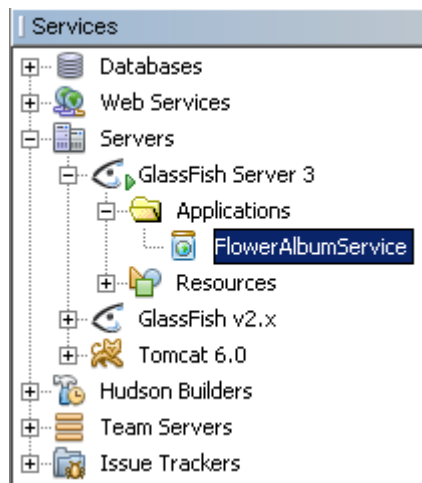
Hàm `getThumbnails` xử lý như sau

```
@WebMethod(operationName = "getThumbnails")
public List<Image> getThumbnails() throws IOException { List<byte[]> flowers
= allFlowers();
List<Image> flowerList = new ArrayList<Image>(flowers.size()); for (byte[]
flower : flowers) {
    flowerList.add(getImage(flower, true));
}
```

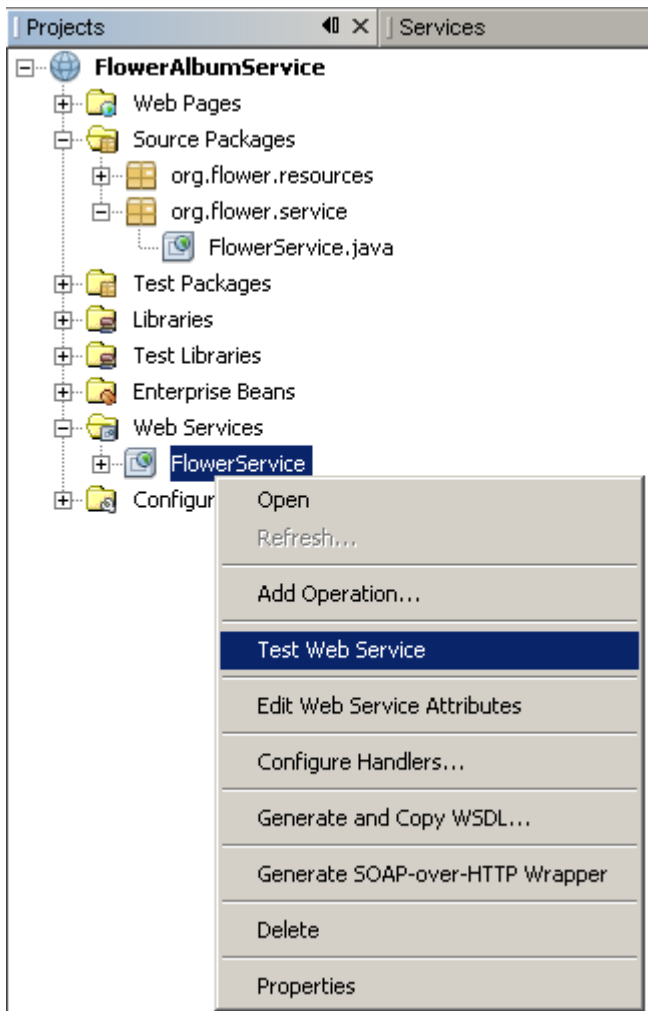
Từ danh sách mảng byte xử lý bởi hàm `allFlowers`, ta sẽ duyệt qua danh sách và chuyển đổi từng mảng byte thành đối tượng `Image` như ở phần trước, các đối tượng `Image` sẽ được lưu vào danh sách để trả về cho dịch vụ khách yêu cầu.

e.Publish và kiểm tra tính sẵn sàng phục vụ của Web Service

Chọn chuột phải vào `FlowerAlbumService` và chọn `Deploy`. Khi này, Netbeans sẽ khởi động Glashfish server và deploy ra file war của project tới Glashfish. Ta sẽ thấy kết quả như hình sau



Ta mở mục Web Services trong khung cửa sổ Project, chọn phải chuột vào FlowerService và chọn Test Web Service



Trên trình duyệt, ta thấy như sau và nhập “rose” vào ô tham số getFlower

The screenshot shows a web browser window with the title "FlowerServiceService Web Service Tester". The page content includes a heading, a description of the form's purpose, instructions on how to use it, and two method definitions with interactive buttons and input fields.

FlowerServiceService Web Service Tester

This form will allow you to test your web service implementation ([WSDL File](#))

To invoke an operation, fill the method parameter(s) input boxes and click on the button labeled with the method name.

Methods :

public abstract byte[] org.flower.service.FlowerService.getFlower(java.lang.String) throws org.flower.service.IOException_Exception
 ()

public abstract java.util.List org.flower.service.FlowerService.getThumbnails() throws org.flower.service.IOException_Exception
 ()

Ấn nút getFlower, ta sẽ thấy trên trình duyệt hiển thị như sau

Method returned

[B : "[B@2943e5"

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header/>
  <S:Body>
    <ns2:getFlower xmlns:ns2="http://service.flower.org/">
      <name>rose</name>
    </ns2:getFlower>
  </S:Body>
</S:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:getFlowerResponse xmlns:ns2="http://service.flower.org/">
      <return>iVBORwOKGgoAAAAANSUHEUgAAAoAAAAKACAIAAACDr150AACAAE
    </ns2:getFlowerResponse>
  </S:Body>
</S:Envelope>
```

Vậy Web Service đã sẵn sàng phục vụ

Bài tập:

Bài tập nâng cao:

VPS là loại mạng gì? Trình bày ưu điểm và khuyết điểm?

Những trọng tâm cần chú ý trong bài:

- Lợi ích khi kết nối các máy tính thành mạng
- Các loại mạng thường gặp và đặc trưng của nó

Yêu cầu về đánh giá kết quả học tập:

Nội dung:

+ Về kiến thức: Trình bày được các lợi ích khi kết nối các máy tính thành mạng, vì sao mạng chuyển mạch gói có tốc độ trao đổi thông tin nhanh hơn tốc độ trao đổi thông tin trong mạch chuyển mạch tin báo?, thế nào là mạng cục bộ LAN (Local Area Networks) và nêu các đặc trưng cơ bản của nó.

+ Về kỹ năng: phân biệt được mạng cục bộ, mạng đô thị, mạng diện rộng.

+ Năng lực tự chủ và trách nhiệm: Tỉ mỉ, cẩn thận, chính xác, ngăn nắp trong công việc.

Phương pháp:

+ Về kiến thức: Được đánh giá bằng hình thức kiểm tra viết, trắc nghiệm, vấn đáp

+ Về kỹ năng: Đánh giá kỹ năng phân biệt được các loại mạng.

+ Năng lực tự chủ và trách nhiệm: Tỉ mỉ, cẩn thận, chính xác, ngăn nắp trong công việc.

BÀI 4: KHAI THÁC WEB SERVICE

Mã bài: MD 26 - 04

Giới thiệu:

Trong bài này hướng dẫn tạo ứng dụng window form kết nối với webservice, ứng dụng java swing kết nối tới webservice và cuối cùng là web asp net kết nối với webservice

Mục tiêu của bài:

- Tạo được ứng dụng Window Form kết nối tới Web Service.
- Tạo được ứng dụng Java Swing kết nối tới Web Service.
- Tạo được ứng dụng Web ASP.NET kết nối tới Web Service.
- Thực hiện các thao tác an toàn với máy tính.

Nội dung chính:

1. Ứng dụng Window Form kết nối tới Web Service

a. Tạo mới ứng dụng Windows Forms Application

Tạo ứng dụng Windows Application bằng cách nhấn phải chuột lên Solution trong Solution Explorer, chọn Add > New Project

Chọn Template “Windows Form Application”, đặt tên là ConsumeConvertWS, nhấn OK

Nhấn chuột phải lên project vừa tạo, chọn “Set as Start Up Project” để đảm bảo project vừa tạo là project sẽ được thực thi khi nhấn F5.

b. Thêm tham chiếu tới Web Service vào project

Chọn chuột phải vào project, chọn “Add Web Reference...”. Gõ vào URL đến WSDL vào ô URL, nhấn nút Go

Đặt tên cho là ConvertCurrencyReference, ấn OK

c. Tạo giao diện ứng dụng đơn giản



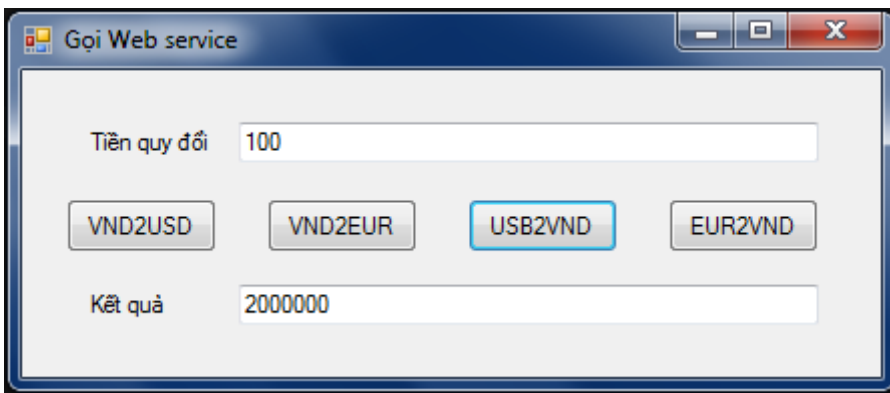
Đặt tên cho TextBox là txtMoney, Label hiển thị kết quả là txtKetQua. Thêm đoạn mã thực thi chương trình như sau

```
using System;
using System.Collections.Generic;
using System.ComponentModel; using System.Data;
using System.Drawing; using System.Text;
using System.Windows.Forms;
namespace ConsumeConvertWS
{
public partial class Form1 : Form
{
private ConsumeConvertWS.convertWS.ConvertWS ws; public Form1()
{
InitializeComponent();
ws = new ConsumeConvertWS.convertWS.ConvertWS();
}
private void VND2USD_Click(object sender, EventArgs e)
{
double mon = Double.Parse(txtMoney.Text); double usd = ws.VND2USD(mon);
lblKetQua.Text = usd.ToString();
}
private void VND2EUR_Click(object sender, EventArgs e)
{
double mon = Double.Parse(txtMoney.Text); double usd = ws.VND2EUR(mon);
lblKetQua.Text = usd.ToString();
}
private void USD2VND_Click(object sender, EventArgs e)
{
double mon = Double.Parse(txtMoney.Text); double usd = ws.USD2VND(mon);
lblKetQua.Text = usd.ToString();
}
}
```

```
private void EUR2VND_Click(object sender, EventArgs e)
{
    double mon = Double.Parse(txtMoney.Text); double usd = ws.EUR2VND(mon);
    lblKetQua.Text = usd.ToString();
}
}
}
```

d.Kiểm tra hoạt động ứng dụng kết nối tới Web Service

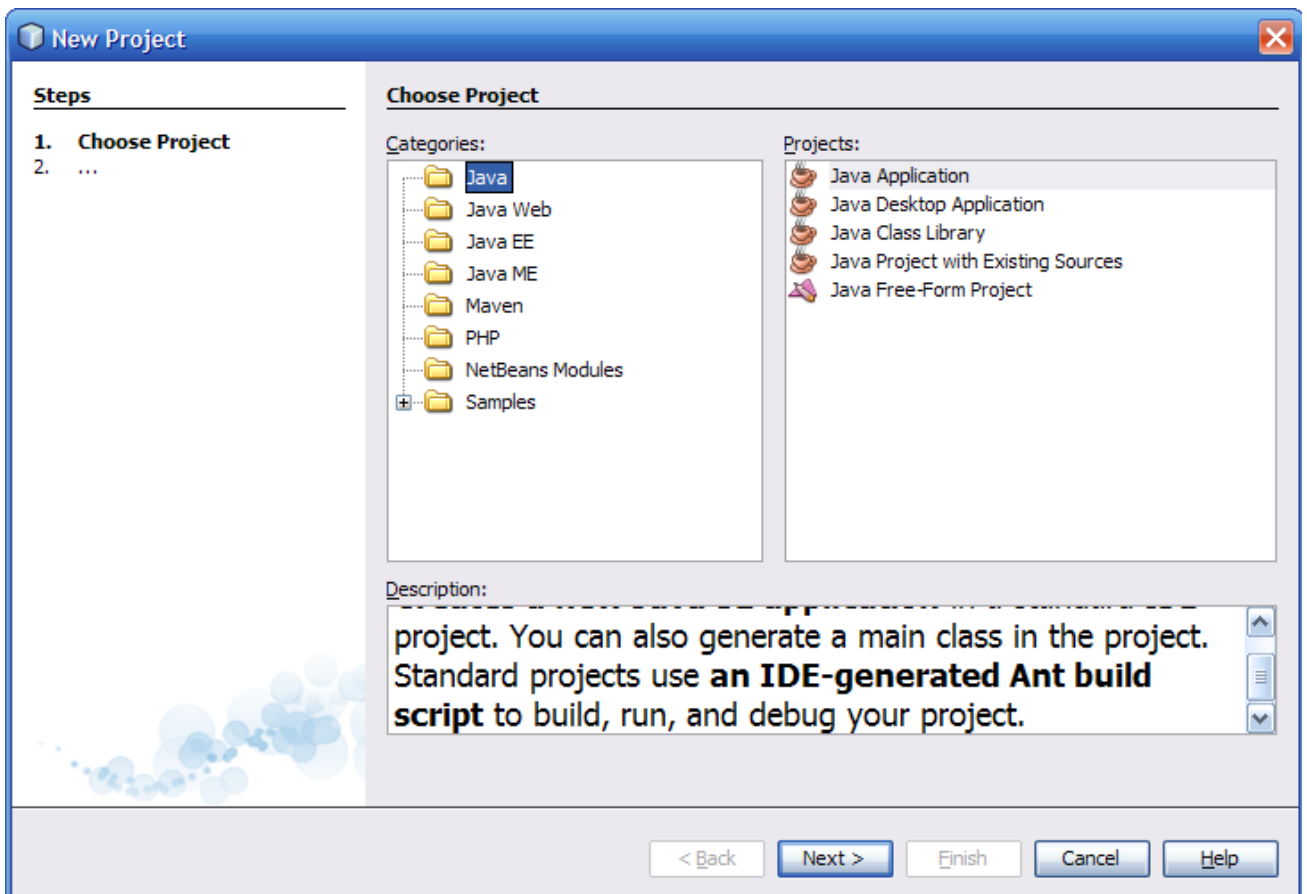
Ta chọn F5 để thực thi ứng dụng, nhập 100 vào ô Tiền quy đổi, click nút USB2VND. Kết quả, ta thấy như sau



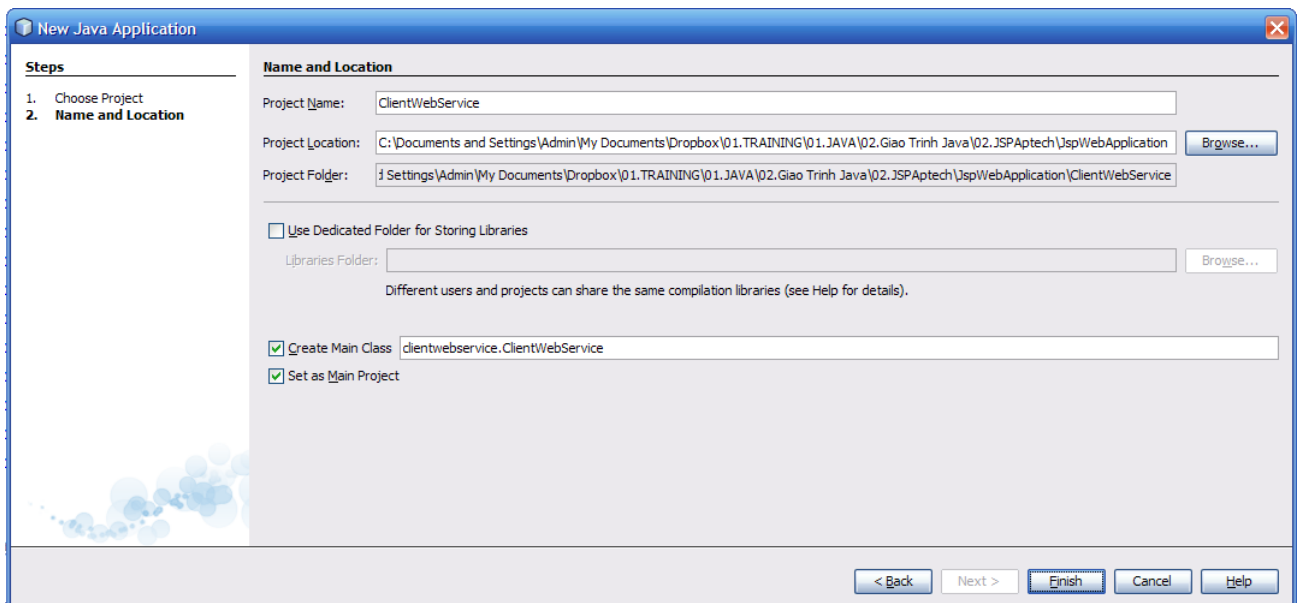
2. Ứng dụng Java Swing kết nối tới Web Service

a.Tạo mới ứng dụng Java Application

Chọn từ menu Netbeans, chọn File > New Project, chọn mục Java và Java Application bên cạnh

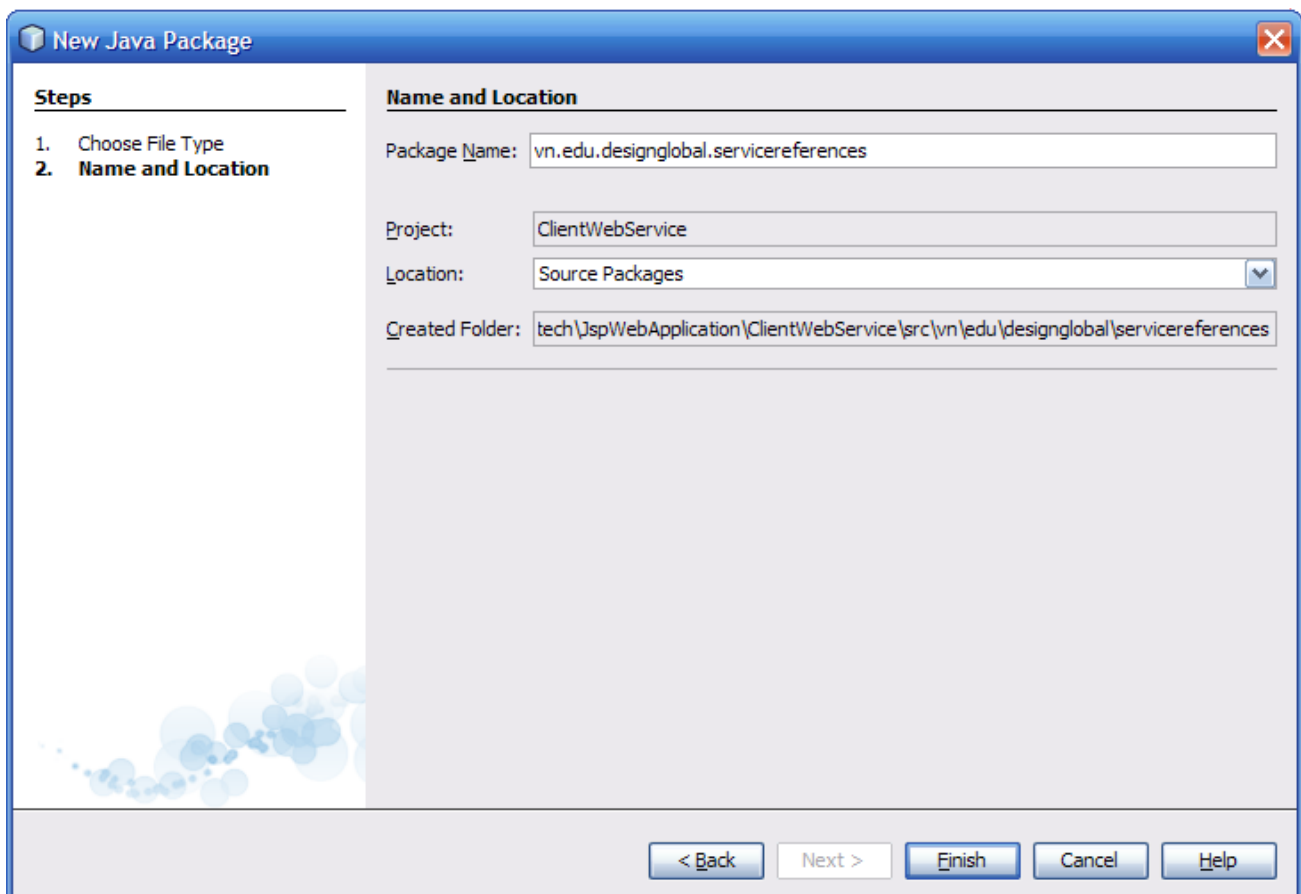


Cửa sổ New Java Application mở ra, ta đặt tên ClientWebService cho ứng dụng và thiết lập Set as Main Project.

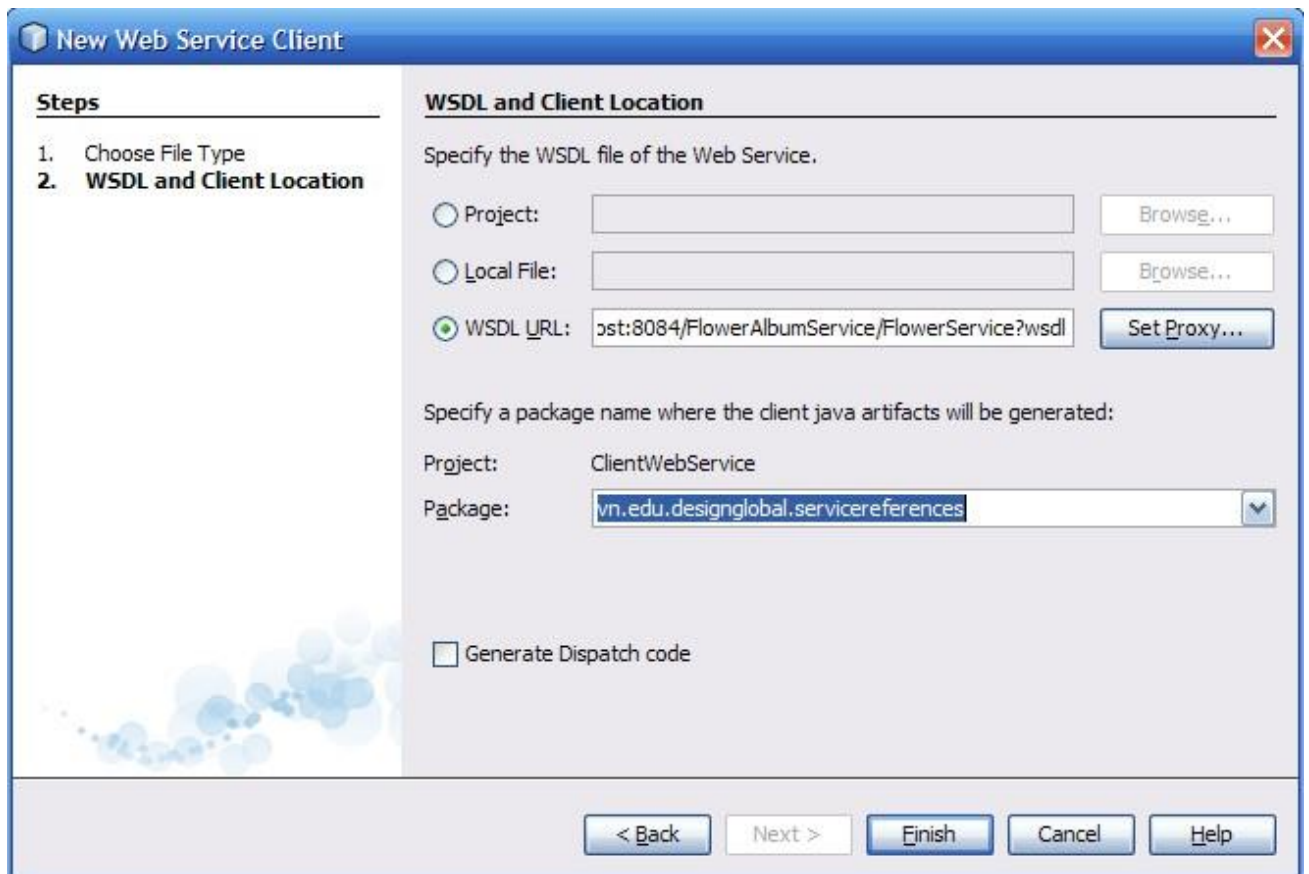


b. Thêm tham chiếu tới Web Service

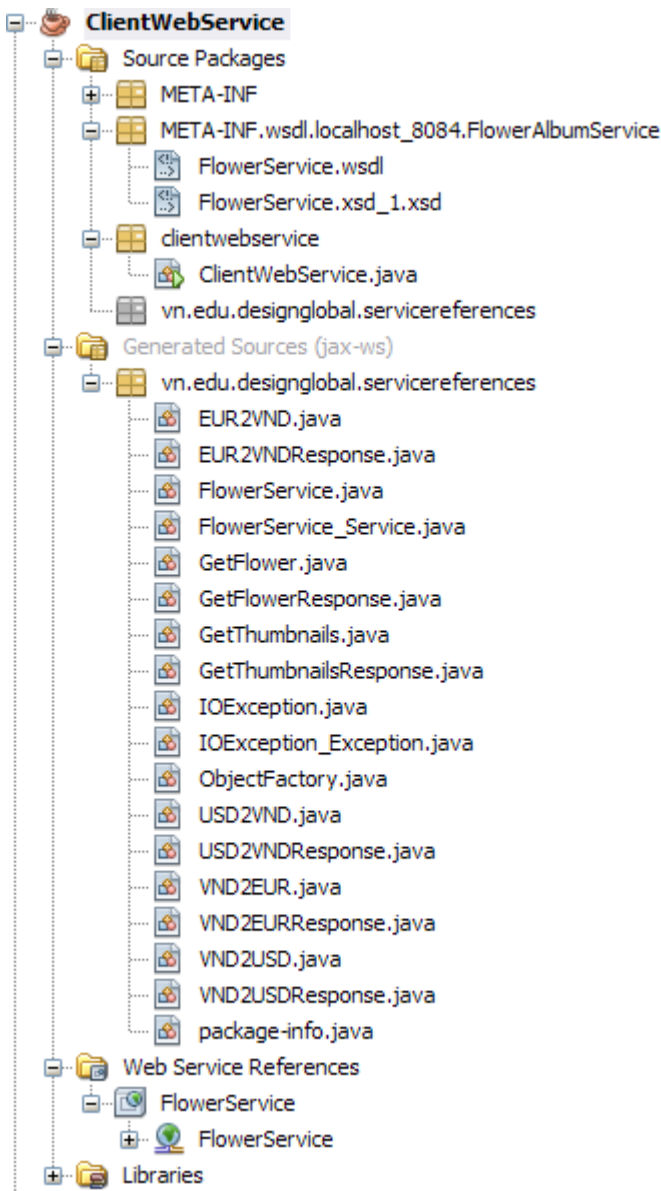
Ta tạo mới package cho project, chọn chuột phải vào project, chọn New > Java Package, cửa sổ mở ra như hình sau. Đặt tên vn.edu.designglobal.servicereferences cho package



Ta thêm tham chiếu tới Web Service cung cấp chức năng chuyển đổi tiền tệ ở trên. Chọn chuột phải vào project, chọn New > Web Service Client ; lựa chọn mục WSDL URL và nhập đường dẫn publish của Web Service ; chọn package để lưu giữ tham chiếu của Web Service như hình sau



Sau khi thêm thành công ta sẽ thấy như hình sau



c. Thiết kế giao diện ứng dụng

Thiết kế giao diện ứng dụng như hình sau và F5 để kiểm tra hoạt động ứng dụng, ta sẽ thấy kết quả như sau



Đoạn mã nguồn thực thi chương trình như bên dưới

```
public class ClientWebService extends JFrame implements ActionListener {
    private static final long serialVersionUID = 1L;
    private JTextField tfMoney, tfDisplay;
    private JButton btnVND2USD, btnVND2EUR, btnUSD2VND, btnEUR2VND;
```

```

public ClientWebService()
{
    setTitle("Gọi Web service");
    setDefaultCloseOperation(EXIT_ON_CLOSE); setSize(450, 150);
    setResizable(false);

    JLabel lblTitle = new JLabel("CHUYỂN ĐỔI TIỀN TỆ", JLabel.CENTER);
    lblTitle.setFont(new Font("Arial", Font.BOLD, 18));

    this.add(lblTitle, BorderLayout.NORTH); createGUI();
}

void createGUI() {
    JPanel pCen = new JPanel(new GridLayout(2, 1)); JPanel pBot = new JPanel();
    this.add(pBot, BorderLayout.SOUTH); this.add(pCen,
    BorderLayout.CENTER); JPanel p1 = new JPanel();

    JLabel l1, l2;
    p1.add(l1 = new JLabel("Tiền qui đổi:", JLabel.RIGHT));
    p1.add(tfMoney = new JTextField(25));
    pCen.add(p1);

    JPanel p2 = new JPanel();
    p2.add(l2 = new JLabel("Kết quả:", JLabel.RIGHT)); p2.add(tfDisplay = new
    JTextField(25)); tfDisplay.setEditable(false);
    pCen.add(p2);

    l2.setPreferredSize(l1.getPreferredSize());

    pBot.add(btnVND2USD = new JButton("VND2USD"));
    pBot.add(btnVND2EUR = new JButton("VND2EUR"));
    pBot.add(btnUSD2VND = new JButton("USD2VND"));

    pBot.add(btnEUR2VND = new JButton("EUR2VND"));
    btnVND2USD.addActionListener(this);

    btnVND2EUR.addActionListener(this);

    btnUSD2VND.addActionListener(this);
    btnEUR2VND.addActionListener(this);
}

```

```

}
@Override
public void actionPerformed(ActionEvent e) {
    Object o = e.getSource();
    DesignGlobalWebService stub = new DesignGlobalWebService();
    DesignGlobalWebServiceSoap ws = stub.getDesignGlobalWebServiceSoap();
    double dong = Double.parseDouble(tfMoney.getText());
    if (o.equals(btnVND2USD)) {
        try {
            double result = ws.vnd2USD(dong); tfDisplay.setText(result + "");
        } catch (Exception e1) { e1.printStackTrace();}
    } else if (o.equals(btnVND2EUR)) {
        try {
            double result =
                ws.vnd2EUR(dong); tfDisplay.setText(result + "");
        } catch (Exception e1) { e1.printStackTrace();}
    } else if (o.equals(btnUSD2VND)) {
        try {
            double result = ws.usd2VND(dong); tfDisplay.setText(result + "");
        } catch (Exception e1) { e1.printStackTrace();}
    } else if (o.equals(btnEUR2VND)) {
        try {
            double result = ws.eur2VND(dong); tfDisplay.setText(result + "");
        } catch (Exception e1) { e1.printStackTrace();}
    }
}
public static void main(String[] args)
{
    new ClientWebService().setVisible(true);
}

```

3. Ứng dụng Web ASP.NET kết nối tới Web Service

a. Tạo mới ứng dụng ASP.NET Web Application

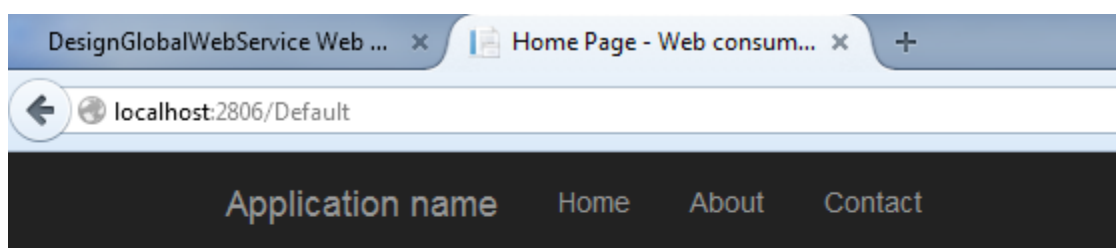
Chọn chuột phải solution trong cửa sổ Solution Explorer, chọn mục ASP.NET Web Application, đặt tên WebClientConvertCurrency

b. Thêm tham chiếu tới Web Service

Chọn chuột phải vào project, chọn Add reference để mở cửa sổ thêm tham chiếu tới Web Service

c. Thiết kế ứng dụng Web Application

Thiết kế ứng dụng như hình sau, và chạy thực thi ứng dụng kiểm tra với dữ liệu như hình bên.



Tiền quy đổi	<input type="text" value="100"/>
Kết quả	<input type="text" value="2000000"/>
<input type="button" value="VND2USD"/> <input type="button" value="VND2EUR"/> <input type="button" value="USD2VND"/> <input type="button" value="EUR2VND"/>	

Bài tập:

Bài tập nâng cao:

VPS là loại mạng gì? Trình bày ưu điểm và khuyết điểm?

Những trọng tâm cần chú ý trong bài:

- Lợi ích khi kết nối các máy tính thành mạng
- Các loại mạng thường gặp và đặc trưng của nó

Yêu cầu về đánh giá kết quả học tập:

Nội dung:

+ Về kiến thức: Trình bày được các lợi ích khi kết nối các máy tính thành mạng, vì sao mạng chuyển mạch gói có tốc độ trao đổi thông tin nhanh hơn tốc độ trao đổi thông tin trong mạch chuyển mạch tin báo?, thế nào là mạng cục bộ LAN (Local Area Networks) và nêu các đặc trưng cơ bản của nó.

- + Về kỹ năng: phân biệt được mạng cục bộ, mạng đô thị, mạng diện rộng.
- + Năng lực tự chủ và trách nhiệm: Tỉ mỉ, cẩn thận, chính xác, ngăn nắp trong công việc.

Phương pháp:

- + Về kiến thức: Được đánh giá bằng hình thức kiểm tra viết, trắc nghiệm, vấn đáp
- + Về kỹ năng: Đánh giá kỹ năng phân biệt được các loại mạng.
- + Năng lực tự chủ và trách nhiệm: Tỉ mỉ, cẩn thận, chính xác, ngăn nắp trong công việc.

BÀI 5: BẢO MẬT TRONG WEB SERVICE

Mã bài: MĐ 26 - 05

Giới thiệu:

Trong bài này giới thiệu về vấn đề bảo mật trong webservice, trình bày một số kiểu giả mạo đánh cắp thông tin và cách phòng tránh, thực hiện việc bảo mật cho webservice

Mục tiêu của bài:

- Trình bày được tổng quan về bảo mật..
- Trình bày được một số kiểu giả mạo, đánh cắp thông tin và cách phòng chống.
- Thực hiện được bảo mật trong Web Service.
- Thực hiện các thao tác an toàn với máy tính.

Nội dung chính:

1. Tổng quan về vấn đề bảo mật

Cùng với sự phát triển không ngừng của Internet, hệ thống thông tin ngày một phát triển vượt bậc. Khái niệm business không còn giới hạn ở bên trong mà đã phát triển cả ở trên Internet.

Khái niệm thương mại điện tử ra đời là sự minh chứng cho sự phát triển đó. Tuy nhiên, đi cùng với sự phát triển đó thì vấn đề an toàn của nó cũng là một vấn đề nóng và cấp bách. Tuy không có khái niệm về sự an toàn tuyệt đối nhưng cũng phải cần một cơ chế an toàn thích hợp để khai thác thương mại trên Internet. Các giải pháp hiện nay đang được sử dụng như mã hóa khóa, chữ ký số có thể đảm bảo ở một mức nào đó. Nhưng cũng phải cần không ngừng hoàn thiện và phát triển các giải pháp đó để đảm bảo phù hợp hơn trên môi trường Internet.

An toàn thông tin trên Internet là một vấn đề chung hiện nay. Sự an toàn web service lại càng cần được sự quan tâm hơn nữa, khi các thông tin nhạy cảm như tài khoản cá nhân ở ngân hàng có thể bị đánh cắp. Điều gì xảy ra nếu sử dụng business service mua hàng, chứng khoán, chuyển tiền mà không có một sự bảo đảm an toàn cần thiết. Đây là một chuẩn an toàn chung cần thiết cần được đáp ứng:

- Identification: định danh được những ai truy cập tài nguyên hệ thống.
- Authentication: chứng thực tư cách truy cập tài nguyên của người muốn sử dụng.
- Authorization: cho phép giao dịch khi đã xác nhận định danh người truy cập.

- Integrity: toàn vẹn thông tin trên đường truyền.
- Confidentiality: độ an toàn, không ai có thể đọc thông tin trên đường đi.
- Auditing: kiểm tra, tất cả các giao dịch đều được lưu lại để kiểm tra.
- Non-repudiation: độ mềm dẻo, cho phép chứng thực tính hợp pháp hóa của thông tin đến từ một phía thứ ba ngoài 2 phía là người gửi và người nhận.

Những yêu cầu trên giúp cho hệ thống an toàn hơn, tránh được phần nào những truy cập không hợp lệ.

HTTP (HyperText Transfer Protocol) là giao thức thường sử dụng nhất cho việc trao đổi thông tin trên Internet, tuy nhiên không an toàn, bởi vì tất cả thông tin được gửi dưới dạng văn bản trong mạng ngang hàng không an toàn. HTTP thuộc về nhóm của những nghi thức như SMTP, telnet, và FTP, được thiết kế trong giai đoạn đầu của Internet khi mà vấn đề an toàn chưa được quan tâm đến nhiều. Một phát triển của HTTP là HTTPS, nó là một chuẩn an toàn cho HTTP. HTTPS cho phép chứng thực client và server qua những chứng thực giữa client và server.

Trước khi có web services security (WS-Security) thì ý nghĩa thông thường của an toàn web service là bảo mật kênh truyền dữ liệu. Nó được thực hiện cho những SOAP/HTTP dựa trên cơ chế truyền thông điệp bằng cách sử dụng giao thức HTTPS. Không giống sự an toàn mức thông điệp, HTTPS cung cấp sự an toàn tới toàn bộ gói dữ liệu HTTP. Bởi vậy, chúng ta không có một tùy chọn nào để áp dụng sự an toàn có chọn lọc chỉ trên những thành phần của một thông điệp.

Mặc dầu HTTPS không bao phủ tất cả các khía cạnh trong chuẩn an toàn chung nhưng nó cũng đã cung cấp một mức bảo chứng đầy đủ với định danh và chứng thực, sự toàn vẹn thông điệp, và độ tin cậy. Tuy nhiên, authentication, auditing, and nonrepudiation chưa được cung cấp. Bên cạnh đó, HTTPS là một giao thức nên khi thông điệp đi qua HTTP server thì lại không an toàn.

2. Một số kiểu giả mạo, đánh cắp thông tin và cách phòng chống

2.1.Message Replay Attack

Message Replay là một kiểu tấn công trên mạng kiểu bắt và gửi lại gói tin mà client đã gửi cho server

Trong quá trình này Attacker như một trung gian của client và server, nó bắt gói tin client và sao chép gói tin đó, chỉnh sửa và gửi lại cho server. Và nếu server có gửi lại cho client thì Attacker cũng có thể bắt được.

Ta phải ngăn chặn việc gửi lại gói tin kiểu này bởi vì các gói tin gửi đi sẽ gây ra không nhất quán dữ liệu, các thông tin gửi đi bị sai lệch ảnh hưởng đến client, việc gửi replayed message liên tục và một cách tự động sẽ làm chết server.

Giải pháp cho vấn đề này là sử dụng cache lưu lại tên định danh cho message và server sẽ loại bỏ các message có định danh bị trùng. Như vậy các message có một tên

định danh duy nhất được gửi đi và chắc chắn rằng các message không bị giả mạo trong quá trình truyền.

Web service security đã cung cấp việc sử dụng Usenametoken trong đó có thể username và password

Các giá trị Nonce được tạo ra ngẫu nhiên cùng với Created là thời gian mà message được gửi đi. Các giá trị này đều được Hash để che dấu thông tin

Giá trị thời gian mà message phải được đồng bộ với thời gian của server. Khi gói tin đến server thì server sẽ giới hạn khoảng thời gian mà message đến, nếu vượt ra khoảng thời gian giới hạn gói tin bị loại bỏ.

2.2. Web Spoofing

Web spoofing là một kiểu lừa đảo trên web phổ biến hiện nay. Có 4 kiểu web spoofing là IP spoofing, ARP spoofing, Web spoofing, DNS spoofing

3. Bảo mật trong web service

Ngày nay công nghệ web services đã và đang được triển khai và ứng dụng trong rất nhiều lĩnh vực khác nhau bao gồm cả những lĩnh vực nhạy cảm, đòi hỏi tính an toàn cao như tài chính, ngân hàng, ... Do đó web service cần cung cấp một mức an toàn đủ để hỗ trợ những công việc như thế. Bên cạnh mặt được của công nghệ web services mang lại thì việc đảm bảo an toàn, tin cậy, toàn vẹn thông tin trao đổi trên web service cũng là một điều rất quan trọng trong quá trình xây dựng web services, bằng việc sử dụng ws security và các thành phần của nó giúp cho thông tin trao đổi trên web services trở nên an toàn hơn.

Trước hết chúng ta xem xét những nhân tố rủi ro ảnh hưởng đến mức an toàn của những ứng dụng dựa trên web service. Chúng ta sẽ sử dụng một kịch bản rút tiền ngân hàng qua mạng để xem xét vấn đề.

Đây là một ứng dụng client/ server đơn giản mô tả một người rút tiền (client) kết nối tới trung tâm dữ liệu của ngân hàng để sử dụng một ứng dụng web service để thực hiện yêu cầu của mình. Nếu không có sự an toàn nào đã được áp dụng, thì có ba nhân tố mạo hiểm chính:

- Những giao dịch không hợp pháp (Unauthorized transactions) : một người nào đó không có quyền nhưng vẫn yêu cầu rút tiền. Giao dịch này không hợp pháp. Chúng ta cấm vấn đề này bằng cách sử dụng cơ chế chứng thực của WS - Security. Một ví dụ của sự chứng thực bao gồm phải có một kết hợp user ID/ password trong SOAP message.

- Những thông báo không mã hóa (Readable messages in clear text-no encryption): số hiệu tài khoản và số dư tài khoản trong gói SOAP rất dễ bị đọc lên trên mạng. Việc lộ thông tin này là do thông tin tài khoản và số dư được gửi qua mạng dưới định dạng văn bản. Để giải quyết vấn đề này, thông tin này phải được mã hóa ở mức kênh chuyên thông điệp hoặc ở mức thông điệp (WS - Security).

Những thông điệp bị thay đổi hoặc mất mát (SOAP message susceptible to modification-no integrity). Trong quá trình chuyển thông tin từ người rút tiền đến trung

tâm dữ liệu, nó có thể bị chặn. Và những thông tin này có thể bị thay đổi, ví dụ như số tài khoản là 1234 thì bị thay đổi thành số 9876. Vấn đề này dẫn đến thiếu sự toàn vẹn.

Những ví dụ trên, chúng ta đã mô tả sự an toàn liên quan tới những yêu cầu của sự chứng thực, tính bí mật, và sự toàn vẹn thông tin.

Trước khi có WS-Security, sự an toàn kênh chuyên thông điệp rất thường được sử dụng. Sự an toàn kênh chuyên thông điệp ở chỗ là nó mã hóa toàn bộ thông điệp, dẫn đến sử dụng CPU cao hơn. Tuy nhiên với WS-Security, nó cung cấp những cách tối ưu hóa những thao tác an toàn, mà yêu cầu ít thời gian sử dụng CPU hơn. Dựa vào mức an toàn cần thiết mà một hoặc nhiều hơn những cơ chế an toàn này có thể được áp dụng cho một ứng dụng.

Tuy nhiên việc chọn cơ chế an toàn cho web service phải đòi hỏi sao cho người dùng không cảm thấy quá phức tạp tạo một sự gò bó, do đó việc chọn cơ chế an toàn nào trong ws security thì phụ thuộc nhiều vào loại service và những tính năng mà service này cung cấp, ví dụ như service về giao dịch tài chính ngân hàng phải có cơ chế an toàn hiệu quả hơn so với service chọn lọc và phân loại tin tức hay service cho biết tỷ giá trao đổi giữa các loại ngoại tệ,... Bên cạnh đó còn một điểm cần quan tâm đó là sự an toàn không chỉ phụ thuộc vào những giải thuật, những tiêu chuẩn, và những cơ chế mà ws security mang lại, mà nó còn tùy vào thái độ của các công ty có hiểu rõ tầm quan trọng của an toàn thông tin khi triển khai các ứng dụng, giao dịch trên mạng hay không cũng rất cần thiết.

Có hai hình thức bảo mật, đó là bảo mật trên kênh truyền và bảo mật ở mức thông điệp. Hiện nay hầu hết các dịch vụ đều kết hợp cả hai hình thức để tối ưu cho việc bảo mật.

Bảo mật ở mức kênh truyền : trên kênh truyền phải bảo đảm được thông điệp an toàn và toàn vẹn. Ở mức này thì ta thường dùng kênh truyền an toàn như HTTP + SSL để tạo kết nối an toàn đến client. An toàn ở mức này đòi hỏi cơ sở hạ tầng mạng phải tốt.

Bảo mật ở mức thông điệp : để tăng mức độ an toàn cho thông điệp, ta sử dụng thêm WS security cung cấp mức an toàn cho thông điệp. Các dữ liệu được mã hóa và được sử dụng chữ kí số để tránh bị đánh cắp thông tin. Cả hai bên server và client đều sử dụng các key để có thể chứng thực lẫn nhau.

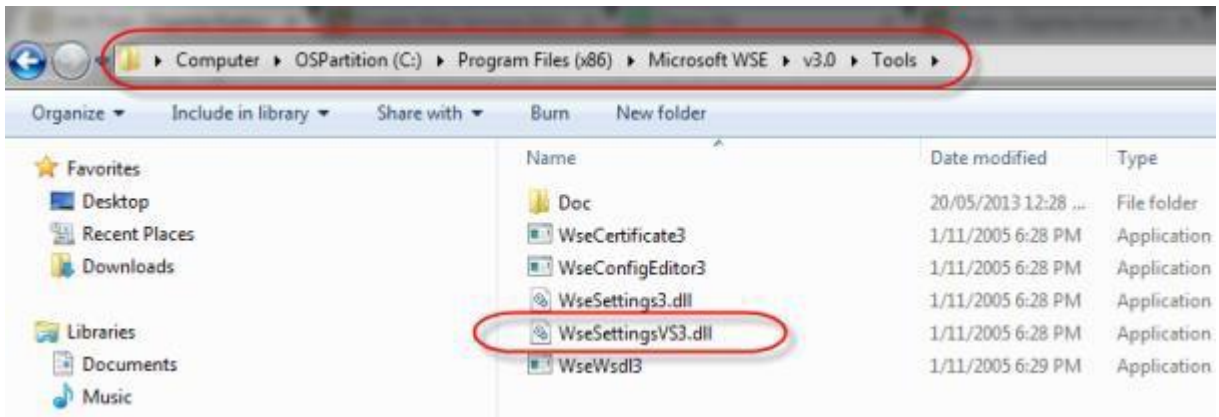
4. Thực hiện bảo mật trong web service

1.1. Cài đặt WSE 3.0

Download gói cài từ Microsoft tại link sau <http://www.microsoft.com/en-us/download/details.aspx?id=6545>

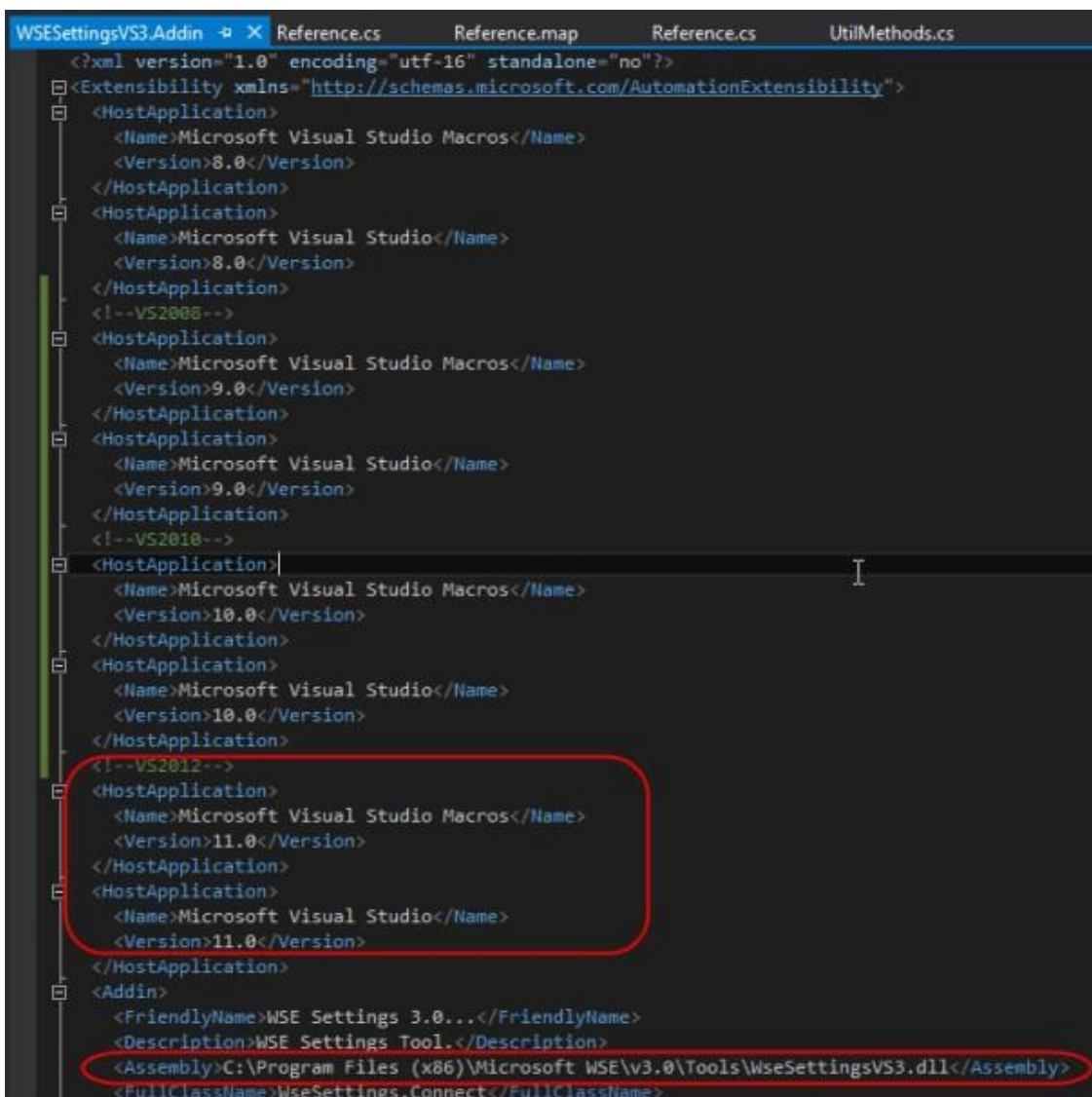
a. Tích hợp WSE 3.0 vào Visual Studio 2008, 2010, 2012, 2013

Sau khi cài đặt thành công WSE 3.0, ta cần khởi động lại máy và không bật Visual Studio trong quá trình cấu hình. Ta sẽ thấy thư mục WSE như sau



Vào thư mục sau %ALLUSERSPROFILE%\Application Data\Microsoft\MSEnvShared\AddIns

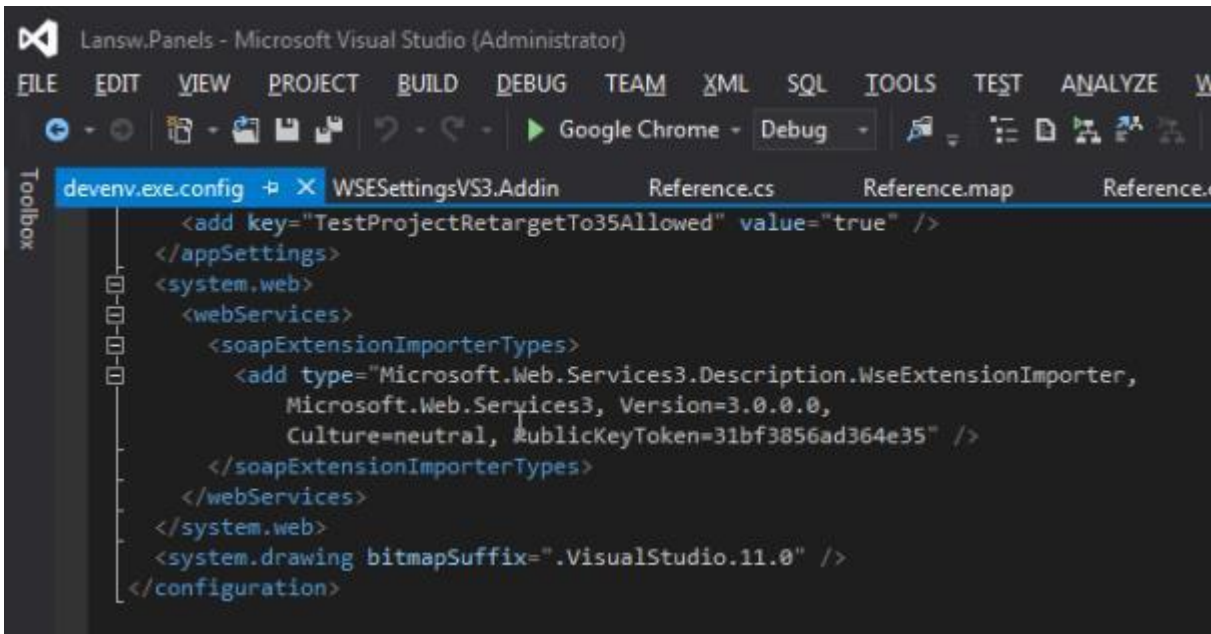
Trong thư mục ta sẽ tìm thấy file WSESettingsVS3.AddIn, trước khi chỉnh sửa ta sẽ cần backup lại file này. Ta sẽ chỉnh sửa nội dung như hình sau



Ta sẽ thêm vào các khối bôi đỏ tương ứng cho Visual Studio 2008, 2010, 2012, 2013.

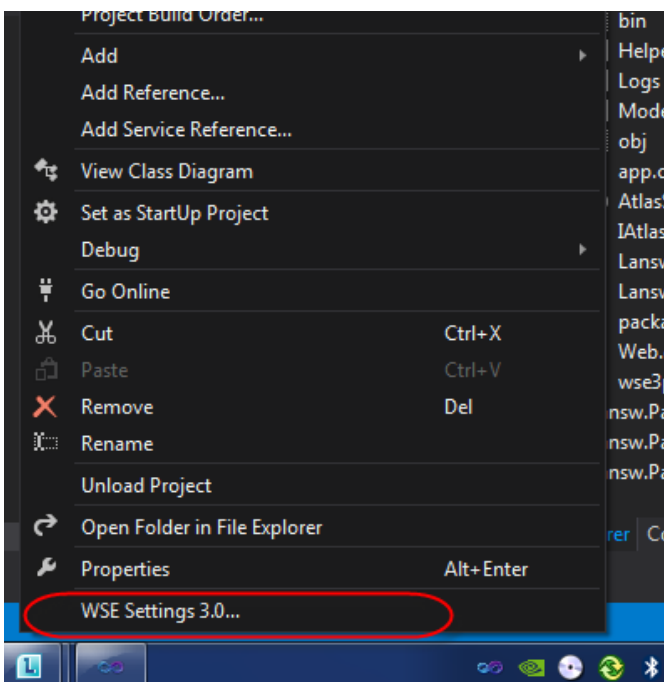
Tới thư mục C:\Program Files (x86)\Microsoft Visual Studio 11.0\Common7\IDE (cho VS2012) hoặc C:\Program Files\Microsoft Visual Studio 9.0\Common7\IDE (cho VS2008) hoặc C:\Program Files\Microsoft Visual Studio 10.0\Common7\IDE (cho VS 2010) hoặc C:\Program Files\Microsoft Visual Studio 12.0\Common7\IDE (cho VS 2013).

Mở file devenv.exe.config và thêm những dòng sau (nhớ backup lại file trước khi chỉnh sửa)



```
<add key="TestProjectRetargetTo35Allowed" value="true" />
</appSettings>
<system.web>
  <webServices>
    <soapExtensionImporterTypes>
      <add type="Microsoft.Web.Services3.Description.WseExtensionImporter,
        Microsoft.Web.Services3, Version=3.0.0.0,
        Culture=neutral, PublicKeyToken=31bf3856ad364e35" />
    </soapExtensionImporterTypes>
  </webServices>
</system.web>
<system.drawing bitmapSuffix=".VisualStudio.11.0" />
</configuration>
```

Tới đây ta đã tích hợp thành công WSE 3.0 vào công cụ Visual Studio

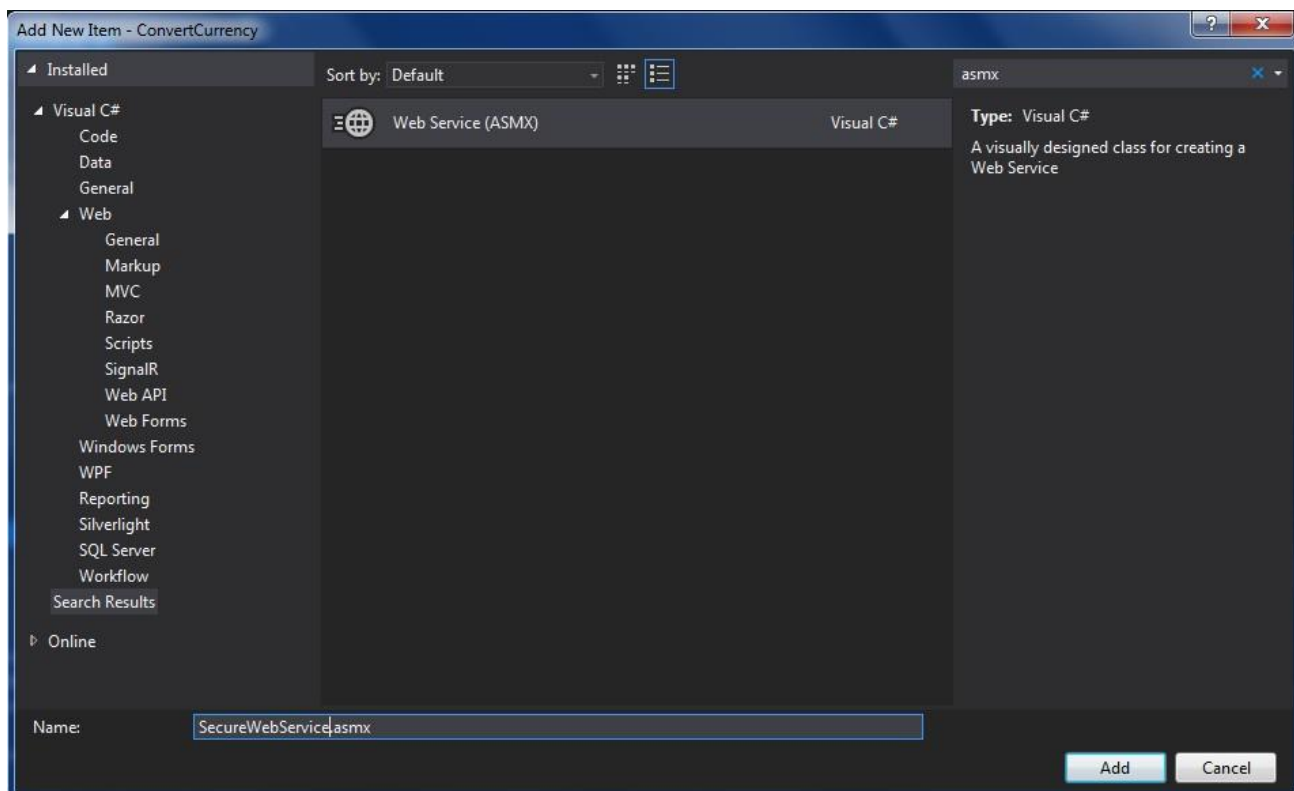


Chú ý: nếu hệ thống không cho phép chỉnh sửa trực tiếp các file trên, ta sẽ tạo ra file mới và ghi đè lên file cần chỉnh sửa.

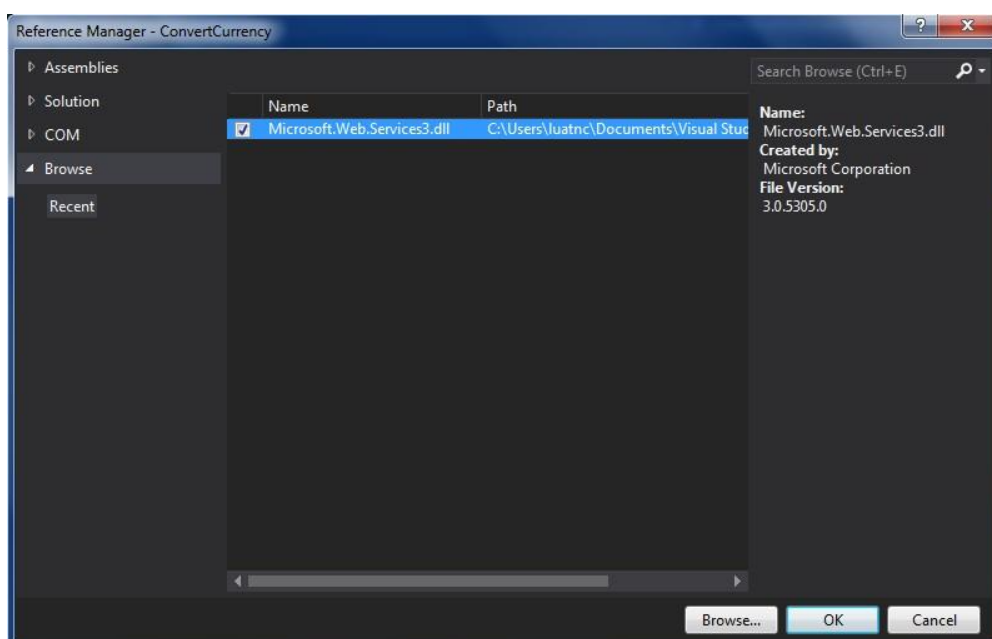
1.2. Xây dựng ứng dụng bảo mật Web service thông qua bảo mật thông điệp SOAP với WS-Security

Tạo mới Web Service với WSE 3.0

Tạo ứng dụng bảo mật Web service đặt tên SecureWebService.asmx như sau



Thêm tham chiếu tới thư viện WSE 3.0 đã cài đặt, chọn Add Reference và chọn tới file Microsoft.Web.Services3.dll trong thư mục đã cài đặt WSE



Ta thêm khai báo sử dụng thư viện tới file SecureWebService.asmx như sau
using Microsoft.Web.Services3;

```
using Microsoft.Web.Services3.Security.Tokens;
```

b.Thực thi Secure Web Service

```
[WebMethod]
```

```
public string Hello(string name)
```

```
{  
    //Get the current soap context  
    SoapContext ctxt = RequestSoapContext.Current;  
    if (ctxt == null)  
    {  
        //This request is using a different protocol other than SOAP.  
        return "Please format the request as a SOAP request and try again.";  
    }  
    //Iterate through all Security tokens  
    foreach(SecurityToken tok in ctxt.Security.Tokens)  
    {  
        if (tok is UsernameToken)  
        {  
            UsernameToken user = (UsernameToken)tok; return "Hello  
Authenticated user " + user.Username;  
        }  
    }  
    return "Hello Liar";  
}
```

Trong đoạn mã trên, ta lặp qua các thông tin bảo mật Header vì mỗi thông điệp SOAP có thể chứa nhiều khai báo Header. Sau đó, tìm kiếm chắc chắn rằng có khai báo UsernameToken trong đó.

Ta sẽ F5 để chạy ứng dụng Web Service để cho phép nó sẵn sàng phục vụ



The following operations are supported. For a formal definition, please review the [Service Description](#).

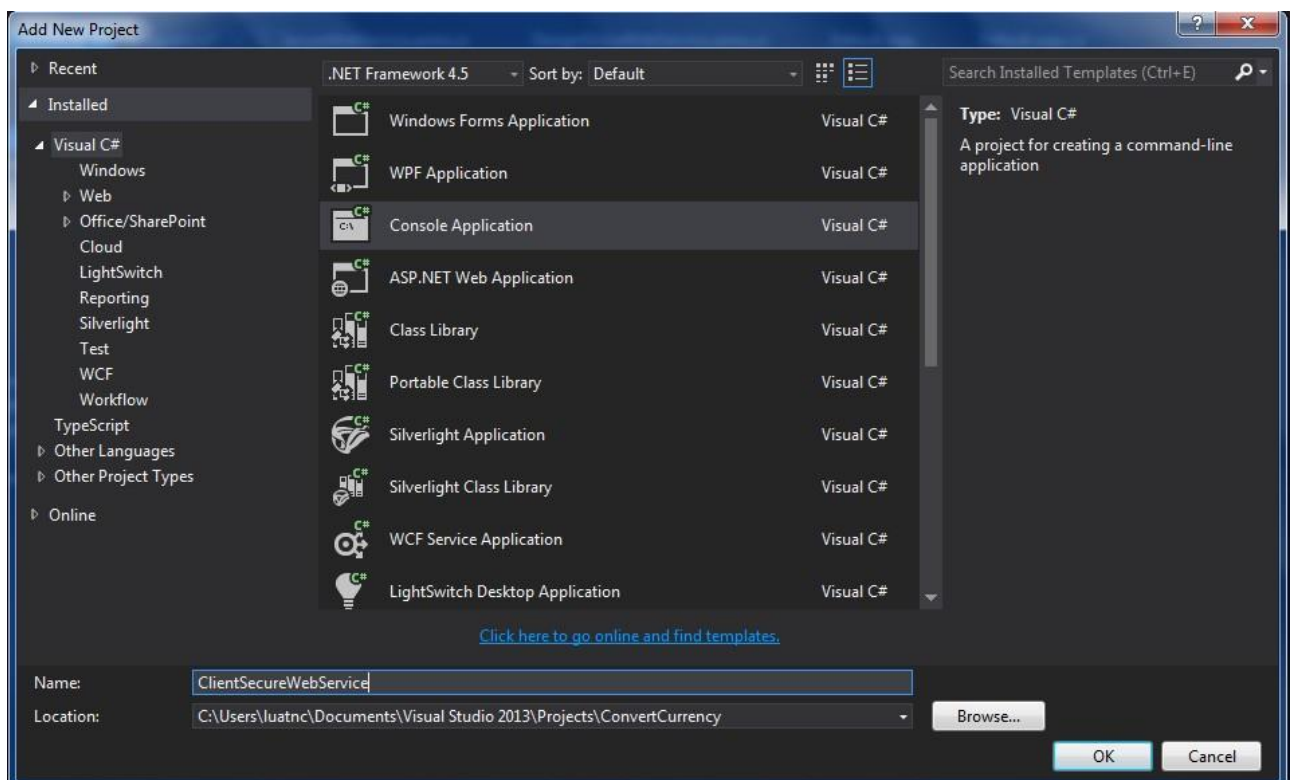
- [CheckAuthentication](#)

This web service is using <http://tempuri.org/> as its default namespace.

Recommendation: Change the default namespace before the XML Web service is made public.

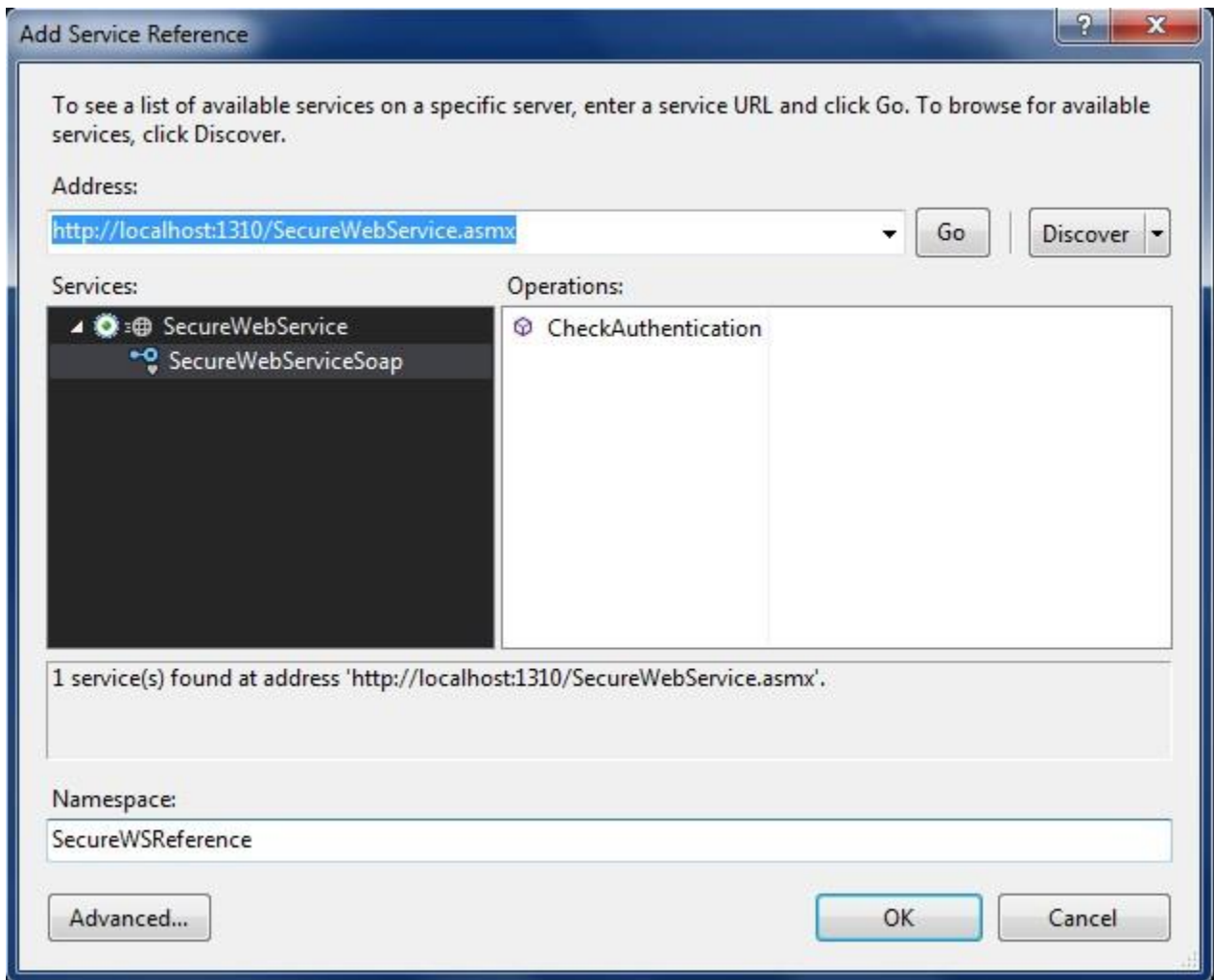
c. Tạo ứng dụng client kết nối tới SecureWebService

Tạo project Console Application đặt tên ClientSecureWebService như hình sau



Thêm tham chiếu tới thư viện WSE 3.0 đã cài đặt, chọn Add Reference và chọn tới file Microsoft.Web.Services3.dll trong thư mục đã cài đặt WSE.

Thêm tham chiếu tới Secure Web service vừa tạo, bằng cách cung cấp URL như hình sau và đặt tên SecureWSReference



Ta thêm khai báo sử dụng thư viện tới file program.cs như sau

```
using Microsoft.Web.Services3;
using Microsoft.Web.Services3.Security.Tokens;
```

Phần thực thi service như sau

```
[STAThread]
static void Main(string[] args)
{
    Console.WriteLine("Enter Name: "); string name = Console.ReadLine();
    Console.WriteLine("Enter Password: ");
    string password = Console.ReadLine();

    SecureWSReference.SecureWebServiceSoapClient proxy = new
    SecureWSReference.SecureWebServiceSoapClient();

    proxy.RequestSoapContext.Security.Tokens.Add(new UsernameToken(name,
password, PasswordOption.SendHashed));
    proxy.RequestSoapContext.Security.Timestamp.TtlInSeconds = 300;
```

```

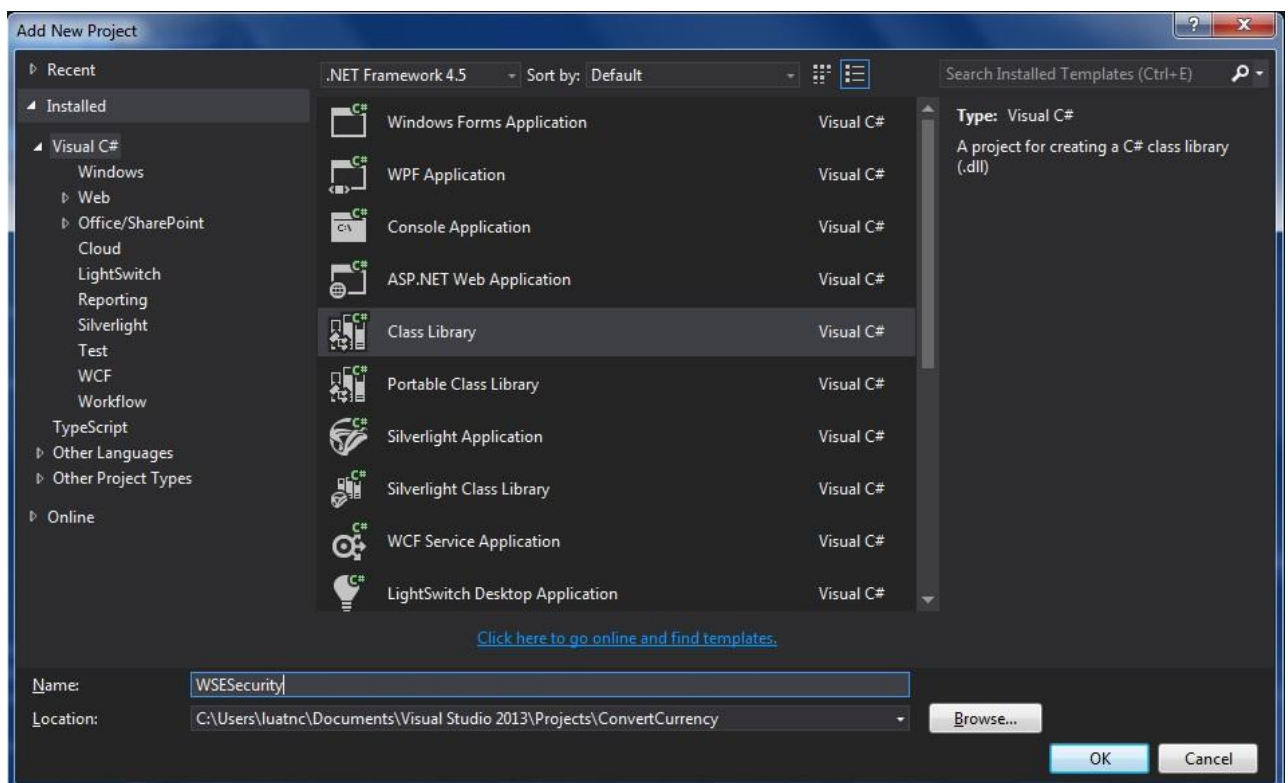
    Console.WriteLine(proxy.Hello(name)); Console.WriteLine("Hit enter to end.");
    Console.ReadLine();
}

```

Trong đoạn mã trên, thành phần UsernameToken được đóng gói vào trong SOAP Header và gửi đi và Password được băm nhờ giải thuật SHA1 trước khi gửi. Nhờ đó, việc bảo mật thông tin trong thành phần SOAP Header dựa vào WS-Security là rất đảm bảo. Bên cạnh đó, client hoàn toàn có thể cấu hình thời gian Timeout trong trao đổi thông điệp để tăng cường khả năng chống tấn công từ bên ngoài

Ở trên, ta thấy WS-Security đã dùng cơ chế UsernameToken để bảo mật thông tin trong SOAP Header. Tuy nhiên, với những ứng dụng yêu cầu phải có khả năng xác thực phức tạp hơn, như kiểm tra thông tin tài khoản có được lưu trữ trong Cơ sở dữ liệu, ... Do đó, ta cần đóng gói thành thư viện riêng để đảm nhận xử lý các bussiness logic phức tạp tùy thuộc vào yêu cầu của hệ thống.

Tạo mới project Class Library như sau



Chọn chức năng hỗ trợ WSE bằng cách chuột phải vào project và chọn WSE 3.0 Setting.

Đổi tên Class1.cs thành CustomAuthenticator.cs và nhập vào nội dung như sau

```

[SecurityPermissionAttribute(SecurityAction.Demand,
Flags = SecurityPermissionFlag.UnmanagedCode)]
public class CustomAuthenticator : UsernameTokenManager

```

```

{
//Returns the password or password equivalent for a user name protected override string
AuthenticateToken(UsernameToken token)
{
    if (token == null)
        throw new ArgumentNullException(); if (token.Username == "hoand")
        return "mypassword";
    else
        return null;
}
}

```

Ở đây, ta có lớp CustomAuthenticator kế thừa từ lớp quản lý UsernameTokenManager và ghi đè phương thức AuthenticateToken xử lý xác thực.

Vậy ta đã có thư viện đảm nhận việc xác thực thông tin UsernameToken nằm trong SOAP Header gửi tới. Thư viện này sẽ đảm nhận việc xử lý bussiness logic phức tạp và được sử dụng bởi Server và tránh việc lộ các thông tin quan trọng đối với bất cứ thành phần nào khác.

Để sử dụng lại thư viện này ở các project khác, ta cần build ra file WSESecurity.dll và sẽ được tham chiếu tới bởi các Web service project. Thêm nữa là ta cần tích hợp nó vào luồng xử lý WSE pipeline của các Web service bằng cách khai báo thêm (dưới thẻ <configuartion>) vào file web.config như sau

```

<microsoft.web.services3>
<security>
<securityTokenManager qname="wsse:UsernameToken"
type="WSESecurity.CustomAuthenticator, WSESecurity"
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-wssecurity-secext-1.0.xsd"/>
</security>
</microsoft.web.services3>

```

Thư viện Web Service Enhancement 3.0cung cấp một phương thức đơn giản nhưng hết sức tuyệt vời để quản lý bảo mật trao đổi thông điệp trong Web Service. Module xử lý xác thực hoàn toàn có thể cập nhật lại bussiness logic mà không cần phải biên dịch lại Web Service thông qua thư viện WSESecurity xử lý riêng như ta đã thực hiện ở trên

Bài tập:**Bài tập nâng cao:**

VPS là loại mạng gì? Trình bày ưu điểm và khuyết điểm?

Những trọng tâm cần chú ý trong bài:

- Lợi ích khi kết nối các máy tính thành mạng
- Các loại mạng thường gặp và đặc trưng của nó

Yêu cầu về đánh giá kết quả học tập:**Nội dung:**

+ Về kiến thức: Trình bày được các lợi ích khi kết nối các máy tính thành mạng, vì sao mạng chuyên mạch gói có tốc độ trao đổi thông tin nhanh hơn tốc độ trao đổi thông tin trong mạch chuyển mạch tin báo?, thế nào là mạng cục bộ LAN (Local Area Networks) và nêu các đặc trưng cơ bản của nó.

+ Về kỹ năng: phân biệt được mạng cục bộ, mạng đô thị, mạng diện rộng.

+ Năng lực tự chủ và trách nhiệm: Tỉ mỉ, cẩn thận, chính xác, ngăn nắp trong công việc.

Phương pháp:

+ Về kiến thức: Được đánh giá bằng hình thức kiểm tra viết, trắc nghiệm, vấn đáp

+ Về kỹ năng: Đánh giá kỹ năng phân biệt được các loại mạng.

+ Năng lực tự chủ và trách nhiệm: Tỉ mỉ, cẩn thận, chính xác, ngăn nắp trong công việc.

TÀI LIỆU THAM KHẢO

- [1] Website: <http://w3schools.com>
- [2] Website: <http://jquery.com>
- [3] Website: <http://asp.net> <https://facebook.com>